

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

UZMAN SİSTEMLER
VE
ÜRETİM PLANLAMADA UYGULANMALARI

End. Müh. Sare Olcay AKIN

F.B.E. Endüstri Mühendisliği Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Yaşar Baki CENGİZ

Prof. Yaşar B. Cengiz

Prof. Dr. Turay Gökçen

Prof. Dr. Turat Dingmen

İSTANBUL, 1997

15.4.1997

İÇİNDEKİLER

TEŞEKKÜR	iv
ÖZET	v
ABSTRACT	vii
 1. BÖLÜM : UZMAN SİSTEMLER	
1.1. Giriş.....	1
1.2. Uzman Sistemlerin Tanımı	2
1.3. Uzman Sistemlerin Mimarisi	3
1.4. Uzman Sistemlerin Avantajları	5
1.5. Uzman Sistemlerin Kullanım Alanları ve Sınırları	7
1.6. Uzman Sistemlerin Uygulama Örnekleri	10
1.7. Uzman Sistemler İçin Temel Terminoloji	13
 2. BÖLÜM : UZMAN SİSTEMLERDE BİLGİNİN ORGANİZASYONU	
2.1. Giriş	15
2.2. Bilgi Süzme	16
2.3. Bilgi Analizi	18
2.4. Bilgi Temsili	18
2.4.1. Global Veri Tabanı	20
2.4.1.1. Kural Tabanlı Bilgi Temsili	20
2.4.1.2. Yapılanmış Bilgi Temsili	21
2.4.1.3. Mantık Tabanlı Bilgi Temsili	25
2.4.2. Bilgi Tabanı	28
2.4.3. Kontrol Stratejisi	28
2.4.3.1. İleriye Doğru Zincirleme	31
2.4.3.2. Geriye Doğru Zincirleme	35
2.4.3.3. Belirsizlikle Mücadele	38

3. BÖLÜM : UZMAN SİSTEMLERİN TASARLANMASI

3.1. Uzman Sistemlerin Tasarlanmasında Kullanılan Yazılımlar	39
3.2. Uzman Sistemlerin Geliştirilmesi	44
3.2.1. Problemin Belirlenmesi Ve Çözüm Metodolojisi	44
3.2.2. Prototip Oluşturma	47
3.2.3. Kullanım ve Süreklilik Sağlama	48

4. BÖLÜM : ÜRETİM PLANLAMADA UZMAN SİSTEMLER

4.1. Üretim Planlama Problemleri	50
4.2. Üretim Planlamaya Önceki Yaklaşımlar	51
4.3. Üretim Planlamaya Uzman Sistem Yaklaşımının Uygulanması	52
4.3.1. Üretim Planının Hazırlanması	54
4.3.1.1. Yıllık ve Orta Vadeli Planlama	54
4.3.1.2. Kısa Vadeli Planlama	55
4.3.2. Kullanıcı Arabirimi	57
4.3.3. Model Tabanı	59
4.3.4. Veri Tabanının Oluşturulması	60
4.4. Üretim Planlama İçin Geliştirilmiş Uzman Sistem Örnekleri	62
4.4.1. ReDS (Requirement Driven Scheduling)	62
4.4.1.1. ReDS'nin Prensipleri	62
4.4.1.2. ReDS'nin Kavramsal Yapısı	65
4.4.1.3. ReDS'NİN Mimari Yapısı	67
4.4.2. ReDS ² (Real Time Distributed Scheduling)	69
4.4.2.1. ReDS ² Çekirdeği	71
4.4.2.2. ReDS ² İşlemleri	72
4.4.2.3. ReDS'nin İtme ve Çekme Davranışı	75
4.4.3. AERPLAN	75
4.4.3.1. Problem Alanının Analizi	76
4.4.3.2. Uzman Sistem Nasıl Yardım Edebilir?	79
4.4.3.3. AERPLAN Sisteminin Mimarisi	81

4.4.3.4. Problem Çözümü	82
4.4.3.5. Sistemin Performansı	90
5. BÖLÜM : SONUÇ	93
REFERANSLAR	ix
ÖZGEÇMİŞ	xii



TEŞEKKÜR

Tez çalışmam sırasında yardımlarını esirgemeyen danışmanım Prof. Yaşar Baki CENGİZ'e, Yrd. Doç. Dr. İbrahim ERDEM'e, Ar. Gör. Tufan DEMİREL'e, Ar. Gör. Burak TALAY'a, tercümelerim sırasındaki yardımları için kardeşim Nuray Akın'a ve manevi destekleri için aileme teşekkür ederim.



ÖZET

Günümüzde bilgisayarların gelişimi ile en güç problemlere bile çözüm bulunabilmektedir. Ancak öyle problemler vardır ki, güçlü hesaplama mekanizmalarına rağmen geleneksel yöntemlerle çözüme ulaşamaz. Günlük yaşıntıda karşılaşılan bu problemler belirsiz, kesinliği olmayan ve karmaşık bilgi işlemleri gerektirir. Yaygın algoritmalarla formüle edilemeyen bu tip problemler için, sistematik bir araştırma yapılarak olası tüm çözüm yolları tek bir yolda birleştirilir. İşlemler, uzman bilgiye dayalı deneysel yöntemlerle sınırlanır. İşte bu uzman bilginin yer aldığı sistemlere uzman sistemler adı verilir.

Uzman sistem kavramı, uzman kişilerin çözümleme yeteneklerini ve muhakeme gücünü taklit ederek, yüksek maliyetli kaynaklarla ilgili uzmanlık bilgisini makinaya yüklemek ve kullanılabilir hale getirmektir. Uzman kişiler, kendi alanlarına giren karar verme problemlerini yaratıcı yönleriyle çözerler. Önceki olaylardan kazandıkları bilgi birikimleri uzmanların düşüncelerini destekler. Belirsizlik taşıyan karar durumlarında geçerli bilginin kullanılması ve uygun sonucun çıkarılması uzmanlık olarak adlandırılır. Böyle bir davranışın programlama tekniğine uygun olarak gösterilmesi istendiğinde, kazanılmış bilgilerle, olaya özgü verilerden sonuç çıkarılmasını sağlayan bir kılavuz gereklidir. Bir tarafta genel olarak bilgilenmeyi temsil eden durumlar, diğer tarafta da uzmanın deneyimlerini ifade eden kurallar vardır, bunların birleştirilmesi ile sonuçlar çıkarılır.

Uzman sistemler, standart çözüm prosedürü olmayan karar süreçlerinin desteklenmesine hizmet verdiklerinden hızla işletmelere girmeye başlamıştır. Yapay zekanın bir kolu olan uzman sistemler, bir çok bilimsel alanda problem çözümüne yardımcı olmada önemli başarılar elde etmişlerdir. Uzman sistemlerin işletme yönetiminde özellikle üretim yönetiminde uygulanabilirliği ise ispatlanmıştır. Üretim planlama ve çizelgeleme, üretim yönetimindeki en zor görevlerden biri olmuştur. Çizelgeleme teorisindeki gelişmelere rağmen, bir çok çizelgeleme problemi analitik çözümlerin uygulanabilmesi için hala çok karmaşıktır. Planlama probleminin zorluğu, çoğu zaman birbiriyle çelişen ve iyi tanımlanamayan bir çok farklı kısıtla ve farklı amaçla uğraşmak zorunda olmasından ileri gelir. Üretim ortamının dinamik bir ortam olması da, planlama görevinin zorluğuna katkıda bulunan bir etkidir.

Hazırlanan bu tez, uzman sistemlerin üretim planlamaya uygulanabilirliği üzerine yorumlar getirmektedir. Birinci bölümde uzman sistemler genel hatlarıyla anlatılmıştır. Uzman sistemlerin yapısı, avantajları ve dezavantajları, kullanım alanları, dünyadaki uygulamaları incelenmiştir.

İkinci bölümde, uzman sistemlerin tasarlanmasında karşılaşılan sorunlardan biri olan uzmanlık bilgisinin toplanması ve bu bilginin bilgisayarın anlayacağı şekilde temsil edilmesi konuları açıklanmıştır. Bilgi temsilde kullanılan metodlar detaylı olarak açıklandıktan sonra, bilgisayarın sonuç çıkarmada kullandığı teknikler örneklerle anlatılmıştır.

Üçüncü bölümde, uzman sistemleri tasarlamak için kullanılabilecek yazılım araçları ve uzman sistem oluşturma aşamaları incelenmiştir.

Uzman sistemlerin üretim planlama ve çizelgeleme problemlerine uygulanış yöntemini anlatan iki örnek dördüncü bölümde ayrıntılı olarak incelenmiştir. Burada ReDS (Real-time Distributed Scheduling) ve AERPLAN uzman sistemleri tanıtılmıştır. Üretim planlama ve çizelgeleme görevinin karmaşıklığını oluşturan nedenler, planlamaya uzman sistemlerden önceki yaklaşımlar, örnek bir işletmeye uzman sistem yerleştirmek için yapılan bir analiz de bu bölümü oluşturan konulardandır.

Sonuç bölümü olan beşinci bölümde, uzman sistemler ile üretim planlamaya nasıl yardım edilebileceği kısaca anlatılmıştır.



ABSTRACT

Today with the development of computers it is possible to find solutions even to the most difficult problems. However, there are some problems for which it is not possible to find solutions despite of strong calculating methods. These problems are encountered in everyday life and require indefinite and complex knowledge processing. For these problems which cannot be formulated by common algorithms, all the possible solution methods are aggregated in a single way by a systematic survey. The process is restricted with experimental methods that depend on expert knowledge. These systems which involve expert knowledge are called expert systems.

The expert system concept means loading down the expertise knowledge to the computer about high costly resources by imitating experts problem solving skills and reasoning talent. Experts solve decision making problems by using their intuition. Their experience support their thoughts. Using valid knowledge and inferring the appropriate solution for indefinite decision situations is called as expertise. To present such behavior compatible with the programming techniques, a guide is necessary to infer a solution from the gained knowledge and domain specific data. There are facts representing knowledge and there are rules expressing expert's experiences, and solutions are inferred by connecting them.

Expert systems begin to take role in business environment, because they serve to support decision processes which do not have standard solution procedures. Expert systems which are a branch of artificial intelligence, have already achieved significant success in helping to solve problems in a number of scientific fields. The applicability of expert systems in business administration, especially in the field of production management is broadly based. Production planning and scheduling has been one of the most difficult tasks in production management. Despite the advances in scheduling theory, many actual scheduling problems are still too complex to be applied analytical solutions. The difficulty of planning problems stem from having to deal with a large set of diverse constraints and multiple objectives that are often conflicting. The dynamic nature of the production environment contributes to the complexity of the task.

This thesis comments on the applicability of expert systems in production planning and scheduling. In the first chapter, expert systems are explained generally. Their structures, advantages and disadvantages, their application areas and worldwide applications are investigated.

In the second chapter, taking expert knowledge and representing it in a computer-compatible way, which is one of the difficulties encountered in designing expert systems, is explained. After the methods used for representing knowledge are explained in detail, inference techniques are illustrated.

In the third chapter, software tools in designing expert systems and steps of implementing an expert system is explained.

Two examples that explain applications of expert systems in production planning and scheduling problems are explained in the fourth chapter. Here, ReDS (Requirement Driven Scheduling) and AERPLAN expert systems are presented. The reasons that cause the complexity of production planning and the scheduling task, previous approaches to planning, an analysis to install expert systems to a firm are the subjects involved in that chapter.



1. BÖLÜM

UZMAN SİSTEMLER

1.1. GİRİŞ

Uzman sistemler (expert systems), bilgisayara dayalı olarak problem çözmede yeni bir şekil olarak, seksenli yılların ortalarında ilgi odağı olmuş ve birçok uygulama alanı bulmuştur. Bilgi teknolojisinde günümüzde ulaşılan düzeyi ifade eden yapay zekanın (artificial intelligence) uygulama alanlarından biridir.

İşletme ve karar kuramları açısından, yönetim kararlarının bilgisayarlarla desteklenmesinde, artık yalnızca kantitatif ağırlıklı değerlemeler aşılmaktadır; kalitatif (niteliksel) karar faktörleri de bilgisayarda etkin olarak işlenebilmektedir. Bir uzman sistem ile, bir uzmanlık alanının kararları, karar destekleyici sistemle birleştirilebilmektedir. Böylece uzman sistemler, şimdiye kadar uzman kişilerin bilgi birikimi ve deneyimleriyle çözdükleri problemleri çözmeyi üstlenmektedirler. Bunun için uzman sistem, bünyesinde bir uzmanın bilgisini barındırmakta ve kullanıcıya bu bilgiler ışığında ulaştığı sonucu sunmaktadır.

Uzman sistemler, yapay zekanın endüstride en yaygın olarak kullanılan araçlarından biridir. Bu araçların temel amacı, emek yoğun sistemleri bilgi yoğun sistemler haline getirmek ve verimliliği artırmaktır. Yapay zeka için kesin bir tanım yapmak mümkün olmasa da, yapılan tanımların ortak noktası, bilgisayarlar ile zeki davranışların özleştirilmesidir. Yapay zekanın ve yapay zeka programlarının rakamlar ve veriler yerine bilgiye dayalı mantıksal bir süreçten ibaret olduğu vurgulanmaktadır. Bu dalın önde gelen isimlerinden olan Feigenbaum ve McCorduck (1984) yapay zekayı, "bilginin sembolik olarak temsiliyle, mantıksal işlemleri yapan teknikler ve insana özgü akıllılık derecesinde hareket eden bilgisayar programları" olarak tanımlarlar. Bu amaç doğrultusunda İngiliz bilim adamı Alan M. Turing'in ana tasarımını ve psikologların

insanın düşünme ve hatırlama sistemi hakkındaki düşüncelerini temel alarak çalışmalar yapılmış ve çeşitli yapay zeka alt dalları geliştirilmiştir. Bu alt dalların başlıcaları oyun programları, otomatik teorem ispatı, insan davranışlarının modelinin kurulması, robotik, makina öğrenimi, görme sistemleri, konuşulan dilin bilgisayarlara kazandırılması ve uzman sistemlerdir.

1.2. UZMAN SİSTEMLERİN TANIMI

Uzman sistemler belirli bir alanda, sadece o alan ile ilgili bilgilerle donatılmış ve problemlere o alanda uzman bir kişinin getirdiği şekilde çözümler getirebilen bilgisayar programlarıdır. Buradaki zeki davranış, ilgili uzmanın probleme yaklaşırken oluşturduğu çözüm mekanizmasında yatar. Uzman sistemlerle ilgili literatürdeki tanımlar şöyledir:

" Bilgi, tecrübe ve uzmanlaşmış profesyonel kişilerin yargılarını içeren programlara uzman sistemler adı verilir. Bu programlar genelde insan uzmanlığını gerektiren problemlerin çözümünde kullanılır. Problem çözümlemenin ötesinde, uygun soruların sorulması sebep-sonuç çıkarmanın açıklanması gibi diğer uzman fonksiyonlarını da gerçekleştirirler (Waterman et al, 1983)."

" Uzman sistemler, uzmanların problem çözme ve karar verme işlemlerinin doğrultusunda hareket etmeye çalışan bilgisayar programlarıdır. Uzman sistemin amacı, özel problemleri çözmek, sunmak ve çözümü detaylı bir şekilde kullanıcılara açıklamaktır (Madni, 1988)."

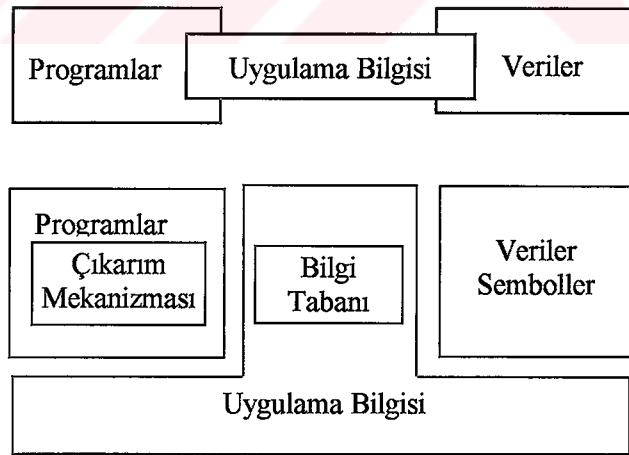
" Uzman sistemler insan uzmanının dizayn, planlama, teşhis etme, yorumlama, özetleme ve tavsiye etme gibi yapabileceği türden faaliyetleri yapmak için hazırlanmış bilgisayar programlarıdır (D'agapeyeff, 1989)."

Tanımlardan da anlaşıldığı gibi uzman sistemler bilgisayar programlarından başka bir şey değildirler. En önemli özellikleri akıllı tarzda işlem yapacak şekilde hazırlanmış olmalarıdır. Yani, hazırlanan programın, sanki bir insan uzmanının var olduğunu

hissettirmesi lazımdır. Bu sistemde hedef, gerek duyulan bilgi ve tecrübeyi mümkün olduğu kadar yazılımın içine katmak ve bilgisayarda insan uzmanın kopyasını sağlamaktır. Bu sistemlerin kurulması ve uygulanması, mühendisler, bilim adamları ve yöneticilerin, programlama dilleri ve simülasyon projelerini gerçekleştirme konularında özel bir eğitime gerek duymadan, kolay ve doğru olarak çalışmalarına imkan verecektir.

1.3. UZMAN SİSTEMLERİN MİMARİSİ

Geleneksel bilgisayar programlaması, komutların ikili ayrımına dayanır; bir tarafta veriler, diğer tarafta programın kendisi vardır (Bilginoğlu,1993). Veriler, somut durum için çözümü deterministik hale getiren, duruma özgü bilgilerdir. Program kısmı ise, çözümü gerçekleştiren prosedürleri tanımlar. Zeki programların yapısal özelliği ise, problem konusundaki bilgiyi, bir bilgi tabanı şeklinde programın üçüncü kısmı olarak oluşturmasıdır. Böylece, sonuç çıkarmaya yarayan muhakeme kısmı diğer iki kısımdan kesin olarak ayrılmaktadır (Şekil 1.1).



Şekil 1.1. Geleneksel Veri İşleme ve Yapay Zeka

Bilgi tabanı ve çıkarım mekanizması uzman sistem mimarisinin temel taşlarını oluştururlar. Bilgi tabanı, ilgili alan bilgisini içerir. Her uzman sistemin çalışmak için bilgi tabanına ihtiyacı vardır. Bilgi tabanı, uzmanın öğrenerek kazandığı görüşlerinden, hipotezlerinden ve sezgilerinden oluşan bilgileri tanımlar. Bir bilgi tabanı iki tür bilgi içermektedir; durumlar ve kurallar. Durumlar, spesifik bir alandaki değişik olayları göstermektedir. Kurallar ise, daha önceden belirlenmiş olan sezgilerin bir türüdür. Bu bilgiler IF....THEN şeklinde kurallar halinde toplanır. Kurallar her zaman iki bölümlüdürler; ilk bölümde "IF" ifadesi ile aranılan bir şart, ikinci bölümde ise ilk bölüme bağlı "THEN" ifadesine bağlı bir sonuç bulunur.

Bu toplanmış ve hafızada saklanmakta olan bilgileri kullanarak problemlere çözümler üreten ise, çıkarım mekanizmasıdır. Çıkarım mekanizması, bir karar ya da çözüm bulana kadar mevcut bilgiler ve kurallar arasında araştırma yapan genel problem çözme stratejilerini içerir. Çıkarım mekanizması bilgi tabanındaki özel bilgiyi kullanıp mantık yürütme işini çeşitli yöntemlerle yapabilir. Kullanılan yöntem, çıkarım mekanizmasına adını da verir. Bunların en önemlileri, ileriye doğru zincirleme ve geriye doğru zincirleme yöntemleridir.

Uzmanlardan alan bilgisinin elde edilmesi oldukça zordur ve bugün uzman sistemlerin geliştirilmesinde bu konu en büyük darboğazı oluşturmaktadır. Bilgi tabanının oluşturulmasında kullanılan bilginin çeşitli kaynakları bulunmaktadır. Ancak bilginin gün geçtikçe artması ve uzman sistem yazarken ulaşılan bilginin daha üst düzeyinin bulunması, bilgi elde etmenin kısa dönemli bir faaliyetin ötesinde dinamik bir süreç olmasını gerektirmektedir. Bilginin elde edilebileceği kaynaklar, insan uzmanlar, genel veri tabanı, matematik modeller, simülasyon programları ve makina öğrenmesidir.

Genel olarak uzmanlardan elde edilen bilgi, kuralları belirler. Hangi durumlarda nasıl davranılacağı ya da bir ara sonuca gitmek için ne gibi bir yol izleneceği, uzmanlardan öğrenilir. Genel veri tabanından bilgi edinilmesi arayüzler vasıtasıyla olur ve

kısa dönemli bilgilerin öğrenilmesi sağlanır. Matematik modeller ve simülasyon, çeşitli senaryoların incelenerek hangi kuralın daha iyi sonuç vereceğinin bulunmasını sağlar.

Uzman sistemlerin diğer programlardan farklı olan bir özelliği de, açıklama modülünün bulunmasıdır. Açıklama modülünden kasıt, kullanıcıya çeşitli yardımların verilmesi, kelimelerin ve soruların açıklanması olduğu kadar uzman sistemin çıkardığı sonucu nasıl ve neden çıkardığını da açıklayabilmesidir.

Öğrenme elemanı, yeni bilgilerin otomatik olarak veya el ile kazanılmasını sağlar.

Son zamanlarda uzman sistemlerin geliştirilmesinde uzman sistem kabukları denen sistemlerden istifade edilmektedir. Bunlar, hazır hale getirilmiş, çıkarım mekanizması ve bilgi saklama özellikleri ile donatılmış sistemler olup, sadece alan bilgisi olmayan içi boş uzman sistemlerdir. Ayrıca, kullanıcının kendisinin özel çıkarım mekanizmaları geliştirmesine imkan veren ve kabuklardan daha gelişmiş sistemler de vardır. Bunlara geliştirme çevreleri denmektedir. Uzman sistemlere ait bu elemanlar Şekil 1.2. 'de görülebilir (Bilginoglu,1993).

1.4. UZMAN SİSTEMLERİN AVANTAJLARI

Bir uzman sistemin istenmesi, öncelikle bulunabilirliğine ve uygun olmasına bağlıdır. İnsanlardan farklı olarak, kompüterize edilmiş bir uzman sistem hiçbir zaman bilgisini beraberinde götürmez, yılın her gününde kullanıma hazırdır. Uzman sistemdeki bilgi kolayca saklanabilir ve kopya edilebilir.

Bir uzman sistemin insan uzmanlara nazaran diğer bir avantajı, kompüterize edilmiş uzman sistemin performansının zirvesinde olması, bilgisinin sınırları içerisinde daima mümkün olan en iyi fikri, düşüncüyü, tavsiyeyi üretmesidir.

Ekipman kullanımının artması
 Personel sayısının veya uzmanlık seviyesinin azalması
 Finansal kayıpların azalması
 Gelirlerin artması
 Kolay bakım

ÖLCÜLEMİYEN YARARLAR

Problemlerin daha iyi anlaşılması
 Yöneticilerin ve uzmanların etkinliklerinin artması
 Daha iyi müşteri servisi
 Kullanışlı bir eğitim aracının ilavesi
 Daha iyi ve hızlı yönetim kararları
 Teknolojinin daha iyi anlaşılması
 Kararlılık
 Bilgi akışının gelişmesi
 Rekabet kabiliyetinin artması
 Daha etkin pazarlama ve satış
 Mevcut sistemle daha iyi entegrasyon

1.5. UZMAN SİSTEMLERİN KULLANIM ALANLARI VE SINIRLARI

Uzman sistemler, birçok değişik problem tipini çözmek için tasarlanmışlardır. Temel faaliyet alanlarını aşağıdaki gibi sıralayabiliriz:

- **YORUMLAMA:** Durum tanımlamalarını gerçekleştirmek için, sezgilerden hareket ederek hazırlanan verileri kullanırlar. Yorumlama sistemleri problem durumunun sembolik olarak gösteriminden çok, gerçek data ile çalışırlar. Örneğin kimyasal işlem yapan bir fabrikada, prosesin durumunu çıkarmak için ölçüm sonuçlarının okunması işleminin yorumlanması.

- **TAHMİN:** Verilen durumların muhtemel sonuçlarını çıkarırlar. Tahmin sistemleri, özel giriş verilerinden olası durumları üretmek için, gerçek olayları ve durumları yansıtan programlar olan simülasyon modelleri kullanırlar. Örnek olarak, belli bir makina arızası haberi alınarak, bunun sonucunda üretim hattı boyunca nerelerde hata veya üretim aksamasının olabileceğinin tahmini ve böylelikle tedbirler alınmasını sağlamak, verilebilir.
- **TEŞHİS:** Teşhis koyan uzman sistemler, sistem arızasının olası neticelerini çıkarmak için durum açıklamalarını, davranış karakteristiklerini ya da parça tasarımı hakkındaki verileri kullanırlar. Teşhis sistemleri sadece programı teşhis etmekle yetinmeyerek iyileştirmeye de yardımcı olmaktadır.
- **TASARIM:** Bir grup problem kısıtlamasına dayanarak, amaçların çeşitli konfigürasyonlarını geliştirirler. Örnek olarak esnek üretim sistemlerinin tasarımı ve yerleşimleri düşünülebilir. Tasarım sistemleri, genellikle tasarım fikirlerini test etmek ya da doğrulamak için simülasyon kullanırlar. Çünkü tasarım, planlama ile sıkı bir şekilde bağlantılıdır. Birçok tasarım sistemlerinde arzu edilen tasarımı elde edebilmek için planlar oluşturup geliştirecek mekanizmalar hazırlanır. Tasarım sistemi arzu edilen konfigürasyonların oluşması için planlar yaparak ve bu planları, problem kısıtlarının belirlediği şartlarda değerlendirerek birçok gereksiz araştırmayı saf dışı edebilir.
- **PLANLAMA:** Amaçları başarmak için plan geliştirme işlemidir. Örnek olarak belirli sürede üretilmesi gereken mamuller için üretim programının hazırlanması olabilir. Planlama sistemleri genellikle geriye dönüş yapabilmelidir. Yani problem sınırlarını aştığı için, problemin bir bölümünü reddedip, analizin yeni bir araştırmaya başlayabileceği bir noktaya dönüş yapabilmelidir. Bu nedenle bazı planlama sistemleri planlama işini alt problemlere böler ve çözüm bulunamayan seçim noktalarından hareketle yeniden planlamayı sağlamak için bu problemleri sıralamaya çalışırlar.
- **EĞİTİM:** Belirli bir konuda sahip oldukları uzmanlık bilgilerini sistemli bir şekilde bu konuda temel bilgilere sahip olan kullanıcılara aktararak, birçok alternatifi

incelemelerini sağlarlar. Bu gün bir uzman sistem, teorik olarak yapılan mühendislik eğitiminde öğrencilere, gerçek hayatta gerçekleştirme imkanı olmayan fikirlerini denemelerine ve sonuçlarını görerek konuları daha iyi bir şekilde kavramalarına yardımcı olmaktadır.

- **KONTROL:** Çeşitli üretim proseslerinde operasyon parametrelerinin öngörülen sınırlar içinde tutulması amacıyla, bu parametrelere etki yapan faktörlerin değişimlerinin kontrol altına alınması ve değerlerin istatistiklerinin tutulması işlemidir.
- **HATA GİDERME:** Yanlış fonksiyonlar için çareler önermedir.

Yukarıda belirtilen fonksiyonlara göre uzman sistem, sadece basit uygulamalar için doğru sonuç verecekmiş gibi görünmektedir. Bazı kompleks uygulamalarda, uzman sistem yazılımını anlamak zordur. Uzman sistemin yayılmasını yavaşlatan diğer faktörler şunlardır (Bonnet et al, 1986):

1. Bilginin her zaman hazır olarak mevcut olmayışı
2. Uzman bilgilerinin elde edilmesindeki güçlükler
3. Aynı konu uzmanlarının farklı bakış açılarına sahip olmaları
4. Uzman sistemlerin dar alanlara yönelik oluşları
5. Üst yönetimlerin tutuculuğu
6. Uzmanların hepsinin objektif ve bağımsız denetimlerinin olmayışı
7. Ortak terminolojinin yetersizliği
8. Maliyet yüksekliği ve hazırlanış sürelerinin uzunluğu

Yüksek seviyede deneyimli bir uzman için dahi, zaman baskısı altında iken durum hakkında saptamalarda bulunmak zordur. Nadir olarak bulunan ve çoğunlukla pahalı olan bilgi mühendisine ihtiyaç duyulur ki, bu kişiler uzman sistem geliştirmeyi oldukça pahalı yaparlar.

Uzman sistemler iyi çalışmasına rağmen, insan uzmanın yapay olandan ayrıldığı önemli alanlar vardır. Fakat bu, yapay zekanın bir sınırlamasını değil, sadece yapay zekanın mevcut durumunu gösterir (Tablo 1.1).

Tablo 1.1. İnsan Uzmanlığı ile Yapay Uzmanlığın Karşılaştırılması

<u>İnsan uzmanlığı</u>	<u>Yapay uzmanlık</u>
- Yenilikçi	- Yaratıcılıktan yoksun
- Uyumlu	- Açıklamaya ihtiyaç duyar
- Hissetme yeteneğine sahip	- Sembolik girdi
- Geniş alanlı	- Dar alanlı
- Tecrübi bilgiye sahip	- Teknik bilgiye sahip

1.6. UZMAN SİSTEMLERİN UYGULAMA ÖRNEKLERİ

İlk uzman sistem uygulamalarının işletmecilik konusunda olmadığı bilinmektedir. Burada, yetmişli yılların ortasında Stanford Üniversitesi'nde geliştirilmiş olan tıbbi tanı programı **MYCIN** anılabilir. Sistem, menenjitin ve buna yakın bakteriyel hastalıkların tanı ve tedavisinde doktor kararlarını desteklemekteydi. Uzman sistemler konusundaki bir çok gelişmeye bu sistem öncülük etmiştir. MYCIN, LISP programlama diline dayalı olarak yazılmıştı ve 400 kuralı içermektedir. Sistemde sağlıklı bir bilgilenmenin işlenmesi yanında bilgi güvenlik faktörü yardımıyla, bilginin kesinlik derecesi bilinmekte ve kesin olmayan sonuçların ayıklanması sağlanmaktadır. O yılların teknolojisi ile 5 yılda ve 100 adam-yıllık maliyetle gerçekleştirildiği belirtilmektedir. (Bilginöğlu, 1993)

İşletmeciliğin her dalında kullanım alanı bulan uzman sistem uygulamaları, finans konusunda da verimli çalışmaların yapılmasını sağlamıştır. Finansal kurumlarda kullanılan uzman sistem uygulamaları içerisinde sık sık tekrarlanan, klasik programlama dilleri ile çözülemeyen ve belirli bir karar verme özelliğinin gerekli olduğu işlemler önemli yer tutmaktadır. Kredi değerlendirme, kredi kartlarına harcama yetkisinin verilmesi, portföy yönetimi finansal kurumlardaki başlıca uygulama alanlarındandır.

Japonya başta olmak üzere çeşitli ülkelerdeki finansal kurumlar, uzman sistemler alanında zengin ürünler hazırlamışlardır. Mitsubishi Grubu, yapay zeka ve uzman sistem uygulamalarında liderliğini sürdürmektedir. Bu grubun içinde yer alan Mitsubishi Bank ve Mitsubishi Research Institute arasında yapılan işbirliği sonucunda,

- Yabancı para değerlerinin takibi ve tahmini
- Yatırım danışmanlığı
- Kredi yönetimi
- Müşteri değerlendirme

gibi konularda projeler geliştirilmiştir.

A.B.D' de kullanılan uzman sistem uygulamaları içerisinde American Express tarafından kullanılan **Authorizer's Assistant** en etkin çalışmalardan birisidir. Sabit kredi sınırlarının tanımlanmadığı American Express kredi kartlarında, kart sahiplerinin yapacağı harcamalar, Authorizer's Assistant uzman sisteminin yardımı ile onaylanmaktadır. Bu sistem yaklaşık 40 değişik veri bankasından birkaç saniye içinde topladığı

- Müşterinin son faturasını ödeyip ödemediği
- Kart sahibinin harcama profili
- Kart sahibinin daha önce bu boyutta bir alışveriş yapıp yapmadığı

gibi bilgileri ve kendi tavsiyesini karar vericiye aktarmaktadır. Yılda 27 milyon \$ tasarruf sağlayan sistem, zaman olarak işlemleri %25 oranında kısaltırken, hataları %30 oranında azaltmıştır (Bilginoğlu, 1993).

Endüstride kullanılan uzman sistemler, üretimin çeşitli aşamaları için geliştirilmiştir. Tasarım, planlama, üretim, dağıtım vb. süreçlerde kullanılan çeşitli uzman sistemler mevcuttur.

Tasarım alanında başarısı kanıtlanmış uzman sistemler olarak **XCON** (Expert Configurer) ve **XSEL** (Expert Selling Assistant) programları anılmaktadır. Üretici firma Digital Equipment'in ürün yelpazesi, 50 farklı merkezi işlem birimi ve 400'den fazla bellekleme seçeneğini içermektedir; bunların kombinasyonu milyonlarca konfigürasyon olanağı demektir. XCON müşteri siparişlerini uygunluk ve tamlik temeline göre denetlemekte, eksik önemli birimleri veya elemanları uyarmakta ve değişiklik önerileriyle ilgili diyagramlar vermektedir. Sistem 7,000 kuralı içermektedir ve 180 adam-yıllık maliyete katlanılmıştır (Simons, 1985). Aynı program paketinden geliştirilmiş olan XSEL, yerel satış elemanlarının desteklenmesini sağlamaktadır.

Ürün niteliklerini müşterinin isteklerine göre değiştirmek için kullanılan bir uzman sistem **ALADIN**'dir. ALADIN (Aluminum Alloy Design Inventor) sistemi, belirli niteliklere sahip alüminyum alaşımının tasarlanması sırasında mühendislere yardımcı olmaktadır. Sistem istenen niteliğe göre mevcut alaşıma çeşitli katkı maddeleri ilave edilmesini, işleme yöntemini ve istenen yapıyı sağlayacak mikro yapıyı sunabilir. **EL** sistemi, elektronik şema tasarımlarını analiz eden; **HI-RISE** sistemi bina tasarlamada ve inşaat işlerinde yardımcı olan; **VT** sistemi ise asansör tasarlamak için geliştirilen uzman sistemlerdir (Ünüvar ve diğerleri, 1995).

Üretim planlamada uygulanmak üzere geliştirilen ilk sistem, Digital Equipment Corporation tarafından kullanılan ve ISA (Intelligent Scheduling Assistant) olarak bilinen sistemdir (Frost et al, 1984). ISA orta vadeli planlama sürecinde çalışır. Siparişleri makinalara, sonlu çizelgeleme algoritmalarını hatırlatan sezgisel yaklaşımlar kullanarak yükler. ISA, çizelgeyi nasıl oluşturduğunu kontrol eden 300 kuralı içerir. Her birinin çizelgelenmesi 10 saniye süren 300 siparişi programlamak için kullanılmaktadır. Sistem, problem çözümünde tekrarlı formülasyon yaklaşımını kullanır. Örneğin ISA, verilen bir

siparişin istenen zamanda tamamlanmasını sağlayan programı bulamazsa, siparişin bir kısmının istenen zamanda tamamlanması çözümünü araştırır.

Fabrika çizelgelemede kullanılan bir başka uzman sistem, Carnegie Mellon Üniversitesi'nde geliştirilen ISIS (Intelligent Scheduling and Information System)'dir. Sistem, Westinghouse Electric'te her biri en az 10 işlem gören 200 siparişi çizelgelemek amacıyla kullanılmaktadır. Sistem, bir işlemin başlama ve bitiş sürelerini belirlemenin yanı sıra işlem için gerekli makinaların seçimini de yapmaktadır. Bu seçimi yaparken, makinaların işleme oranları, makinalar üzerindeki mevcut iş yükleri ve makinaların işlem kabiliyetleri gibi kısıtları dikkate almaktadır. Sistem özellikle, insan planlayıcıların zamanlarının %80 - %90'ını alan problem kısıtlarının belirlenmesi işine yardımcı olmaktadır (Kanet et al, 1987).

1.7. UZMAN SİSTEMLER İÇİN TEMEL TERMİNOLOJİ

Alan uzmanı: Yılların eğitimi ve deneyimi ile belirli bir konuda problem çözmede son derece profesyonel olan kişidir. Uzman sistem geliştirme sürecinde bilgi mühendisi ile sürekli etkileşim içindedir.

Son kullanıcı: Tamamlanmış uzman sistemi kullanan kişidir. Bu kişi bilgi mühendisi, alan uzmanı veya herhangi bir kullanıcı olabilir.

Bilgi mühendisi: Yapay zeka alanında ve bilgisayar biliminde bilgi sahibi olan, uzman sistemin nasıl yaratılacağını bilen kişidir. Bilgi mühendisi insan uzmanlardan süreçlerini, stratejilerini, mesleki uygulama kurallarını öğrenerek, bunları uzman sistem içine aktaran kişidir.

Uzman sistem kurma aracı: Uzman sistem oluşturmak için kullanılan programlama dili ve destek çevredir.

Destek çevre: Uzman sistemle kullanıcı arasındaki etkileşime yardımcı olan, bir uzman sistem kurma aracı ile birlikte bulunan yardımcı araçlardır. Bunlar kompleks hata ayıklama yardımcıları, düzeltme programları ve ileri grafik araçlarını içerebilir (Beerel, 1987).



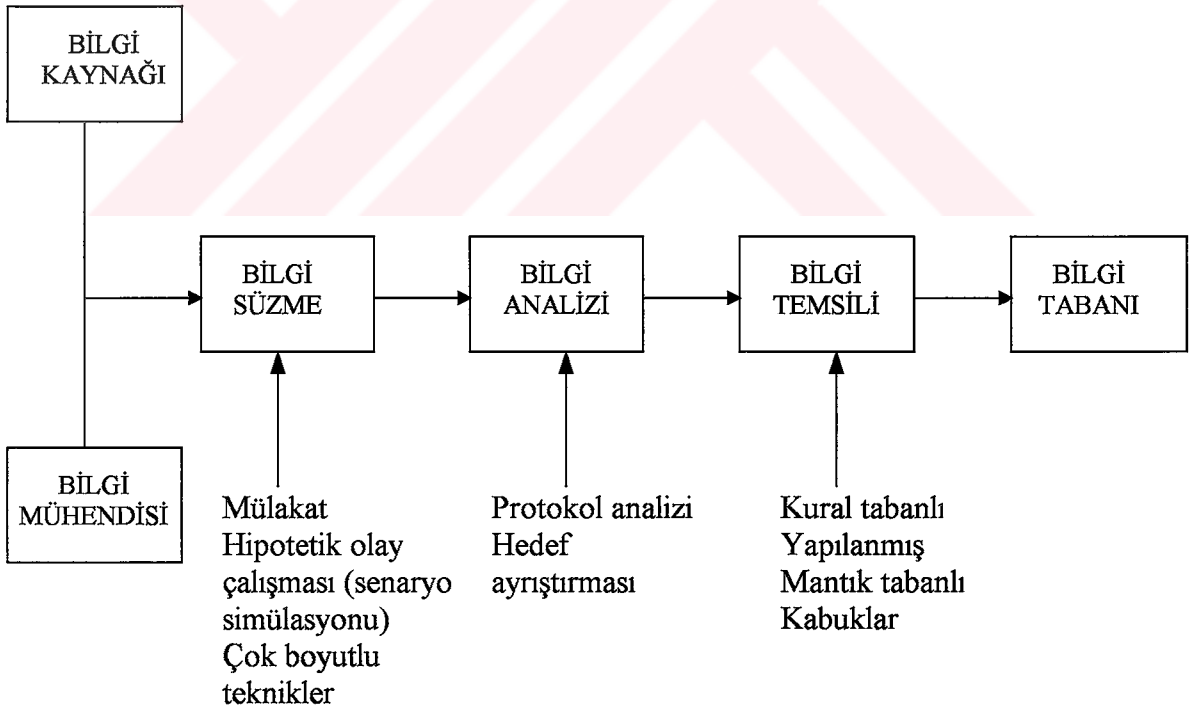
2. BÖLÜM

UZMAN SİSTEMLERDE BİLGİNİN ORGANİZASYONU

2.1. GİRİŞ

Uzman sistemlerde bilginin organizasyonu prosesi 3 safhaya ayrılabilir (Şekil 2.1):

1. Bilgi süzme: Uzmardan bilgiyi sağlama prosesi.
2. Bilgi analizi: Birinci safhada elde edilen bilginin anlaşılabilirliğini sağlama prosesi.
3. Bilgi temsili: Analiz edilmiş bilginin, uzman ve bilgi mühendisi arasındaki iletişimin geliştirilmesi için anlaşılabilir ve kullanışlı bir formda ifade edilmesi prosesi (Chan, 1995).



Şekil 2.1. Bilginin Organizasyonu Prosesi

2.2. BİLGİ SÜZME

Uzman sistem, bilgilerin kompüterize sistemin içinde birleştirilmesi için uygun bir biçimde alınmasına ve gösterilmesine ihtiyaç duyar. Bu işlem, bilgiyi süzme olarak bilinir. Bilgi mühendisi, giriş sisteminde ağırlıklı olarak personel mülakatlarına güvenir. Fakat bazı önemli engeller vardır ve bunlara dikkat etmekte fayda vardır.

1. Uzmanın alanı çoğunlukla tekniktir ve bilgi mühendisi konuya aşina olmadıkça anlamada ve sunmada zorluk çekecektir.
2. Bilgi mühendisi konuyla ilgili araştırmalar yapar. Fakat sezgisel tecrübeler uzmanlar tarafından elde edilir ve bu sebeple uzman burada çok kıymetlidir.

Uzmandan bilgi toplama aşamasında ona hipotetik olay çalışmaları verilir bunları nasıl çözeceğini, hangi gerçekleri kabul ettiğini açıklaması istenir. Problemler akıllıca seçildiğinde bu metot çok avantajlıdır. Genellikle her bir problem hakkında bilgi sağlamak amacıyla şu aşamalar gerçekleştirilir.

- Mümkün olan sonuçların tesbiti ve kuralların kullanılması dahil olmak üzere, problemin çözülmesi için konunun sınırlarını çizmek amacıyla soru sormak.
- Belirlenen kuralların her birinin nasıl spesifik, nasıl genel olacağını belirlemek.

Bilgi mühendisinin tipik soruları şunlar olabilir:

- Ne zaman yaparsın?
- Niçin yaparsın?
- Bu kurallar hangi koşullar altında uygulanacak?
- Bu kurallar hangi koşullar altında uygulanmayacak?
- Nasıl kurdun?

Uzmandan bilgi elde edilmesinde kullanılabilecek diğer teknikler şöyledir:

- Çok boyutlu teknikler: Aralarında kompleks bir etkileşim varolan elemanların bilgi alanının kuruluşuna dair plan sunarlar. Bunlar bir üçüncüden farklı veya benzer olan iki şeyin tespit edilmesini gerektirir.
- Otomatik bilgi sunma ve elde etme: Gelecekte bir uzman sistem, verilen örnek setlerinden özel durum bilgi alanına uygulanabilir kuralları ve genel prensipleri direkt olarak ayırt edebilecektir. Bu günkü uzman sistemlerde tümevarım teknikleri sonuç çıkarımı için kullanılmaktadır. Bazı uzman sistem kabukları bu tümevarım kolaylığını kullanabilirler. Kabuklar tipik olarak uzmanın uygulanabilir bir konuda anahtarları girmesiyle çalışırlar, kabukların otomatik tümevarım modülü çağrılır ve kural setleri işlemeye başlar. Bunun için standart bir algoritma vardır. Tablo 2.1' de personel seçimi için kullanılan bir örnek verilmiştir. Dokuz özel durum, çıkarsama yapılan dört kurala öncülük eder.

Tablo 2.1. Bir Eleman Seçimi İçin Kural Yönetimi

DURUM	DERECE	KALİFİKASYON	TECRÜBE	KARAR
1	1	EVE	1 YIL	TEKLİF
2	3	EVE	2 YIL	RED
3	2	EVE	2 YIL	TEKLİF
4	1	EVE	*	TEKLİF
5	2	EVE	2 YIL	TEKLİF
6	*	HAYIR	*	RED
7	2	EVE	3 YIL	TEKLİF
8	3	HAYIR	3 YIL	RED
9	3	*	*	RED

* Bilinmiyor

IF DERECE=1 AND KALİFİYE THEN TEKLİF

IF DERECE=2 AND KALİFİYE AND TECRÜBE=3 OR TECRÜBE=2 THEN
TEKLİF
IF DERECE=3 THEN RED
IF NOT KALİFİYE THEN RED

2.3. BİLGİ ANALİZİ

Bilgi analizi safhası uzman sistem gelişimi için önemlidir, çünkü bu safhanın sonucu, problem alanının kavramsal modelidir. El kitaplarından, uzmanlardan, kullanıcılardan elde edilen veri, kullanılmadan önce bilgiye dönüştürülmelidir. Üretilecek uzman sistemin etkinliği, direkt olarak toplanacak bilgiye dayanmaktadır.

Analiz adımı üç safha arasında en zor olanıdır, çünkü uzmanın yorumları ile aklındaki gerçek bilgiler arasında bire bir eşleşme olduğunu düşünmek yanıltıcıdır. Bilgi, uzmanın yorumlarından çıkarılacaktır, bu yüzden yorumların uzman bilgisine tercüme edilmesi başarılı sistemlerin inşası için önemlidir. Analiz safhasında kullanılabilecek teknikler şöyledir:

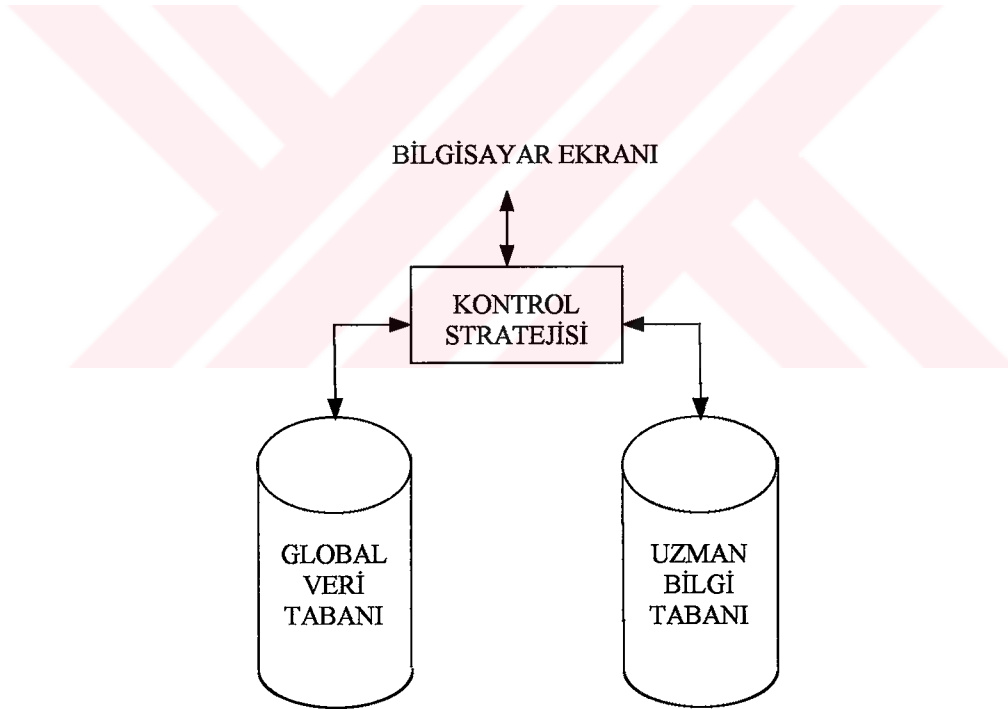
1. Protokol analizi: Bilgi mühendisi, ses ve video teypleriyle uzmandan görevlerin işlemesi yönünde gözlemler alır. Bu gözlemler anahtar değişkenler ve kurallar için araştırılır ve problem çözülürken uzmanın yaklaşım sırası esas alınır.
2. Hedef ayrıştırması: Bu tip analiz çok uygun teknikler içerir ve bilgi bir hiyerarşi içinde şekillendirilir. Uzman, hipotetik problem setine sahip bir mülakatçı ile karşılaştırılır. Problem hakkında mümkün olduğu kadar çok soru ile bir bilişim kurmaya çalışılır. Soru tipleri bilgi mühendisi tarafından bilgi alanının yapısına yerleştirilir. Bu teknik çalışma alanına aşina bir mühendis tarafından kullanılabilir.

2.4. BİLGİ TEMSİLİ

Uzman sistemlerin geliştirilmesindeki temel problemlerden bir tanesi de, uzmanların sahip olduğu ve kullandığı bilginin nasıl temsil edilip kullanılacağıdır. Tipik bir uzman sistemde bilgiler, aşağıda gösterildiği gibi üç kategoride temsil edilebilir:

1. Belirli bir problemi tanımlayarak girdi verilerini ve mevcut çözüm statülerini ve durumlarını izleyen global bir veri tabanı.
2. Genel bir problem kontroluyla ilgili olayları ve sezgisellikleri tanımlayan bir bilgi tabanı. Bu kontrol bilgisi, kurallar formu şeklinde olduğundan bazen prosedüre dayalı veya alan bilgisi olarak tanımlanır.
3. Problem çözüm yaklaşımını tanımlayan veya problemi çözmek için verilerin ve bilgilerin nasıl kullanılacağını gösteren bir kontrol veya sonuç düzeyi.

Bu nedenle uzman sistemler tipik olarak, veri ve dil yapılarının oluşturulmasını sağlamaktadır. Bu yapılar, problem verisinin, kontrol edilebilir spesifik problem çözme bilgisinin ve bu bilginin işletilmesi için kontrol yapılarının (sonuç veya prosedüre yönelik) saklanması ve temsil edilmesi için kullanılmaktadır (Şekil 2.2).



Şekil 2.2. Uzman Sistemlerde Bilginin Organize Edilmesi

2.4.1. GLOBAL VERİ TABANI

Veri tabanındaki girdi, çözülmekte olan belirli bir problem ile ve mevcut olay durumlarıyla ilgili veri veya olaylar formundaki açıklayıcı bir bilgidir. Açıklayıcı bilgiyi sunmak için bir çok alternatif yol vardır. Burada uygun sunuş formunun seçilmesi uygulamaya bağlı olmaktadır. Temel ilgi odakları, etkin saklama, yeniden kullanma ve modifikasyonun kolaylığı konuları üzerinedir. Halen kullanılan metotlar aşağıda açıklanmıştır.

2.4.1.1. KURAL TABANLI BİLGİ TEMSİLİ

Uzman sistemlerin bir avantajı, bilgi tabanına bir çok kuralın eklenebilmesidir; buna, artan bilgi kazancı da denir. Bu inanın doğru olmadığı daha sonraları ortaya çıkmıştır. Mantık programlarında bile bilgi tabanındaki kuralların olduğu sıra, ulaşılacak sonuçları etkileyebilir. Bu düşüncenin çok önemli olması, bir çok sahanın yapıyı kaybetmesi ve çok karmaşık olmalarıdır. Bu karmaşıklığın nedeni çok fazla parametre ve değişkenin bulunmasıdır. Karar boşluğu potansiyel olarak çok geniştir. İnsan uzman tarafından kontrol edilen işlem, bir avuç dolusu değişken olarak tanımlanabilir. Dolayısıyla karar boşluğu iyi sınırlandırılmıştır. Her bir kuralın sol kısmı, özel bir cümlecik kümesi içerir ve hangi cümleciklerin içerileceğine dair direkt bir kontrole sahiptir. Her değişkenin alabileceği değerlerin sırası önceden belirlenebilir. Bu, iki şey sağlar:

1. Her yeni kural, mevcut bulunan başka bir kuralla üst üste çakışıyor mu diye kontrol edilebilir,
2. Bilgi tabanı, hiçbir karar alanının kaplanmamış olmaması açısından tamamlanmışlık için kontrol edilebilir. Kuralların sol kısmının uzun ve karmaşık hale gelmesi mümkündür, ama eğer böyle bir durum meydana gelecek gibi görünüyorsa, kuralları yapılar haline getirmeyi veya bloklara ayırmayı denemekte fayda vardır.

2.4.1.2. YAPILANMIŞ BİLGİ TEMSİLİ

Yapay Zeka programları, insanoğlunun davranışlarına bir bakıma benzemek amacını güder. Programın kullandığı bilgi, insanın kullandığı bilgiyi andırmalıdır. Kurallar insan beyninin bilgiyi stoklama yolunun iyi bir temsili olmayabilirler. Kural tabanlı format, insan aklının içindekileri modellemeyi gütmmez, bunun yerine bilgisayarı insana benzer davranışlar üretmek üzere programlar.

Yapılanmış sistemde bilgiyi göstermek için 3 ana kalıp vardır (Efsthathiou, 1989).
a- Şebekeler, b-Çatılar, c-Nesneler.

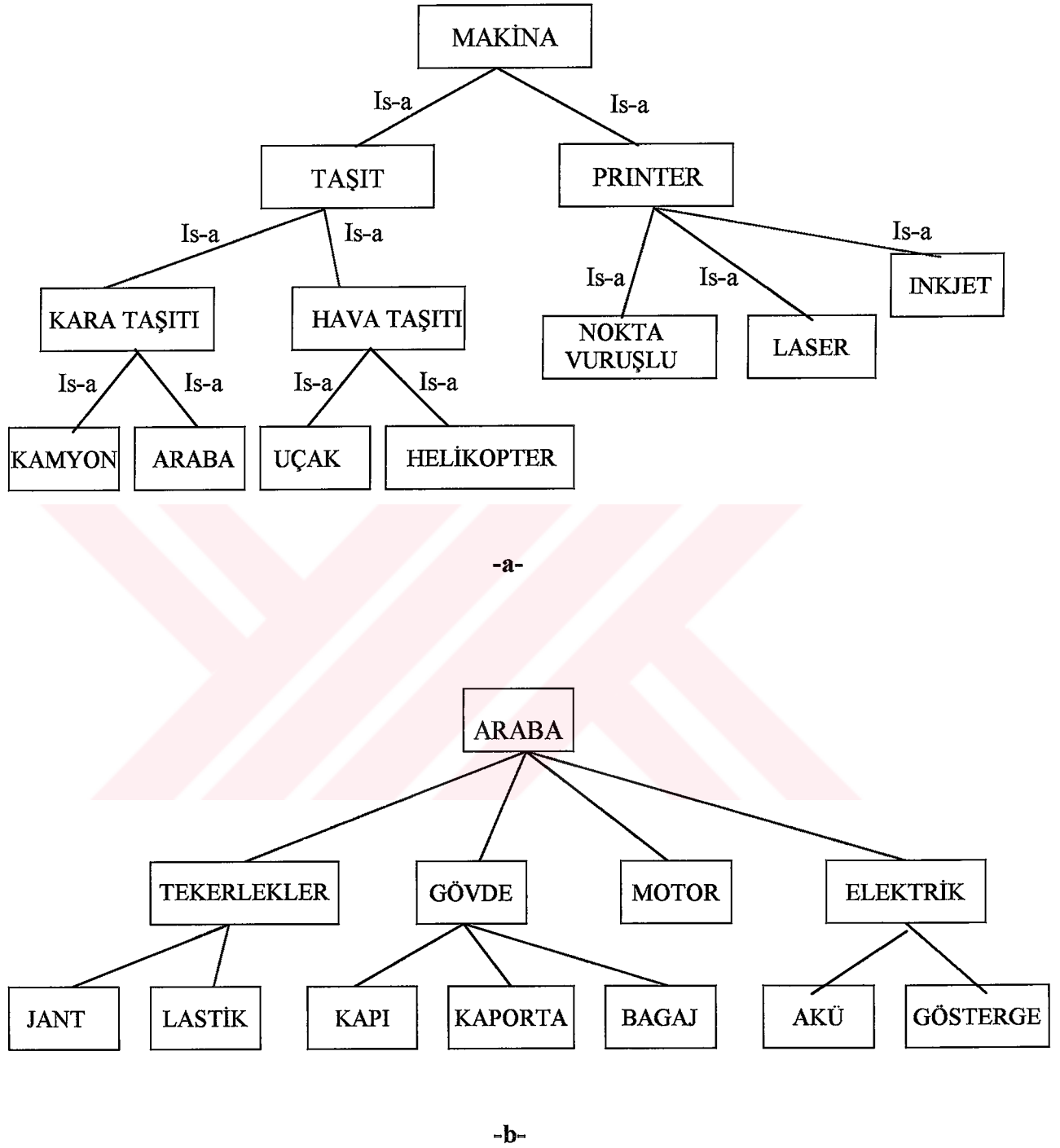
a- Şebekeler

Burada ifade edilen şebekeler matematiksel olanlardır. Bir şebeke, bağlarla birleştirilebilen düğümlerden oluşur. Düğümlere genellikle isim verilir. Şebekelerin özel bir çeşidi ağaçtır, bağların bir yönü vardır. Ağaçlar genellikle sınıflandırmaları göstermek üzere kullanılır, buradaki bağlar “is-a” veya “is-part” olarak bilinir (Şekil 2.3-a ve Şekil 2.3-b).

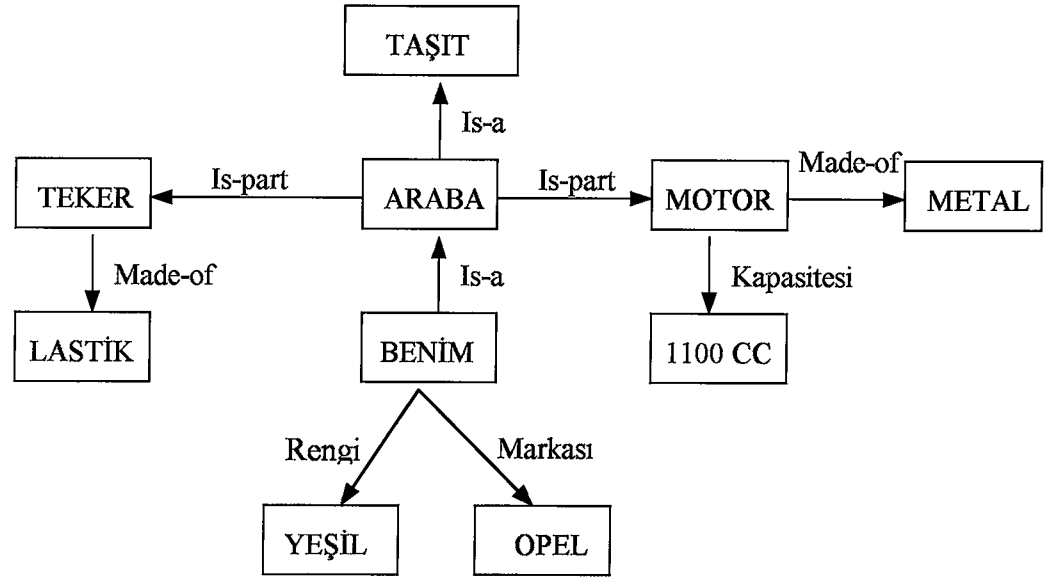
Bazı şebekeler network’ler oluşturmaları için döngülere sahiptirler, bazen düğümler arasında çok değişik bağlar bulunabilir (Şekil 2.4). Bu tür şebekeler anlam ağı adını alırlar, çünkü doğal dildeki kavramları temsil etmek için kullanılmışlardır.

b. Çatılar

Anlam ağları, insan aklında da görüldüğü gibi kavramlar arasında bir çok ilişkinin olduğunu göstermişlerdir. Ancak şebekenin hiyerarşik gösterim türü de önemlidir. Çatılar, veri tabanındaki bilgileri gruplandırmak için kullanılırlar. Çatılarla önce genel hayvanlar sınıfına, sonra kuşlar sınıfına, sonra kanaryalar ve kanaryaların örneğine ait bilgiyi temsil edebiliriz (Tablo 2.2) (Efsthathiou, 1989).



Şekil 2.3-a. Is-a ve Şekil 2.3-b. Is-part Hiyerarşileri



Şekil 2.4. Bir Anlam Ağı

Tablo 2.2. Çatıların hiyerarşisi

HAYVAN**Bir:** Canlıdır**Ayak sayısı:** 2, 4, 6, 8 ve daha fazla**Yaşadığı yer:** Delik, Yuva, İn**Renk:** Değişik**KUŞ****Bir:** Hayvandır**Ayak sayısı:** 2**Yaşadığı yer:** Yuva**Renk:** Değişik**Uçar:** Evet, Hayır**Göçmen:** Evet, Hayır

KANARYA**Bir:** Kuştur**Renk:** Sarı**Uçar:** Evet**TWEETY****Bir:** Kanaryadır**Yaşadığı yer:** Kafes**Düşmanı:** Sylvester

Çatılar günlük bilgi muhakemesini, diğer bir deyişle günlük hayatın varsayımlarını göstermek için kullanılabilirler. Bulanık küme teorisi iyi tanımlanmış ama insan bilgisini temsilde yeterli olmayan matematiksel yapıları gerektirir. Çatılar küme ve alt kümelerin kavramlarını takip etmezler, bunun yerine uzmanlaşma, örnekler ve istisnalar ile genel sınıfları izlerler. Çatıların insan hafızasına benzer olduğunu bazı psikolojik bulgular ispatlamaktadır.

c. Nesneler

Çatılar ilk olarak 1975'te öne sürülmüştür ve 10 yıl içinde büyük ilgi çekmiştir. Bundan sonra nesneye dönük sistemler hız kazanmıştır. Bunun nedeni, program gelişmesini orijinal modele uygun olarak desteklemeleridir. İlk yararlı nesneye yönelik sistem SMALLTALK'tur. Smalltalk Windows, mouse, pointer ve ikona ile "WIMP" ekran görüntüsünden yararlanmıştır. Nesnesel programlama, LISP ve prosedürel dillerin türevleri altında gereklidir. ACTOR ve C++, söz dizimi bakımından LISP dışındaki programlara daha çok benzeyen nesneye yönelik dillerdir.

2.4.1.3. MANTIK TABANLI BİLGİ TEMSİLİ

Şimdiye kadar, bilgi temsilineki yeni gelişmeler incelendi. Bu gelişmeler insanın muhakeme kabiliyetini model alarak, bilgisayar ve yazılım gücünü kullanarak temsil kalıplarını destekleme yolunu izlemişlerdir. Psikoloji, her yerde bu kalıpların insan aklına benzerliğine kanıt gösterir. Ancak bu teknikler sadece uzmanlar tarafından kullanılabilirler.

Buna karşılık, mantık temeline dayanan paradigma çok farklı bir anlayış içerisindedir. Mantık, kompleks yazılım kullanmak yerine, mantıksal aksiyomların çatılarını ortaya koyar. Bu çatılar, bilgiyi göstermek için bir dizi ilkel ifade gerektirirler. Mantığın matematiksel temeli, varolan sonuçların güvenilirliğini sağlar; çünkü bunlar doğru ve ispatlanabilir çatılarla ifade edilmişlerdir. Tüm sonuçlar mantık içinde bir teorem statüsü taşırlar. Mantık, çok çeşitli deyimlerin açıklandığı geçerli bir dildir. Burada, basit açıklamalar, “uygun formlu formüller” olarak adlandırılmıştır.

Örneğin “M1 makinası boştur.”,

1. MCH-IDLE (M1) {MAKİNA-BOŞ (M1) }
2. STATUS (M1, IDLE) {DURUM (M1, BOŞ) }

olarak ifade edilebilir. Bu şekilde problemler sembolik olarak ifade edilebildiğinden, yapay zekada oldukça kullanışlıdır. Mantığın gerçek gücü şöyle açıklanabilir. Normal lisandaki (Türkçe, İngilizce vb.) cümleler, bir bilgisayar ile işlenerek ve diğer bilgilerle karşılaştırılarak geçerli bir forma dönüştürülür. Bu bölümün geri kalan kısmında mantığın bugünlerdeki şekillerine kısaca değinilecektir (Efstathiou, 1989).

a. Yüklemler Mantığı

Yüklemler mantığı, dünyadaki olayları yüklemler kullanarak gösterir. Örneğin, “Socrates bir insandır” cümlesi, İnsandır (Socrates) olarak gösterilebilir. Burada

“insandır” yüklem ve Socrates de yüklemın öznesidir. Bütün insanlar hakkındaki ifadeler niceleyiciler kullanılarak gösterilirler.

“Bütün insanlar ölümlüdür” ifadesi iki yüklem yardımıyla gösterilebilir .

$$\forall x, \text{insan}(x) \Rightarrow \text{Ölümlü}(x)$$

($\forall$) sembolü evrensel niceliyicidir ve “her” olarak okunur. Yukarıdaki mantık formülü “Bütün x’ler için, eğer x insan ise ölümlüdür” şeklinde okunur. Evrensel niceliyiciden başka, tekil niceliyici olan “ $\exists$ ” de vardır. Bunu kullanarak “herkes birini sever” ifadesi,

$\forall x \exists y : \text{sever}(x,y)$ olarak gösterilebilir. Bu formül şöyle okunur: Her x için bir y vardır ki x, y’yi sever. “Sevmek” fiili iki kişiyi içerir yani ikili bir yüklemidir.

Yüklemler mantığı olayları göstermek için yeni bir yaklaşımdır, çünkü iyi tanımlanmış bir yapısı ve sonuçlandırma metodları vardır. Fakat temsil edilen olaylar bazen yapmacık görünebilir. Bir romandan veya kitaptan alınan cümlelerin yüklemler mantığı ile gösterimi zordur. Ancak tam bilgiyi bu mantıkla göstermek yararlı olabilir.

b. Şekilsel Mantık

Şekilsel mantık, mantığın “ $\Box$ ” ile gösterilen şekilselleşmiş bir biçimidir. “ $\Box$ ” genellikle “...gereklidir ki...” şeklinde okunur. Dolayısıyla, $\Box a$: “a’nın doğru olması gereklidir” şeklinde okunur. Benzer olarak $\Box \neg a$: “a’nın doğru olmaması gereklidir” şeklinde okunur.

Şekilsel mantık bilgi ve inanışlar hakkında muhakeme için kullanılan bir mantık haline dönüştürülebilir. Bu durumda “ $\Box$ ” “Biliyorum ki...” anlamına gelir. Bilgi hakkında çok değişik ve ilginç sonuçlar bu mantık içerisinde ispatlanabilir. Şöyle ki: $\Box a \rightarrow \Box \Box a$,

"a'yı biliyorum, sonra a'yı bildiğimi biliyorum" anlamına gelir. Bu, "Bildiğimi biliyorum" şeklinde genelleştirilebilir.

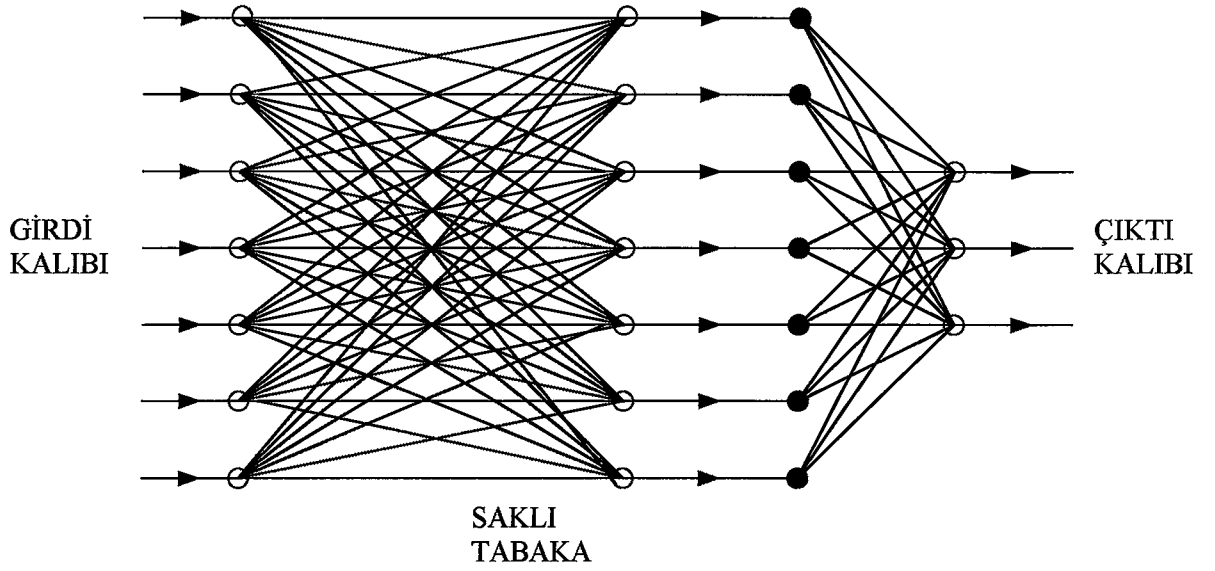
c. Sinir Ağları

Sinir ağları, bilgi gösteriminde sonraki safha olarak dikkati çeker. Kurallar, çatılar ve nesneler insan aklını bir şekilde modellemek içindirler. İnsanların nasıl yaptıklarını veya anladıklarını söylemeleri sonucunda kurulmuşlardır. Çoğu zaman düşünmeden davranırız: Bisiklet kullanmak, bir işlemi kontrol etmek için doğru kuralı veya bilgiyi araştırmayız. Kurallar bilgi hakkında konuşmanın yan etkileridir. Psikoloji, bilginin beynin belirli bir alanında bulunmadığını, fakat beynin ağı içerisinde depo edildiğini söyler. Sinir ağları beynin fiziksel yapısını, bilgiyi göstermek için ağı bağlarını kullanarak modellemeye çalışır.

Bir sinir ağı nöron denen elementlerden, nöronlar arasındaki bağlardan ve girdi/çıkış şemasından oluşur. Nöronlar arası bağlar, her bir nöronun başka bir kaç nörona, bir çok nörona veya ağıdaki tüm nöronlara bağlı olabileceği bir yapıya sahiptir (Şekil 2.5) (Efsthathiou, 1989).

Nöronlar, her bir nöronun bağlarından gelen bilgileri nasıl tercüme ettiğini anlatmak ve ağı geride kalan kısmına hangi sinyali nasıl dağıtacağına karar verişini göstermek için öğrenme kurallarına sahiptir. Çok farklı kurallar vardır. Nöronun önceki durumu, çeşitli başlangıçlar ve gelen sinyalleri toplamak için kullanılan fonksiyonlar gibi.

Sinir ağları çok değişik fonksiyonlarda kullanılabilir. "Birleştirici hafıza" formu, basit görüntü tanıma görevlerini yerine getirmek için kullanılabilir. Sinir ağları aynı zamanda optimizasyon problemlerine çözüm üretmek için de kullanılabilir, örneğin gezgin satıcı problemi gibi. n bölge için satıcı bu bölgeler arasındaki en kısa rotayı seçmeli ve her birine sadece bir kere uğramalıdır. Satıcının $n!$ kadar rota seçme olasılığı vardır. Bir sinir ağı, çabucak optimal çözüme yakın bir çözüm bulabilir.



Şekil 2.5. Bir saklı tabakalı basit bir sinir ağı

2.4.2. BİLGİ TABANI

Bu bilgi tabanı, belirli bir tipteki problemin, sorun veya alan bilgisi özelliğini tanımlar. Bu bilgi, genelde prosedüre yöneliktir ve bir problemin verilerinin, problemin çözümü için nasıl kullanılacağını açıklar. Bir uzman sistemin gücü, problem alanındaki spesifik bilgiye dayanmaktadır. Bir çok uzman sistem, insan uzmanların sahip olduğu bilginin araştırılması ve daha sonra da bu bilginin kural ağı oluşturulmak üzere birleştirilmesi ile elde edilen yüzlerce kuralı içermektedir.

2.4.3. KONTROL STRATEJİSİ

Kontrol stratejisinde bilgisayar programı, cevaplandırılacak sorunun ne olacağı hakkında ve daha sonra mevcut problemi çözmek için veri tabanının ve bilgi tabanının ne şekilde kullanılması gerektiği hakkında karar verir. Kontrol stratejileri sabit veya deneysel olarak sınıflandırılabilir. Sabit kontrol stratejisinde, bir uygulama kuralı seçilir ve

daha sonraki kabuller için bir hazırlık yapmadan uygulanır. Deneysel kontrol stratejilerinde ise, bir kural seçilir, bu kural uygulanır. Ancak tatmin edici bir çözüm bulunamadığı takdirde (çözüm yoksa), farklı bir kuralın uygulanması için, bu hesaplama noktasına daha sonra dönmek üzere bir çalışma yapılır.

Bir kontrol komut üretici, problemin çözülmesinde göz önüne alınan adımları organize ve kontrol eder. Gerekli bilgi ve yazılım modüllerini alarak uygun sıra altında yazılım modüllerini işletir, çıktıyı analiz eder ve uygun bir formda cevaplar verir. Buradaki adımlar veya kurallar, IF... THEN kurallarının birbirine bağlanması ile oluşan modellerle gerçekleştirilir. Uzman bir sistem sınırlı bir ustalık alanı dahilinde problemleri çözmek için bilgiyi kullanır. Genellikle uzman sistemlerle bağdaştırılan bilgi simgesi kalıbı IF ... THEN'dir. Bu bilgi, sonuç çıkarmak ve diğer görevleri yerine getirmek üzere yapılan programlardaki değişiklikler için bilgisayara yüklenmelidir. Bilgi, dünyadaki çok çeşitli formlardan ve kaynaklardan toplanacaktır. Bütün bunlar öyle bir biçimlendirilmelidir ki makina için kullanılabilir, aynı zamanda kullanıcı için de anlaşılabilir bir form ortaya çıkmalıdır. En basit uzman sistemler uğraştıkları bilgi sahasında oldukça kısıtlıdırlar ve ilkel muhakeme gücüne sahiptirler. Bilgiyi temsil etme yetenekleri de sınırlıdır. Uzman sistemlerin avantajları ve yararları, onların geniş hafızalarından ve sürekli olarak performans gösterebilmelerinden kaynaklanmaktadır. Uzman sisteme yerleştirilebilecek bilginin sınırlı olduğuna dikkat etmek gerekir; bu sınır sadece bilgisayarın hafıza kapasitesi değil, aynı zamanda kullandığı bilgi temsili kalıbıdır. Bazı bilgi çeşitleri temsile çok daha iyi uyarlar, öyle ki temsil, kullanılan sembolleri ve bu işlem sırasında bilgiyi şifreleyen lisana olan açıklığı yönünden oldukça verimlidir. Bir şehir haritasını düşünelim. Harita, kullanıcılara şehir etrafında bir yerden diğerine giderken kolayca bir rota planlamalarını sağladığı için bir bilgi kaynağıdır. Aynı bilgi bir tablo veya geniş bir IF ... THEN kurallarıyla sunulabilir. Ancak bunun kullanılması çok güçtür ve haritanın tanıdık görsel görüntülerinin eksik olması durumunda, kâğıt ve kalem kullanmadan alternatif rotaların kurulması ve karşılaştırılması mümkün olmayacaktır.

Bilgi temsili kalıbının seçilmesi çok önemlidir, çünkü bu bir dereceye kadar uzman sistemin içine sığdırabileceği bilgi çeşitlerini belirler. Bazı bilgi türleri kurallar tarafından çabuk ve hızlı bir şekilde temsil edilebilir. Ama eğer, diğer bilgi türleri uzman sistemin yaşama sürecinde önemli olmaya başlarsa, bilgiyi mevcut olan kalıba zorla sokmak yerine yeni bilgi temsili kalıpları ortaya koymak yararlı olacaktır. Böylece IF ... THEN kurallarının işlem kontrol uygulamalarına çok uyacaklarını söyleyebiliriz. Uzman sistemden istenen sonuç şekilleri anlaşılmadan bilgi temsil kalıbı seçiminin yapılması mümkün değildir. Sistemi oluşturan kişi bu sonuçların seçilen yolda temsil edilen bilgiden verimli bir şekilde elde edilebileceğini garanti edebilmelidir. Bütün bunlardan başka bilgi, bir mühendis tarafından kalıba dönüştürülmelidir, böylece kalıp anlaşılabilir bir hale gelir veya onu kullanılabilir hale getirecek yazılım tarafından desteklenebilir.

IF ... THEN kuralları yeni bir buluş değildirler. Üretim kuralları, durum faaliyet kuralları gibi isimleri vardır. Kurallar bir "IF" bölümünden, bir de "THEN" bölümünden oluşurlar.

$$\text{IF } A_1 \& A_2 \& \dots \& A_n \text{ THEN } B_1 \& B_2 \& \dots \& B_m$$

"IF" bölümü ($A_1 \& A_2 \& \dots \& A_n$) öncül, durum veya sol kısım olarak da bilinir. "THEN" bölümü ise sonuç, faaliyet veya sağ kısım olarak adlandırılabilir. Kuralın ilk bölümü doğruysa ikinci bölümü de doğrudur, eğer ilk bölüm gerçekleşiyorsa ikinci bölüm de gerçekleşir, ilk bölüm doğruysa ve ikinci bölüm bir işlem tanımlıyor ise, bu işlem icra edilir. Açıklamaları aşağıdaki gibi örnekleyebiliriz:

IF	"siparişin parça numarası = 10 AND sipariş parça gelmiş AND sipariş parçanın gelişi < 15 gün AND sipariş parça stokta"
THEN	"Stok durumu uygundur" yazınız"
IF	"siparişin parça numarası = 10 AND sipariş parça gelmiş"

AND sipariş parçanın gelişi > 15 gün"
 THEN ""Dikkat: stok durumu uygun değildir"yazınız"

İki haftada bir siparişi verilen 10 numaralı parçanın gelmemiş olması ve stoğa dahil edilmemiş olması halinde ortaya çıkabilecek problemleri önlemek amacıyla, kullanıcıyı uyan bir mesaj verilmesi istenmektedir.

Kontrol stratejisi, kuralların veriye uygulanmasını kontrol eder. Kontrol stratejisi çözülecek problemin bir stratejisidir ve bu strateji için bir çok yaklaşım kullanılmaktadır. Burada iki ana teknikten bahsedilecektir. İleriye doğru zincirleme ve geriye doğru zincirleme.

2.4.3.1. İLERİYE DOĞRU ZİNCİRLEME

Zincirleme yöntemleri ilgili konuya ait gerçekleri ve kuralları eşleştirerek sonuca varmayı hedeflerler. İleriye yönelik yaklaşımda başlangıç noktasından (ilk veri ve şartlar kümesinden) hareket edilip sonuca ulaşılır.

Bu teknik, bir kurala ait koşul cümlesiyle başlayan ve eylem kurallarını harekete geçiren işlerde kullanılır. İleriye doğru zincirleme sırasında çıkarım mekanizması IF koşullarının bilgi tabanındaki diğer kurallarla uyumunu araştırır.

İleriye doğru zincirleme tekniği, başlangıç bilgilerinden, ileriye doğru araştırma yaparak ilerler. Kurallar yeterli veri olduğu zaman çalışmaya başlarlar. Yani, yeni gelen verilerle kuralların sonuçları eşlendiği zaman, sistem işlemeye başlar.

Tablo 2.3 'de bu araştırmayı açıklamak için basit kurallar gösterilmiştir. Kurallar ilk iki durumda "A ise X" , " B ise Y" olarak okunabilir. Bu iki kural için A ve B öncüller, X ve Y ise sonuçlardır. " $\Rightarrow$ " sembolü, "belirtir, anlamına gelir" olarak okunabilir, böylece bu kurallar "A belirtir X'i" şeklinde de ifade edilebilir.

TABLO 2.3. Sonuç Çıkarmak İçin Kullanılacak Basit Kurallar

KURAL NUMARASI	KURAL	SONUÇ
1	A	$\Rightarrow X$
2	B	$\Rightarrow Y$
3	C & D	$\Rightarrow Z$
4	E & B	$\Rightarrow W$
5	Z	$\Rightarrow H$
6	X	$\Rightarrow B$
7	F & W	$\Rightarrow Z$

ÖRNEK 1:

Bu örnekte, "B,C,E" verilerimiz olsun. Bunlar çalışan hafızada stoklanır ve bir kutuyla gösterilebilir (Efsthathiou, 1989).

B, C, E

Tablo 2.3 'deki kurallar listesine bakarsak, kural 2'nin öncül olan B, çalışan hafızada olduğundan sağlandığını görüyoruz. Böylece bu kural faaliyete geçer, yani Y sonucu artık doğru olarak düşünülür ve çalışan hafızaya eklenebilir:

B, C, E Y

Kuralları incelemeye devam ettiğimizde 4. kuralın da faaliyete geçebildiğini görürüz, W bilinen olaylar listesine eklenir.

B, C, E Y W

Listedeki diğer olayların harekete geçirilemediği barıdır.Yukarıdan başlayarak listenin altlarına kadar bir kere daha inmek, bilinen olaylar listesine yeni bir bilgi katmaz.

ÖRNEK 2:

Bu kez orijinal veri "C, D, E, X" olsun.

C, D, E, X

Listeye göz attığımızda 3. kuralın faaliyete geçebileceğini, böylece Z'nin bilinen olay haline geldiğini söyleyebiliriz.

C, D, E, X
Z

Bu bilgiyle H'yı sonuçlamak için 5. kuralı da harekete geçirebiliriz.

C, D, E, X
Z
H

Bu kural orijinal veri kümesi altında harekete geçirilemezdi. Daha derin bir araştırma, kural 6'nın da harekete geçirilebileceğini, böylece B'nin bilinen olaylar listesine dahil olabileceğini gösterir.

C, D, E, X
Z
H
B
Y
W

Kurallara daha fazla inceleme yapmak artık yeni bir şey kazandırmayacaktır.

Bölümün başında bahsedilen uzman sistemin üç bileşeni, şimdi daha iyi anlaşılır hale gelmiştir. Kural temeli Tablo 2.3 'deki kural kümelerine eşdeğerdir ve çalışan hafıza yukarıda geçen küçük kutular gibi kullanılır ve onlar gibi yenilenebilir. Kontrol stratejisi ise daha komplekstir.

Yukarıdaki iki örneği düşünelim. Birincisinde bütün kuralların üzerinden geçtik ve bu çıkarılabilecek bütün sonuçları göz önüne serdi. Fakat ikinci örnekte, kuralları ikinci, üçüncü kez inceleme daha başka sonuçlar çıkarmamıza ve çalışan hafızaya daha fazla olaylar eklememize yaradı. Eğer bu örnekte kurallar bir kere daha incelenseydi, sonuçlama prosedürünün neticesi farklı olurdu. Kuralların uygulanma sırası ve bunların üzerinden geçme sayısı kontrol stratejisinin bir bölümüdür. Bir kontrol stratejisi düşünelim: "Kurallar sadece bir kere incelenecektir" olsun. Çalışan hafızanın durumu bu bir defa inceleme sonucu yukarıdakiyle aynı olacaktır. Ancak ikinci örnek için son durum farklı, yani,

C, D, E, X
Z
H
B

olurdu.

Kontrol stratejisinde değişik şekiller mümkündür. Örneğin "Kuralların üzerinden bir kez geçin ve birden fazla kuralı harekete geçirmeyin."

Kuralların tabanda depo edildiği sıra, uzman sistemin verimliliği üzerinde etkiye sahip olabilir. Kuralları dizmek için sezgisel stratejiler kullanılabilir, ama kural tabanlı sistemlerde problemlerin, kuralların dizilişinin sadece uzman sistemin uygulama verimini değil, aynı zamanda ulaşabildiği sonuçları da etkileyebilmesidir. Kontrol stratejisi kuralların dizimini olduğu gibi üzerlerinden geçiş sayısını da düşünmek zorundadır.

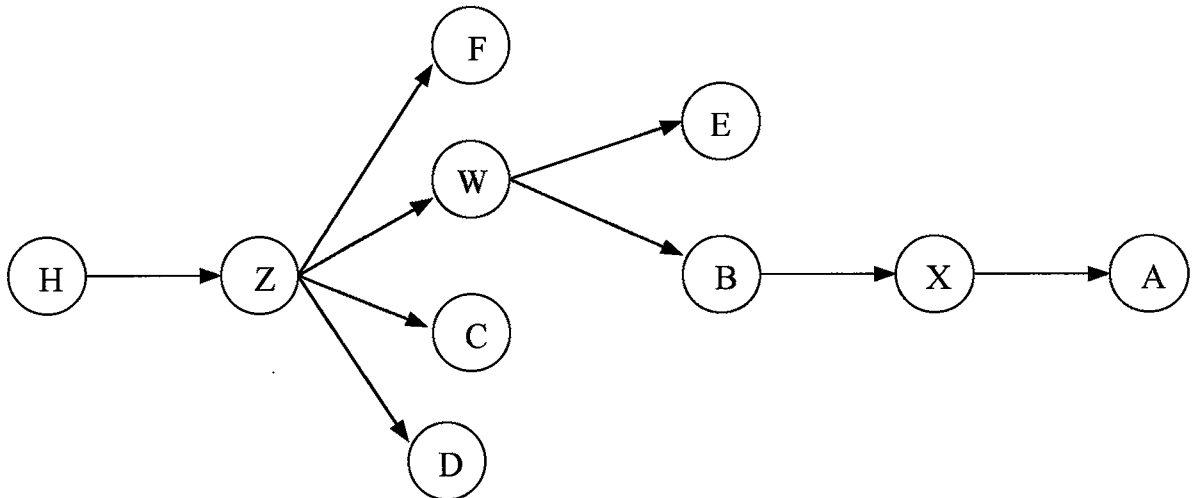
2.4.3.2. GERİYE DOĞRU ZİNCİRLEME

İleriye doğru zincirleme, bilinen verileri kuralların öncülleriyle eşleştirerek çalışır ve başka hangi sonuçların çıkarılabileceğini araştırır. Geriye doğru zincirleme ise, tersine, bilinen veya hipotezleştirilmiş sonuçları kuralların neticeleriyle eşleştirir. Sistem neyi ispatlamak istiyorsa, çözüme onunla başladığı bir çıkarım metodudur. Kurallar, ortaya konan IF ... THEN koşullarını deneyerek, ihtiyaç duyulan yeni değerleri bulmak için bir kuraldan diğerine doğru ilerlerler.

Geriye doğru zincirleme tekniğinde bir problemin özel bir kısmı üzerinde çalışılır. Bu, bir probleme ya da alt probleme özgü bir çözümdür. Kuralların sonuçları, yalnız göz önüne alınan problemlere özgü ya da kısmi çözüm olduğunda çalışır. Bir kural kullanılırsa, onun koşulları üzerinde odaklanılan sisteme özgü yeni bir çözüm olur.

ÖRNEK 1:

Çalışan hafızada F, E, C, D verilerimiz olsun ve H'yi ispatlayalım. Bunun için Şekil 2.6'yı ve Tablo 2.3'ü kullanacağız (Efsthathiou, 1989).



Şekil 2.6. H'yi İspatlamak İçin Kullanılacak Geriye Doğru Zincirleme Ağacı

F,E,C,D İspatla H

Şekil 2.6, F ve W'nin doğru olduğunun gösterilmesiyle ispatlanabilecek Z hipotezini yaratır.

F,E,C,D İspatla H Hip Z Hip F Hip W

F hipotezi orijinal verinin içinde bulunduğu için doğru olarak görülebilir ve başlangıç verisinin içinde olmayan W ispatlanmaya çalışılır. W, E ve B'nin doğruluğunu göstererek ispatlanabilir.

F,E,C,D İspatla H Hip Z HIP F doğru Hip W Hip E Hip B

Orijinal veriye bakılarak E'nin doğru olduğu görülür. B'nin doğru olduğunu göstermek için yapılan incelemeler X'in doğruluğunun gösterilmesini gerektirir.

F,E,C,D İspatla H Hip Z Hip F doğru Hip W Hip E doğru Hip B Hip X
--

X orijinal verinin parçası değildir. A ispatlanarak X'in doğruluğuna ulaşılır.

FECD
İspatla H
Hip Z
Hip F doğru
Hip W
Hip E doğru
Hip B
Hip X
Hip A

Fakat A da orijinal verinin bir parçası değildir ve A'nın doğruluğunu ispatlamanın başka bir yolu yoktur. Bu yüzden X'i, B'yi ve diğerlerini ispatlamak için kullanılan bu metod bırakılır. Hipotez ağacı, H'yi ispatlamak için yeni bir nokta bulunana kadar tekrar incelenir. Bu prosese "geriye doğru izleme" denir. İşlem tamamlandığında, çalışan hafızanın görüntüsü şöyle olur:

F,E,C,D
İspatla H
Hip Z

Z'yi ispatlamak için diğer seçenek, C ve D'nin doğruluğunun gösterilmesidir. Son iki hipotez orijinal veride yer aldığı için Z'nin, dolayısı ile H'nin doğruluğu ispatlanmış olur.

F,E,C,D
İspatla H
Hip Z
Hip C
Hip D

Yukarıdaki örnekte W'yi ispatlamayı denemeden önce, C ve D için daha kısa bir ispat önceden kontrol edilseydi, emekten çok tasarruf edilebilirdi. Bunun için geriye doğru izleme algoritmalarını geliştirici çalışmalar devam etmektedir.

2.4.3.3. BELİRSİZLİKLE MÜCADELE

Kural temelli bir uzman sistem, kalıpla yönetilen sonuç çıkarım sisteminin bir örneğidir ki, bu sistemde giren veriler, kurallar toplamı karşısında eşleştirilir ve veri kalıbı kuralın veya kuralların sol kısmıyla eşleştirildiğinde bunlar harekete geçer ve bir durum oluşur. Belirsizliği temsil ederken sadece birkaç değere bağlı kalmak yerine, bazı metotlar Bulanık Sonuç Çıkarma gibi değerleri kullanırlar.

Bulanık Sonuç Çıkarma, bulanık küme teorisine dayanır. Bir bulanık küme, insanların dünyayı tanımlarken sık sık kullandıkları dile ait kavramları temsil etmek için kullanılır. Örneğin sıcak, soğuk gibi derece sıfatları kesin olarak tanımlanamazlar, çünkü tanımlanan şeye göre bir derecenin altındakiler soğuktur, üstündekiler ise sıcaktır. Bulanık Sonuç Çıkarma, bu problemin üstesinden 0 veya 1 dereceli üyeleri kaldırarak gelir- ki bu bir parçanın kümenin üyesi olup olmadığını belirler- ve üyeliğin derecelerini 0 ve 1 de dahil olmak üzere bu iki rakam arasında sıralar.

Dilbilime ait olan bu terimler “ve, veya, ama” bağlaçlarını kullanarak değiştirilebilir. Bulanık temsil, orijinal dil kalıplarını muhafaza eder, böylece okuyucu için anlayışı kolaylaştırır, ama nümerik olarak gösterimi kompleks hale sokar (Efstathiou, 1989).

3. BÖLÜM

UZMAN SİSTEMLERİN TASARLANMASI

3.1. UZMAN SİSTEMLERİN TASARLANMASINDA KULLANILAN YAZILIMLAR

Uzman sistem yazılımları, bir uzman sistem oluşturma işini basitleştiren programlama sistemleridir. Bir uzman sistem kurma aracı, programlama dili ve destek pakettir. Uzman sistemlerin avantajı, kural tabanlı bilgi temsiline ve temsili kolaylaştırmak için düzenlenmiş dillere yüklenebilir. Kural tabanlı programlama ve bilgi temsiline küçük uzman sistemler, sınırlı bilgi tabanları ile çok fazla çaba harcamadan kurulabilir. Bazı yapay zeka dilleri, kural tabanlı olma niteliklerini diğer avantajlarla birleştirir. Örneğin, PROLOG, mantığa dayalı sonuçlandırmanın avantajlarını içeren kural tabanlı bir dildir. Kapsamlı programlama tipleri kadar uzman sistem kabuğunu yazmaya da başarılı olarak uygulanmıştır.

Uzman sistemlerde bilgi temsili için kullanılan daha genel yapay zeka tekniklerinden en yaygın olarak incelenenleri mantık, semantik ağlar ve çerçevelerdir. PROLOG, LISP, OPS-5, SAIL ve RLL özel diller arasındadır. Bunlardan bir çoğu uzman sistem araştırmalarında kullanılmaktadır (Simons, 1985).

Uzman sistem yazılım sınıfları üç ana gruba bölünebilir. Bunlar özel uzman sistemler, kabuk sistemler ve araçlardır.

Özel uzman sistemler, özel bir konu için özel bir kullanıcı öneren final ürünlerdir.

Kabuk sistemler, daha önce kurulmuş bir uzman sistemden yararlanmak amacıyla geliştirilmiş yazılım araçlarıdır. Bir kabukta özel alana ait bütün bilgiler açık olarak, kurallar halinde temsil edilir. Bu yapı, özel uzman sistemlerin yapısı ile birleştirilerek,

farklı bir görev için yeni bir bilgi tabanı ile orijinal bilgi tabanını değiştirmeyi mümkün kılar. Kabuklar, özel uzman sistemleri çabuk, kolay ve düşük maliyetle kurmak için geliştirilmiş önemli bir yazılım paketidir. EMYCIN, EXPERT, KAS, HEARSAY-3 ve SAGE en çok kullanılan uzman sistem kabukları arasındadır (Turban, 1988).

Kabukların kullanımında bazı problemler çıkmaktadır. Bunlardan biri, problem alanının ihtiyaç ve karakteristiklerinin, kabuğun dizaynıyla yeteri kadar eşleşme sağlayamamasıdır. Bilgiyi sadece bilgi tabanına koymak başarılı bir sonuca ulaşmak için yetersiz kalmaktadır. Bilgiden en iyi yararlanma ve uygun kontrol mekanizması, uygulamanın karakteristiklerine önemli derecede bağlıdır. Bir kabuk, uygulama üzerinde, yabancı bir bilgi temsil formülasyonunu yükleyebilir. Bilgi alanının gerekli tüm bölümlerini ve bu formülasyonları birbirine uygun hale getirme yeteneği oldukça kısıtlıdır. Bu durum, bilgiyi güncelleştirmeyi ve programın sonuç çıkarma prosesinin takibini kullanıcı ve programlayıcı için güncelleştirir.

Araçlar, kabuk sistemleri veya özel uzman sistemleri kurmak için kullanılabilirler. Uzman sistem kabukları, özel uzman sistemleri kurmak için kullanılır. Uzman sistem araçları, kabukları veya özel uzman sistemleri kurabilecek hızlı prototip yaratma ortamı sağlar. Örneğin ART, Inference Corporation tarafından kurulmuş bir pakettir. Sonuç çıkarma mekanizmasının monte edilebileceği modüller içerir.

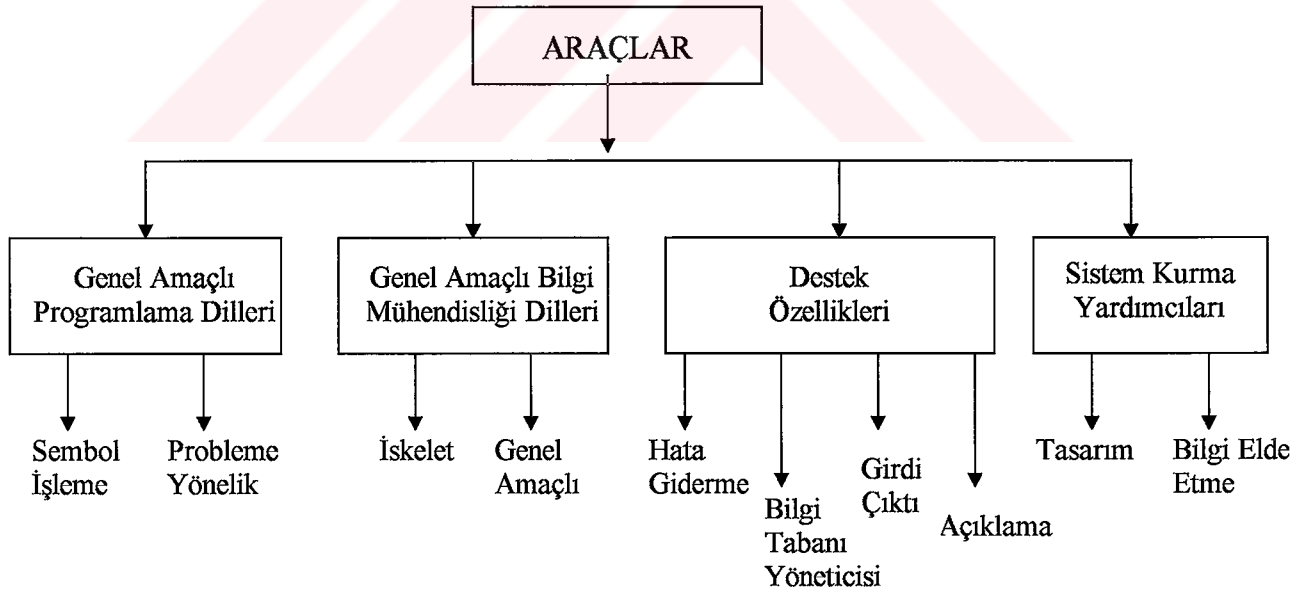
Kabuklar ve araçlar yoğunlaşma dereceleri bakımından farklılıklar gösterirler. Araçlar daha esnektir fakat bir konuda yoğunlaşması daha azdır. Bilgi mühendisine daha çok sorumluluk yüklerler. Kabuklar daha dar bir uygulama alanı sunarlar fakat konuya daha yoğun olarak yaklaşırlar. Araçlar olarak adlandırılan yazılım paketleri dört ana bölüme ayrılabilir (Şekil 3.1):

a. Genel Amaçlı Programlama Dilleri

Çok sayıda programlama dili, yapay zeka ile ilgili çalışmalarla ortaya çıkan problemleri simgelemeye ve çözmeye yönelmiştir. Bunlara sembol işleme dilleri denir.

Sembol işleme dillerinin en önemlileri LISP ve PROLOG'tur. LISP, sembolleri liste yapıları şeklinde işleyecek mekanizmalara sahiptir. Liste yapıları, karmaşık kavramları simgelemek için kullanılan inşa bloklarıdır. PROLOG'da diğer programlama dillerinde yer alan fonksiyonlar ve alt programlara karşı öncelikler bulunmaktadır.

Bazı uzman sistem geliştirme araçları ve kabuklar yapay zeka dili olmayan dillerle yazılmışlardır. Bu diller, probleme yöneliktir ve özel bir sınıf için tasarlanmışlardır. Örneğin, Fortran, Pascal, Basic, C dilleri gibi. Yapay zeka dillerinin düzenlenmeleri daha fazla hafızaya ihtiyaç duyabilir. Bundan dolayı yapay zeka dilleri fazla kısıtlayıcı olabilirler. Probleme yönelik dillerle yazılmış uzman sistemler mikro bilgisayarlarda, yapay zeka dilleri ile yazılmış olanlardan daha hızlı çalışırlar.



Şekil 3.1. Uzman Sistem Tasarlama da Kullanılan Araçlar

b. Genel Amaçlı Bilgi Mühendisliği Dilleri

Bir bilgi mühendisliği dili, uzman sistem geliştirmek için kullanılan ve geniş bir destek çevre içine toplanmış bir uzman sistem dillerinden oluşmuş ileri düzeyde bir araçtır. Genel olarak kabuk sistemlerden daha esnektirler. Diğer yandan, girdi/çıkış prosesleri, bilgi tabanı yapısı için karmaşık özellikleri azaltabilirler. Bundan dolayı programlama çevreleri kabuk programlar tarafından sağlananlar kadar kapsamlı değildir. HEARSAY-III, ROSIE, OPS-5, RLL bu sınıfa giren diller arasındadır.

İskelet sistemler, alan bilgisini mevcut bir uzman sistemden çıkararak türetilen bilgisayar dilleridir. Sistem gelişimini hızlandıran yapı ve yapım kolaylıkları sağlarlar. Esneklikten ve genellemeden yoksundurlar, sınırlı bir problem alanına uygulanırlar.

Genel amaçlı sistemler, farklı problem alanları ve tiplerine uygulanabilir özellikler taşıyan bilgisayar dilleridir.

c. Destek Özellikleri

Programlayıcılar tarafından üretkenliği artırmak için kullanılan araçlardır. Waterman (1983) tarafından, destek özellikleri için dört farklı sınıf tanımlanmıştır.

Hata giderme yardımcıları: Bu mekanizma, kullanıcıya ileride programın nerede duracağını söyleme fırsatı verir. Böylece kullanıcı bazı hataların oluşmasından önce programın işleyişini durdurabilir ve veri tabanındaki mevcut değerleri gözden geçirebilir.

Bilgi tabanı yöneticileri: Otomatik sayıcılar, bir olaya ilişkin kayıtları ve kullanıcı tarafından yapılan değişiklikleri gözler. Eğer kullanıcı bir kural ekler ya da değiştirirse, yönetici, daha sonra başvurulacak bir referans olması için değişim gününü, kuralın ve kullanıcının adını saklar.

Söz dizimi kontrolü, kullanıcının doğru yazım kuralları ve formatla kurallar girmesine yardım etmek için, uzman sistem dilinin gramer yapısı konusundaki bilginin editör tarafından kullanılması durumunda önem kazanır. Kullanıcı, gramerde olmayan bir kural girdiğinde, yönetici hatayı yakalar ve yanlış konusunda kullanıcıyı uyarır.

Oluşum kontrolü, girilen veri ve kuralların, sistemde varolan bilgiyle çelişip, çelişmediğini görmek için, sistem tarafından veri ve kuralların anlam yapısının ve anlamsal ağlarının kontrol edilmesinde kullanılır.

Girdi/çıkı özellikleri: İşleme süresince bilgi edinme, araçtaki mekanizmaların kendi kendine ve kullanıcının çalışan uzman sistem ile iletişim kurmasına izin verir. Menüler, sistem tarafından sorulan bilgi girilmek istendiğinde kullanıcının içinden seçerek bilgi girmesi düşüncesiyle sağlanır. Uzman sistem çalıştığı sürece, kullanıcının istediği bilgiyi girmesine olanak verirler.

Açıklama özellikleri: Kullanıcı, sistemin problemi nasıl çözümlediği konusunda bilgi sağlamak için soru sorabilmelidir. Örneğin, sorulan bir "neden" sorusu, sistemin kullanıcıdan bazı bilgileri neden istediğini açıklamasına sebep olacaktır, "nasıl" sorusu da, sistemin özel sonuçlarını nasıl elde ettiğini açıklamasına sebep olacaktır. Verilen cevap, sonuca ulaşmak için kullanılan kuralları belirlemeyi de içerecektir. Sistem, özel bir olay ya da kuralın farklı olması durumunda daha başka hangi sonuçlara ulaşabileceğini açıklar. Karşıt olarak sistem beklenen bir sonuca neden ulaşamadığını da açıklayabilmektedir.

d. Sistem Kurma Yardımcıları

Uzman kişinin bilgisini sağlamaya ve temsil etmeye yardım eden programları ve yapım sırasında uzman sistem tasarımına yardım eden programları kapsar. Bu programlar, pratik destek geliştirmek için başlangıç niteliğindeki araştırma araçlarıdır. Sistem kurma yardımcıları iki sınıfı şöyledir (Grogono et al, 1991):

1. Tasarım yardımcıları: AGE, TIMM, EXPERT, EASE
2. Bilgi elde etme yardımcıları: TEIRESIAS, ONCOCIN, OPAL, SEEK, SEEK2

Genel olarak, uzman sistem araçlarının çoğunun performansı, sistem kurucusunun ihtiyaç ve amaçlarına karşılık verebilecek düzeyde değildir. Bilgi temsil yöntemi ve kontrol stratejisi çoğunlukla, gerçek problemler için rijit kalmaktadır (Pollitzer et al, 1985).

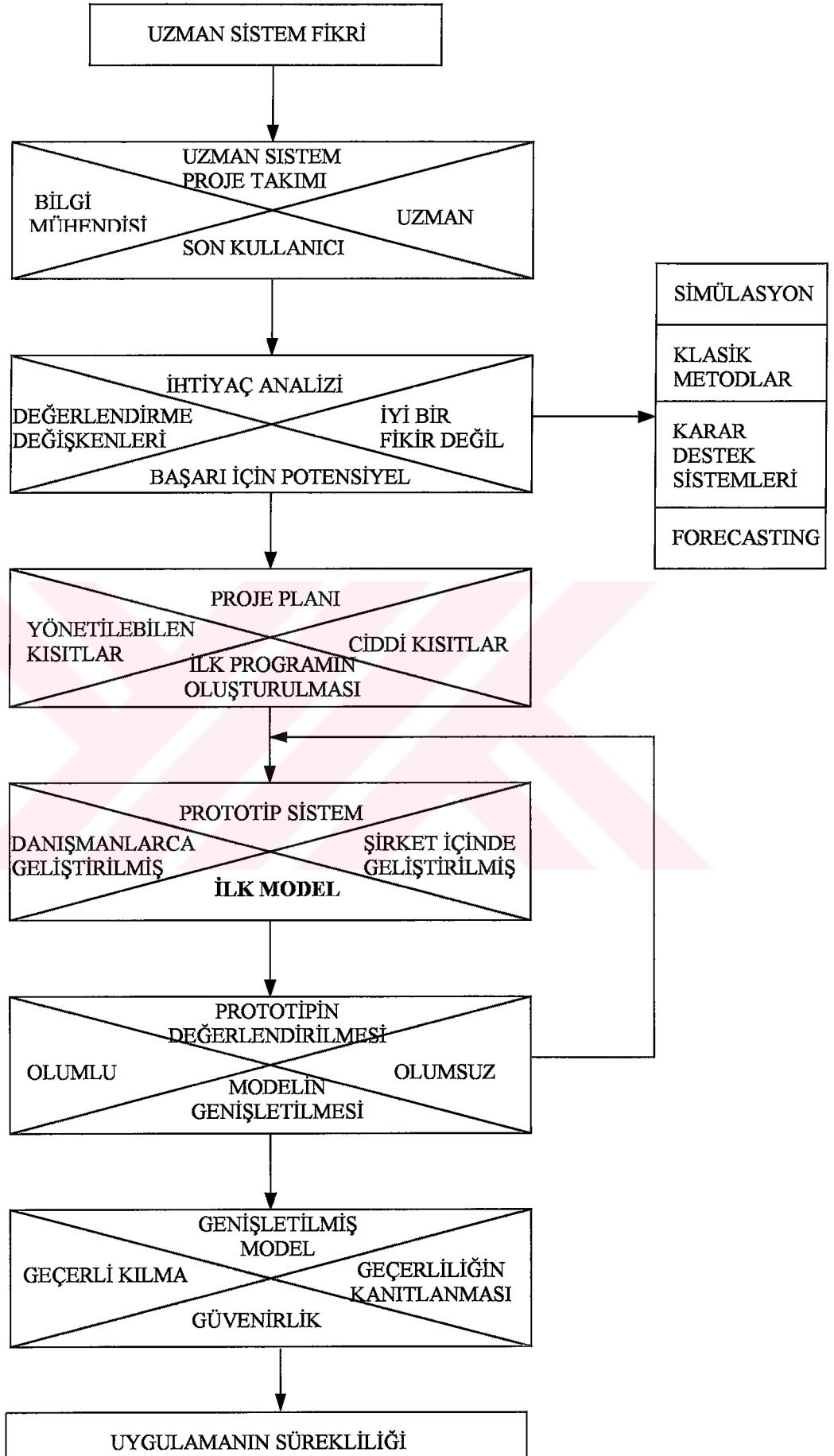
3.2. UZMAN SİSTEMLERİN GELİŞTİRİLMESİ

Bir uzman sistemin geliştirilmesi diğer tip bilgisayar uygulamalarından farklıdır. İlk fark olarak özel bir donanım ve yazılımın kullanımının mümkün olması gösterilebilir. Diğer bir fark uzmandır. Aktif katılım ve istek duymayan bir uzman, geliştirme esnasında hatalara neden olabilir. Bir diğer fark, prototip geliştirmenin önemidir. Prototip uzman sistemin küçük bir örneğidir. Burada, başarılı uzman sistem projeleri için kullanışlı prosedürler anlatılacaktır. Şekil 3.2 bir uzman sistemin geliştirilmesi ve elde edilmesinin adımlarını göstermektedir (Ntuen, 1991).

Uzman sistemlerin başarıyla geliştirilmesi genellikle en az dört farklı, bazan birbirinden bağımsız insan katılımını gerektirir: Uzman, bilgi mühendisi, sistem programcısı ve son kullanıcı. Her birinin problem çözümüne farklı davranışı ve farklı yaklaşımı vardır. Uzlaşma ve işbirliği eksikliği, uzman sistem geliştirmeye daha tasarım aşamasında problem yaratır. Yeni bir girişim, genellikle geliştirme veya yeni bir tasarım fikriyle başlar. Fikir aşamasında yapılacak işlemler şunlardır:

3.2.1. PROBLEMİN BELİRLENMESİ VE ÇÖZÜM METODOLOJİSİ

Çözülecek problemin özellikleri problem tanımlanarak belirlenebilir. Organizasyon daha önce bu tip problemler çözmüş mü, çözmüşse nasıl çözmüş,



Şekil 3.2. Uzman Sistem Geliştirme Çevrimi

çözüm başarılı mı, mevcut modellerin başarısı ve başarısızlığı hakkında ne biliniyor, uzman sistem teknolojisi bu probleme nasıl çözüm sağlar, başarılması gereken amaçlar nelerdir, uzman sistem bu amaçları nasıl gerçekleştirir, diğer teknolojiler daha iyi olabilir mi gibi soruların yanıtları problem hakkında sağlam bilgi elde edilmesine yardım eder. Uygun olmayan uygulama sahaları için uzman sistem geliştirilmesi mümkün değildir. Yanlış bir seçim bizi çözümsüz problemlere ve gelişme aşamasında başarısız ve kullanışsız bir sisteme götürür. Problem alanı dar olmalı ve iyi sınırlandırılmalıdır. Problem sezgisel yeteneklere ihtiyaç duymalıdır. Bunun için problem durumunun detaylı olarak incelenmesi, problem ile ilgili değişkenlerin, kısıtların belirlenmesi gereklidir. Uzman sistem geliştirilmesi ile ulaşılabilecek amaçların ve hedeflerin tanımlanması, böylece çabaların maliyet ve kazandıracaklarının analiz edilmesi gereklidir. Daha sonra, eldeki tanımlanmış bir problem ifadesiyle, uygun bir çözüm metodolojisi araştırılmalıdır. Problemin geleneksel bilgisayar teknikleri tarafından tatmin edici bir şekilde çözülemeyeceği belirlenmelidir. Daha basit bir teknik mevcutsa, problemin bir uzman sistem tarafından çözülmesinin anlamı yoktur. Eğer uzman sistem kullanılması gerekiyorsa insan uzmanının varlığı da sağlanmalıdır. Bir uzman sistemin geliştirilmesinde dikkat edilmesi gereken tuzaklar aşağıdaki gibi sıralanabilir.

- Hiç bir kimsenin ihtiyaç duymadığı bir sistemin geliştirilmesi
- Sistemin nihai kullanıcının kullanım pratiğine uygun olmaması
- Uzman sistem için bilgi sunacak istekli uzman olmaması
- İşletme verilerinin eksik veya gerçekçi olmaması
- Seçilen problem alanının teknik uygunsuzluğu

Bir uzman sistem çabucak geliştirilemez. Görev ve kaynak kısıtları uzman sistem projesini, kontrol edilemeyen zaman gecikmeleri nedeniyle komplike hale getirir. Ekseriya, mevcut kaynak elde edilebilirliğine bağlı, taslak bir çizelgenin planlanmasını gerektirir.

3.2.2. PROTOTİP OLUŞTURMA

Bundan sonraki adım prototip oluşturmaktır. Prototip, projeyi amaçlarına ulaştıracak bir "erken uyarı" sistemidir. Uzman sistemlerde bir prototip, küçük boyutta bir sistemi niteler. Belirli bir senaryo için uzman sistem geliştirmek maliyetli olabilir, gerçek sistemin prototipini geliştirmekle işe başlamak tedbirli bir davranış olur. Prototip, çabuk sonuca varmayı sağlayacak kapasitedeki bilgi temsili ve eksik temel üzerindeki uzman sistemin ana bileşenlerini yaratmayı içerir. Örneğin kural tabanlı bir sistemde, prototip sadece 50 kural içerebilir. Fakat bu az sayıdaki kurallar sınırlı alana danışmanlık üretmek için yeterlidir. Prototipleme burada uzman sistemlere uygulanmasına rağmen, diğer bileşim ve karar destek sistemleri için de kullanılmaktadır.

Prototip geliştirmeye başlamadan önce gerekli bilginin elde edilmiş olması lazımdır. Geliştirme işlemi boş bir kitaba sayfalar eklemek gibidir. Prototip tasarımının etkisinin mühendis tarafından bilinmesi önemlidir; bu şekilde mühendis problemin önemli beklentilerini tesbit eder. Bilgi mühendisi, uzmanın işini temsil eden hedef problemleri az sayıda problem ile sınırlamak zorundadır. Genel ya da genel dışı bilgi beklentilerini yerine getirmek, görevleri başarmak için gereklidir. Bu beklentiler çoğunlukla sistemin son kullanıcıları tarafından iletilir. Böylece bilgi mühendisi son kullanıcıya açık bir model teslim eder. Hedef kullanıcı, uzman sistem tarafından sağlanmış gerekli bilgileri almadıkça, sistemi etkili olarak kullanamaz.

Prototip, geliştirildiği alan için model içerikleri, yapıları ve yetenekleri yönünden test edilmelidir. Prototip bilgi mühendisine, zayıflıklar üzerinde önemli bir geri besleme sağlar ve nihai sistemde nasıl bir teknik performans elde edilebileceğini gösterir.

Prototip, yeni problemlerin üzerinde kullanılmasıyla kontrol edilebilir. Bilgi mühendisi ve uzman, prototipin meseleleri ele alıp almadığını kontrol eder ve bilgi alanının gösterilmesi ve sonuç çıkarma kontrolünün yeterli olup olmadığı konusunda genel stratejilerin saptanıp saptanmadığına bakar. Eğer istenilenler elde edilemezse,

düzeltilmiş bir tasarımla ikinci prototip inşa edilir. Bu aşamada proje için tekrar araç veya kabuk bulmak gerekebilir. Ayrıca bilgi mühendisi bilginin daha uygun tarzda yapılanmasıyla ilgilenir. Her iterasyonda önemli değişiklikler olmasa da prototipin iterasyon sayısı fazla olabilir. Ayrıca prototip diğer bazı fonksiyonları icra eder. Bu ilk anda uzmanın gördüğü ve kullandığı bilgi tabanı sistemi olabilir. Daha sonraki aşamada nihai sistem, uzman tarafından düzeltilmeye ihtiyaç duyacaktır ve bu prototipe giriş olayı yanlışlıkların görülmesini sağlayacaktır. Çalışan bir model ne getireceği bilinmeyen binlerce düşünceden daha değerlidir. Başarılı bir prototip, nihai sistem için mühendisin bir gelişme planı ileri sürebileceği şekilde hareket edecektir. Hedef bilgi ve görev daha kesin olarak tesbit edilebilir. Herhangi bir sistem ilerlemeye ihtiyaç duyarsa, prototip aşamasının sonunda bir performans kriteri seti yerleştirilir.

Genişletilmiş model, prototip modelin geliştirilmiş halidir. Burada karşılaşılabilecek muhtemel iki temel problem,

1. Modelin yeniden yapılandırılabilirliği: Prototip model tarafından sınırlı olarak dikkate alınan ihtimalleri belirlemek için model kontrol stratejilerinin ve planlarının genişletilmesi.
2. Modelin yeniden düzenlenebilmesi: Bilgi temsiliinin, modellenen durumla ilgili dinamik ve koşullardan bağımsız bilgi kaynaklarına uyumlu olarak yeniden yapılabilmesi.

3.2.3. KULLANIM VE SÜREKLİLİK SAĞLAMA

Hiçbir uzman tecrübesiyle yetinmez. Uzmanlar tecrübelerini geliştirebilirler, problemlere çözüm yaklaşımları değişebilir. Aynı şekilde uzman sistemler için de, bilgi ve kurallar değişebilir veya genişletilebilir. Bu aşağıdaki şekillerde gerçekleşebilir:

- Bilgi tabanı sistemle birlikte yeni bilgiler için genişletilebilir.
- Girilen bilgiler hataların anlaşılması sonucu değişikliğe ihtiyaç duyabilir.

- Bilgi tabanı uzmanın sezgisel kurallarının deęiřmesi sebebiyle deęiřiklięe ihtiya duyabilir.
- řirket iindeki prosedürlerin deęiřmesi, eřitli profesyonel istekler vb. deęiřiklikler.

Bonnet, Hatton ve T-Ngoc (1988) 'un gözlemleri uzman sistemler hakkında ařağıdaki sonuçları vermektedir.

"Bir uzman sistem projesi dięer yatırımlar gibi, stratejik ve ekonomik fırsatları, kullanışlılığı, kullanıcıları tarafından veya pazarda kabul edilme fırsatları aısından, sistemli ve alternatif projelerle rekabet halinde deęerlendirilmelidir."

"Bir uzman sistem, anlaşılması kolay fakat üretilmesi zor bir sistemdir. Operasyonel sistemlerin yaratılmasında gerekli olan karmařık teknikler, sadece kolayca elde edilemeyen veya eęitilemeyen kalifiye insanlar tarafından geliştirilmektedir."

4. BÖLÜM

ÜRETİM PLANLAMADA UZMAN SİSTEMLER

4.1. ÜRETİM PLANLAMA PROBLEMLERİ

Programlama, kaynakların, görev topluluklarını yerine getirmek üzere bir zaman süresince tahsisi olarak tanımlanabilir (Baker, 1974). Bu tanımlı iki şekilde genişletmek mümkündür.

1. Programlama kararının bir parçası olan kaynak seviyesi planlamayı içererek
2. Tahsis kararını sadece zaman açısından değil, diğer kaynakların atanması konusunda da vererek. Bu amaçla üretim programlama terimi, üretim planlama faaliyetlerini içerecek şekilde genişletilebilir.

Üretim planlama faaliyetleri yönetim faaliyetleri içinde en karmaşık olan alandır. Kısıtlar geniş ve değişiktir, amaçlar birbiriyle çelişir ve planlama problemi zor bir hal alır. Karmaşıklığı iki ana faktörden ileri gelir. Birincisi, pazar taleplerindeki dalgalanmaların, ticari önceliklerdeki değişmelerin, hammadde ve malzeme bulunabilirliğinin ve üretim ortamının dinamik ve önceden haber verilemez olmasının oluşturduğu karışıklıklardır. Bu sebeplerden dolayı, planlamacılar bir haftadan daha uzun süreler için plan yapmama eğilimindedirler. Bu davranışın ise özellikle tedarik açısından negatif etkisi olmaktadır.

Karmaşıklığı oluşturan ikinci faktör, sağlanmak zorunda olan bir çok kısıtlar ve amaçların olmasıdır. Örneğin,

1. Üretim ihtiyaçları: Düzgün ve stabil bir üretim seyri devam ettirmek ve engellerle mücadele edebilmek için bir miktar stok bulundurmak
2. Müşteri servisi: Müşteri siparişlerinin karşılanmasındaki gecikmelerin önlenmesi ve stoksuz kalmayı önlemek için uygun ürün stok seviyesi sağlamak.

3. Maliyetlerin düşürülmesi: Stokları ve işletme sermayesini azaltarak, pahalı ekipmanın maksimum kullanımını sağlayarak, maliyetleri mümkün olduğu kadar azaltmak.

Bu amaçlar bazan birbiriyle çelişir. Örneğin, müşteri siparişlerinin zamanında karşılanması, planlamacıyı, işe mümkün olduğu kadar erken başlamaya sevkeder. Diğer yandan envanteri minimize etmek için, o işi mümkün olduğu kadar ertelemek ister. Bu çelişkiler bir programı değerlendirmeyi zorlaştırır. Programlama sisteminin, değişik amaçlardan kaynaklanan farklı perspektifleri gözleme kabiliyetinde olması istenir. Eğer böyle gözlemler, gerçek zaman tabanında gerçekleştirilirse, değişik perspektifler, çabuk geri besleme ile üretim ortamının daha iyi kontrolünü sağlar.

4.2. ÜRETİM PLANLAMAYA ÖNCEKİ YAKLAŞIMLAR

Üretim planlama problemlerinin çözümü için Yöneylem Araştırması ve Yapay Zeka araştırmacıları tarafından sayısız yaklaşımlar önerilmiştir. Yıllarca analitik yaklaşımlar yapılmıştır. Optimal çözümlerin bulunmasını amaçlayan bu yaklaşımların, programlama probleminin çok küçük bir alt kümesinde uygulanabilir olduğu ispatlanmıştır: Oldukça idealize edilmiş durumlarda ve küçük iş ve makina sayılarında (Graves, 1981). Burada yapılan tipik varsayımlar,

1. Yapılacak işin iyi tanımlanması ve tamamen bilinmesi
2. Kaynaklar ve işlerin tamamen belirlenmesi
3. İşlemlerin sıralarının iyi tanımlanması
4. Her makinanın sürekli olarak kullanılabilmesi
5. Yeniden işleme olmaması
6. Her bir işlemin sadece bir makina tarafından yapılabilmesi
7. Her makinanın bir anda en fazla bir işi yapabilmesi

Çoğunlukla bu ideal varsayımlarla yapılan programlar gerçekte yetersiz kalır. Analitik düşüncede işler ve makinalar az sayıyla sınırlanmıştır. Gerçekte, bu sayılar

önemli ölçüde büyüktür ve tamsayı programlama, dal-sınır algoritması gibi teknikler hesaplama açısından pratik değildir. Ancak bu teknikler, programlama sistemini tasarlamak için yardımcı olmaktadır. Bu yardım, problemin hesaplama karmaşıklığını belirlemeyi ve etkili algoritmalar geliştirmeyi kapsamaktadır. Deneysel araştırmalardan çıkan Darboğaz Giderme Sezgisel Tekniği, Gecikmelerin Maliyeti Sezgisel Tekniği, sonraları Yapay Zeka araştırmalarında, boyut ve karmaşıklığı yönünden gerçeğe yakın olan, programlama problemine tatminkar çözümler bulmak için kullanılmıştır (Hadavi et al, 1990). Ne Yöneylem Araştırması, ne de Yapay Zeka araştırmaları, siparişin tesliminin kontroluna yeteri kadar önem vermiştir. İyi programlama yapmak için işin tesliminin kontrolunun önemli rol oynadığı ispatlanmasına rağmen, işin gelişi ile çıkışı arasında gözetim yapılmamıştır. İyi bir teslimat kontroluyla, işin bekleme zamanını, envanter seviyelerini azaltmak mümkündür. Ayrıca teslim tarihi karşılanırken çevrim süreleri de azaltılır.

Bir uzman sistem, planlama probleminin ihtiyaçlarını karşılamak için uygun bir yaklaşımdır. Uzman sistem planlamacılara, görevlerini yapmak için yardımcı olacak araçları birleştiren etkileşimli bir bilgisayar sistemidir.

4.3. ÜRETİM PLANLAMAYA UZMAN SİSTEM YAKLAŞIMININ UYGULANMASI

Uzman sistemlerin üretim planlama faaliyetlerine hangi konularda nasıl çözümler getirebileceğinin belirlenebilmesi için bir işletmede üretim planı hazırlanırken yapılan işlemler anlatılacaktır. İşletme, sağlık sektörüne, serum ve serum setleri üretimi ile hizmet vermektedir.

Serum üretiminin aşamalarını şöyle sıralayabiliriz:

1. Karıştırma
2. Dolum
3. Sterilizasyon

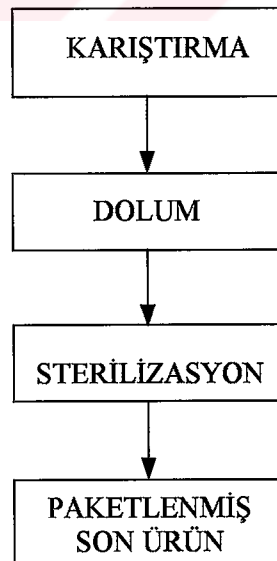
4. Ambalajlama

Karıştırma, serumu oluşturacak hammaddelerin belirli miktarlarda tartılarak, karıştırma tanklarında belirli hacimde su ile homojen bir hale gelene kadar karıştırılmasıdır. Homojeniteye ulaşıncaya karışım dolum odalarına transfer edilir. Üretimde 5 adet karıştırma tankı kullanılmaktadır.

Dolum, karışımın değişik tür ve hacimdeki kaplara konması işlemidir. Dolum boyunca karışım tanklardan direkt olarak boru hatları vasıtasıyla alınır. 2 adet dolum makinasının biri cam şişe, diğeri poşet için kullanılmaktadır.

Sterilizasyon, dolumu yapılan ilacın mikroorganizmalara karşı steril olması için belirli sıcaklıklarda bekletilmesi işlemidir. Bu amaçla kullanılan 2 otoklav bulunmaktadır.

Aynı adı taşıyan son ürün, paketlenmesine (kapların tipi ve hacmi) bağlı olarak birçok ürünü oluşturmaktadır. Toplam olarak 48 karışım, 40 hammaddeyi kullanarak, 110 son ürünü oluşturmaktadır. Ürün yapısı Şekil 4.1'de görülmektedir.



Şekil 4.1. Ürün yapısı

4.3.1. ÜRETİM PLANININ HAZIRLANMASI

Bir üretim planı hazırlanırken hangi ürünün, hangi miktarda, ne zaman, hangi kaynaklar kullanılarak üretileceği belirlenmelidir. Üretim planında yer alacak ürünlerin seçimi, ürün stoklarının istenen seviyeye getirilmesi veya müşteri siparişlerinin karşılanmasıyla ilgilidir. Karıştırma işlemi için, parti büyüklüğünün seçimi, kullanılacak steril otoklavının kapasitesine bağlıdır. Her bir ürünün üretim sırası, hem karıştırma hem de dolum için önemlidir. Çünkü bazı ürünler için birinden diğerine geçiş sırasında ekipmanın yıkanması gerekmektedir. Yıkama işlemi, düşük prodüktivite ve yüksek maliyet demektir. Bu yüzden, planlama sırasında ürünlerin birbiriyle uyumlu olmasına dikkat etmek gerekmektedir.

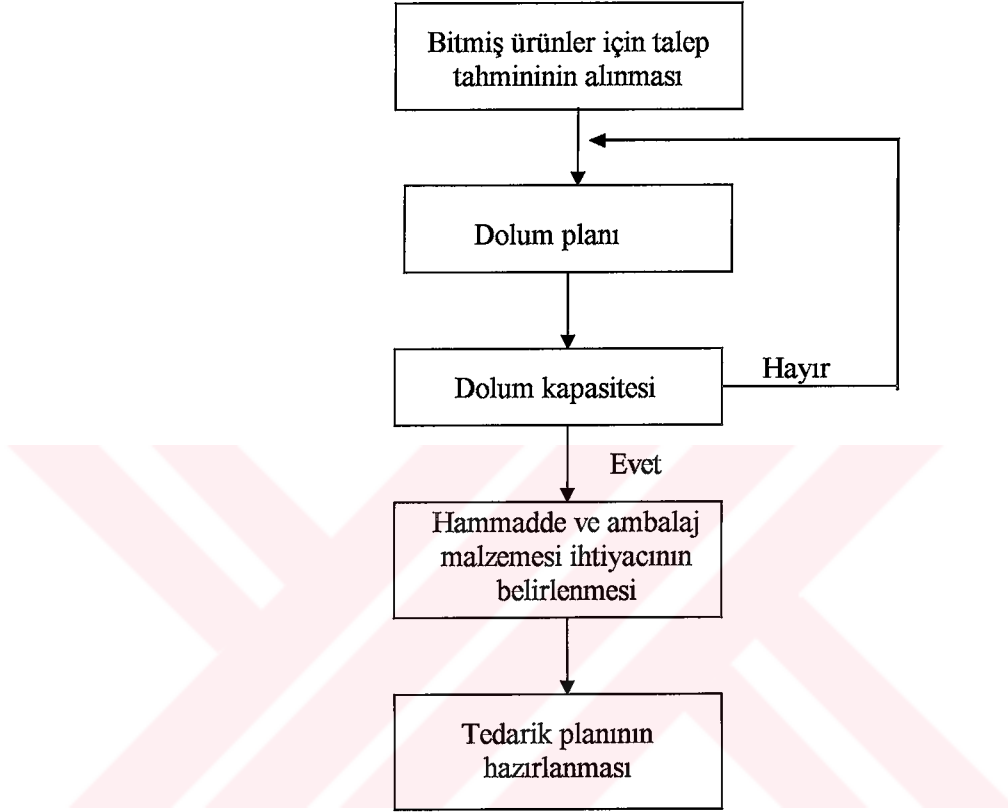
Halen üretim planı el ile yapılmaktadır. Hammaddelerin bir çoğunun yurt dışından gelmesi ve temin sürelerinin 2-3 ay olması, siparişlerin önceden iyi bir şekilde verilmesini gerektirmektedir. Yıllık tedarik planı önceden hazırlanmakta ve yıl boyunca gözden geçirilmektedir. Bitmiş ürün envanterini yönetmek için hedef envanter seviyeleri, pazar taleplerini karşılamak üzere belirlenmiş durumda ve üretim planı hazırlanırken bu kısıtlar dikkate alınmak zorundadır.

4.3.1.1. YILLIK VE ORTA VADELİ PLANLAMA

Yıllık üretim planı, zaman birimi olarak ayları dikkate alınarak hazırlanır ve bütçeyi tanımlamakta ve yıllık tedarik planı hazırlamakta yardımcı olur. Orta vadeli plan, 3 aylık dönemleri kapsar. Şekil 4.2 üretim planının aşamalarını göstermektedir.

Satış-pazarlama departmanından alınan talep tahminlerine göre dolum planı hazırlanır. Bu, karıştırma işlemi için de ihtiyaçları belirler. Son olarak, üretim planına göre hammadde ve ambalaj malzemesi ihtiyaçları hesaplanır. Bu ihtiyaçlara göre yıllık tedarik planı, tedarik sürelerini ve ekonomik sipariş miktarlarını dikkate alarak hazırlanır.

Orta vadeli planlama için, tedarik planı revize edilir, siparişlerin ertelenmesi veya öne alınması alternatifleri değerlendirilir.



Şekil 4.2. Üretim Planının Hazırlanması İçin Bir Akış Şeması

4.3.1.2. KISA VADELİ PLANLAMA

Kısa vadeli planlama aşağıdaki sorulara cevap verir:

1. Ne üretilmeli?
2. Ne kadar üretilmeli?
3. Ne zaman üretilmeli?

4. Hangi kaynakları kullanarak üretilmeli?

Uzman sistem, planlamacının kullanabileceği otomatik ve etkileşimli bir dizi araçlar sunar. Özenle hazırlanmış bir plan şu safhalardan geçer:

1. Planda yer alacak ürünlerin seçimi
2. Üretim miktarlarının tanımı
3. Kaynak dağıtımı ve sıralama
4. Global geçerlilik testi

İlk evrede planlamacının ilgilenmek zorunda olduğu ürünlerin sayısı azaltılarak ve üretilmeyeceğinden emin olduğu ürünler çıkartılarak, seçilmiş ürünler seti (SÜS) tanımlanır. Bunu yapmak için planlamacı, seçilmiş her ürün için üretim öncelikleri endeksi (ÖE) belirler. Bu endeksi belirlemek için seçilmiş ürünler setindeki her bir ürün için, teslim tarihini karşılamak amacıyla üretimin başlangıcına kadar geçmesi gereken süre (BS) hesaplanır. Bu sürenin hesaplanmasında üretim süreleri ve mevcut stok durumları dikkate alınır.

$$BS = (\text{siparişin termin tarihi} - \text{içinde bulunulan tarih}) - \text{üretim süresi}$$

$$\text{Üretim süresi} = \left\{ \left((\text{hazırlık süresi} + (\text{birim işleme süresi} * \text{lot büyüklüğü})) / \text{operatör sayısı} \right) / 60 \right\} + \text{bekleme süresi} + \text{taşıma süresi}$$

Eğer bir ürün için BS negatif bir değerse o ürün hemen programa alınır. Eğer iki veya daha fazla ürün için BS negatif ise, negatif değeri büyük olan ürüne yüksek öncelik verilir.

İkinci safhada, hedef stok ile mevcut stok arasındaki fark ekonomik parti büyüklüğüne bölünerek, kaç parti ürün üretilceği belirlenmektedir.

Üçüncü safha (atama ve sıralama) için, öncelik endeksine göre ve birbirleri ile uyumları dikkate alınarak, ürünler karıştırma tanklarına atanır. Daha sonra dolum makinalarından bu karışımdan elde edilecek paketlenmiş ürünler programlanır.

Son safhada, planın global geçerliliği, planlama prosesi süresince hammaddelerin ve malzemelerin elde edilebilirliği, donanım kullanımı açısından test edilir.

Planlama prosesi, bu dört evrenin sıralanması şeklinde olmakla birlikte, planlamacılar her zaman bu sırayı takip etmezler. Aktif bir süreçte, geri besleme çevrimleri vardır. Belirli bir ürünü seçip, 2. ve 3. adımı uygulayıp 1. adıma geri dönmek, sonra diğer ürünü seçmek gibi yaklaşımlar da kullanılabilir.

4.3.2. KULLANICI ARABİRİMİ

Kullanıcı arabirimi bir uzman sistemin ana bileşenidir. Moreira ve Oliveira (1991) tarafından belirtildiği gibi "Kullanıcılar tarafından görülen sistem, arabirimdir. Kullanıcılar, akıllı kodlamayla, hiyerarşik yapılanmayla, ilgili veri tabanlarıyla, yazılımlarla ilgisizdir. Arabirimin kalitesine oldukça duyarlıdır". Bir planlama sistemi için geliştirilecek uzman sistemde arabirimin tasarlanmasında şu noktalar önemli olacaktır:

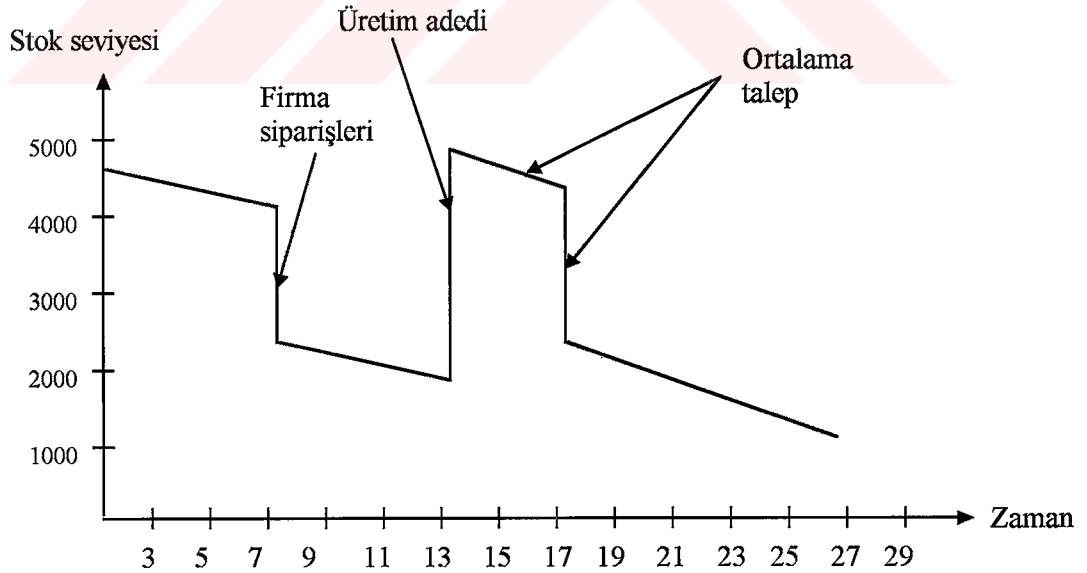
1. Kullanıcı yakınlığı olması ve değişik problem çözme taktiklerine uyarlanabilen şahsi arabirimler sağlayabilmesi.
2. Karar vericilere etkili destek sağlaması, kullanıcıya bilgi karışımını seçme imkanı vermesi.
3. Güçlü olması, girdi hatalarını hoşgörmesi ama iç kontroller ihtiva etmesi ve kullanıcıları hata yapmaktan koruması.

Uzman sistem, kullanıcıya, her planlama evresinde ihtiyaçlarına en iyi uyan bilgi karışımını ve formatını (tablo veya grafik) seçme imkanını verecek şekilde dizayn edilecek bir gösterim sistemini de içermelidir. Bu, karşılıklı etkileşimlerle kullanıcıya potansiyel kararların sonuçlarında hazır feedback sağlar. Örneğin, Tablo 4.1 son

mamuller, mevcut stok seviyeleri, hedef stok, müşteri talepleri, üretilecek miktar, ekonomik parti büyüklükleri ile ilgili bilgiyi göstermektedir. Gösterim sisteminin ana fonksiyonu, planda yer alması gereken ürünlerin seçimidir. Kullanıcının ürünleri dahil etme veya hariç tutma olasılığı vardır. Sistem, ayrıca üretilmesi gereken miktar, müşteri talepleri hakkında ilave bilgi elde edilmesi veya ürün stok seviyelerinin zaman içindeki değişimini gösteren bir grafik tanımlayabilmelidir (Şekil 4.3).

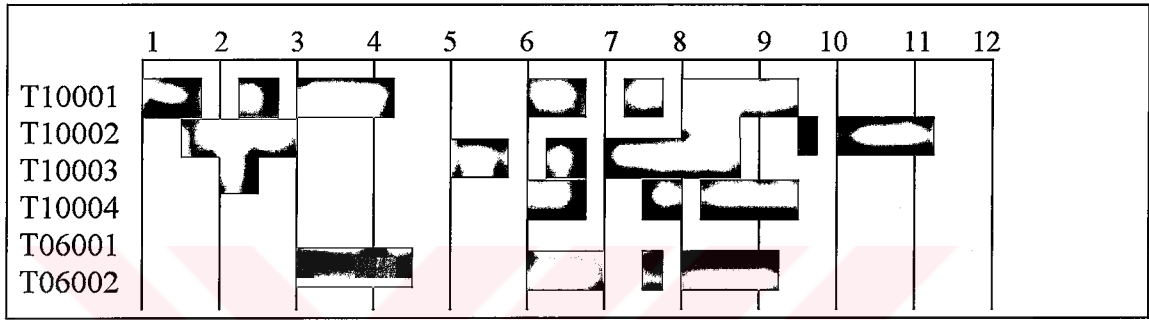
Tablo 4.1. Seçilmiş Ürünlerin Bir Seti

ÜRÜN ADI	HEDEF STOK	MEVCUT STOK	FİRMA TALEPLERİ	ÜRETİLECEK MİKTAR	PARTİ BÜYÜKLÜĞÜ	PARTİ SAYISI
5 % DEXTROSE 100 ML	115,500	58,500	231,000	288,000	18,480	16
10 % DEXTROSE 150 ML	4,800	1,150	9,600	13,250	16,896	1
50 % DEX 500 ML	9,100	7,800	18,200	19,500	8,448	2
ISOLYTE P 250 ML	3,320	2,840	6,640	7,120	9,856	1
ISOTONIC NAACL 3000 ML	5,130	682	10,260	14,708	2,112	7
ISOLYTE S 1000 ML.	15,050	6,340	30,100	38,810	4,752	8
LACTATED RINGER 2000 ML	6,650	2,100	13,300	17,850	2,112	8



Şekil 4.3. Ürün Stok Seviyelerinin Zaman İçindeki Değişimi

Sistem, karıştırma tanklarının zaman içinde kullanımını gösteren, üretilecek ürünlerin tanklara atanması ve sıralanması işlemleri için kullanılacak Gantt şemaları geliştirebilmelidir (Şekil 4.4). Böylece planlamacı bir partiyi bir tanktan diğerine taşıyabilmeli veya bir tank için sıralamayı değiştirebilmelidir. Dolum makinaları ve steril tanklarının kullanımını gösteren benzer şemalar kullanıcı için karar vermede etkin görev yapacak gösterimler olacaktır.



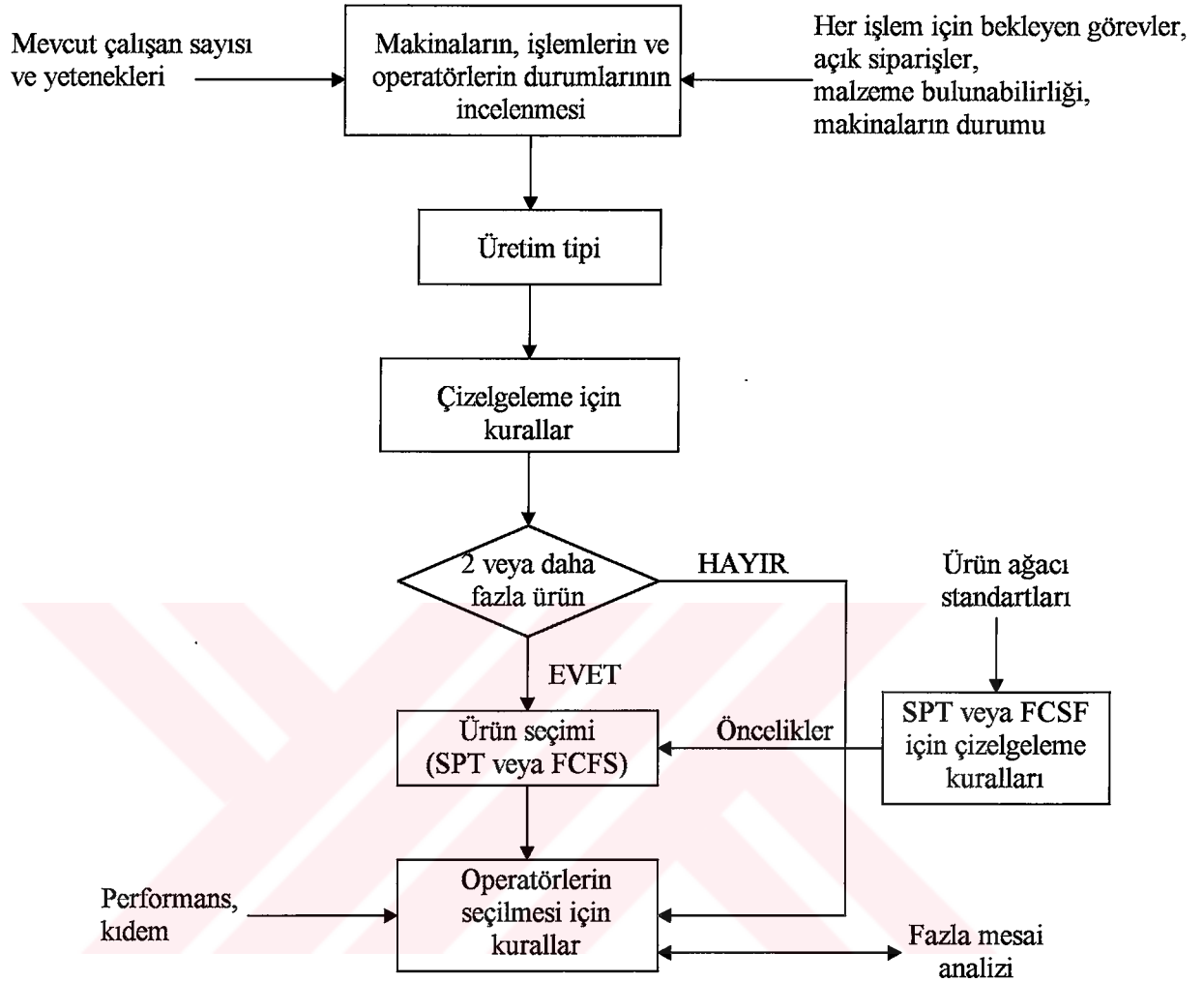
Şekil 4.4. Karıştırma Tanklarının Kullanımı

4.3.3. MODEL TABANI

Bir planlama sisteminin aşağıdaki 3 fonksiyonel alanı içermesi beklenir:

1. Hammadde ve malzeme envanteri ve tedarik yönetimi
2. Üretim planlama ve kontrol
3. Bitmiş ürün envanteri, müşteri siparişleri ve dağıtım yönetimi

Bunlar birbirleriyle çok sıkı ilişkileri olan yönetimlerdir. Tüm bu yönetimlerin başlangıç noktası ana plandır. Ana plan başlangıçta ne kadar üretim yapılacağını ortaya koyan ve uzman bilgilerin en detaylı bir şekilde ortaya dökülmesi için bulunmaz bir sahadır. Bu sahada bilgi temini daha kolaydır ve bilgilerin güvenilirliği üretimin diğer karmaşık safhalarından daha fazladır. Üretim planlama için geliştirilecek uzman sistemin model yapısı Şekil 4.5'teki gibi olmalıdır.



Şekil 4.5. Üretim planlama için geliştirilecek uzman sistemin model yapısı

4.3.4. VERİ TABANININ OLUŞTURULMASI

Veri tabanı kavramı, bilgi tabanı kavramı ile karıştırılmamalıdır. Uzman sistemlerde kullanılan bilginin türü, normal veriden farklı olduğu için, burada kullanılacak olan bilgi birikimine bilgi tabanı ve optimal rakamları bulunacak olan sistemin gerçek dünya girdilerine de veri tabanı adı verilmiştir. Kural tabanlı uzman sistem teknolojisinde,

mantık yürütücünün, bilgi tabanındaki kurallar kümesi ile veri tabanındaki durumlar kümesini birlikte kullandığını tekrar belirtelim. Veri tabanı bilinen durumlar kümesidir. Ancak çözüm ve sonuçlar, uzman bilgiye dayandırılmış olan bilgi tabanına, bir başka deyişle kurallar zincirine dayanmaktadır. Verinin yanlış olarak girilmesi elbetteki çözüm sonuçlarının yanlış olmasını doğurur.

Oluşturulacak bir veri tabanında aşağıdaki bilgilerin yer alması gereklidir.

1. Planı yapılacak dönemle ilgili bilgiler: Bu bilgiler, dönem adı, çalışılan gün sayısı, günlük çalışma saati, vardiya sayısı, işçi sayısı, fazla mesai kapasitesinden oluşur.
2. Maliyet bilgileri: *Normal mesai maliyeti*, işçilik, kullanılan tesis, malzeme gibi unsurlardan oluşmaktadır. *Fazla mesai maliyeti*, normal mesainin yetmediği üretim seviyelerinde ortaya çıkan bir maliyet türüdür. İş gücü düzeyini sabit tutarak üretim hızını arttırmak, işçilerin fazla mesai yapmaları ile mümkündür. Fazla mesai üretimi arttırmak için yapılmaktadır ama bu da işçilerin yapabileceği üretim miktarı ile sınırlıdır. Bu sınırdan fazla üretime ihtiyaç varsa, işçi alınarak veya vardiya sayısı arttırılarak karşılanma yoluna gidilmektedir. İşçi almak veya vardiya sayısını arttırmak da ek maliyetler getirmektedir. Talebin karşılanamaması durumunda *stoksuz kalma* maliyeti oluşacaktır. Bu, müşterilerin taleplerinin rakip firmalarca karşılanması anlamına gelir. *Stok elde bulundurma maliyeti*, stok miktarıyla doğru orantılı olarak değişecektir.
3. Ürün ağacı bilgileri: Her bir ürünü oluşturacak hammadde ve ambalaj malzemesinin birim ürün için gerekli miktarları, isim ve kodları, gerekli işçilik ve makina saatleri ürün ağacı bilgileri olarak veri tabanında yer almalıdır.
4. Talep bilgileri: Dönem içinde üretilmesi gerekli rakamları hesaplamak için gerekli olacak talep tahminleri, ürün adları ve kodlarıyla birlikte sisteme girilir.

4.4. ÜRETİM PLANLAMA İÇİN GELİŞTİRİLMİŞ UZMAN SİSTEM ÖRNEKLERİ

4.4.1. ReDS (Requirement Driven Scheduling)

ReDs, kütle üretimi veya atölye üretimi gibi değişik üretim sınıflarında çalışması için tasarlanmış bir uzman sistemdir. ReDS'nin incelediği programlama işleri, kapasite planlama, sonlu kapasite ile üretim, her makina için sıralama kurallarıdır. Çoklu amaçların yanında ReDS, proses planı değişiklikleri, yeniden işleme gibi dinamik durumlarla da ilgilenir.

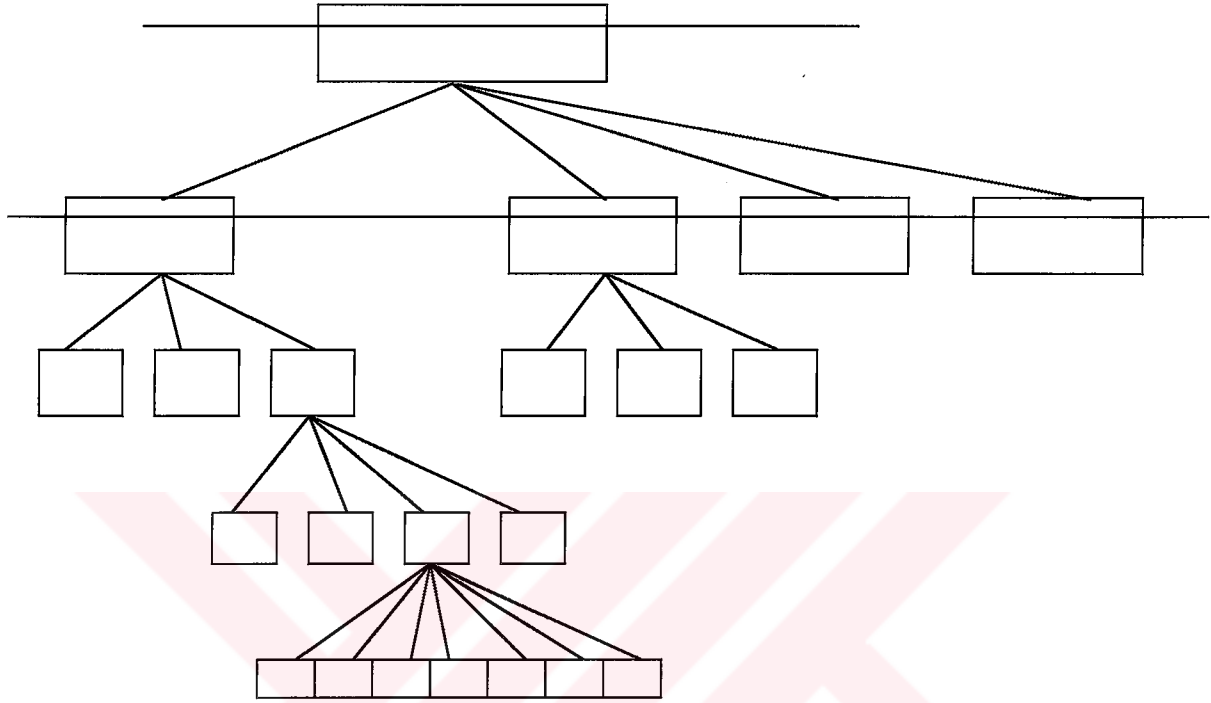
4.4.1.1. ReDS'NİN PRENSİPLERİ

ReDS'nin ilk görevi, idari amaçları karşılarken, programlama işini yapmaktır. Yaratılan programlar fizibildir ve programlayıcı üretimde oluşacak değişikliklere müdahale edebilir. İdari amaçlar, termin tarihlerini karşılamak, ürün envanter seviyelerini azaltmak, ürün çevrim süresini düşürmek, makina kullanımını arttırmaktır. ReDS hem Yapay Zeka hem Yöneylem Araştırması tekniklerini entegre eder. ReDS'de kullanılan önemli tasarım prensipleri şunlardır (Hadavi, 1990):

a. Kısıtların ve Zamanın Ayrılması

Kısıtları ve programlamayı temsil ederken kullanılan temel yapı, geçici ağaç yapısıdır (Şekil 4.6). Her düğüm bir zaman aralığına karşılık gelir ve alt düğümler de bu aralığın alt aralıklarına tekabül eder. Her tip kaynağın kendine ait geçici kısıt ağacı vardır. Programlama da geçici programlama ağacı olarak temsil edilir. Geçici kısıt ağacında, her (i) düğümünün bağlı olduğu kısıt havuzu $CP(i)$ vardır. i 'nin her j alt kümesi için a_{ij} , $CP(j)$ 'den $CP(i)$ 'ye yol vardır. Bu yollara ayırma fonksiyonu denir. Ağacın değişik seviyelerindeki kısıt havuzları, ayırmanın değişik seviyelerini yansıtır. ReDS gelecekteki

faaliyetleri planlamak için, kısıtları en üst seviyede kontrol eder, yakın gelecekteki programlama için kısıt havuzuna kadar iner ve programın fizibil olduğuna emin olur.



Şekil 4.6. Geçici Kısıt Ağacı

2. Siparişlerin Ayrılması

Sipariş sisteme girdikten sonra ve teslimata hazır olmadan önce, ReDS, sipariş üzerinde ön işlem yapar. Bu işlem, sipariş için mümkün başlama zamanlarını ayarlar ve sipariş için "gereklilik fonksiyonu" yaratır. Siparişin gereklilik fonksiyonu, siparişi tanımlamak için gereken kaynaklar ve kaynakların istenen teslim tarihini karşılamak üzere gerekli olduğu zaman aralıkları ile tanımlanır. Örneğin, darboğaz yaratan iş istasyonlarında gereken zaman, gereken envanter, siparişin gereklilik fonksiyonunun tanımlanmasına yardım eder. Bu gereklilik fonksiyonu, siparişin ayrılmasıdır.

3. Teslimat Kontrol Stratejisi

ReDS, FORCE (Factory Order Release Control and Evaluation) denen teslimat kontrol stratejisine sahiptir. FORCE, bir işi yapılacağı kesinleşinceye kadar üretime vermemeye çalışır. Ana amaç, çevrim süresini mümkün olduğunca azaltmaktır. Süreklilik endeksi, c_i , kavramı; - her adımdaki işlem süresi ve siparişin beklenen tamamlanma süresinin fonksiyonu - FORCE içinde tanımlanmıştır. Her siparişin çevrim ömrü boyunca ona bağlı olan süreklilik endeksleri vardır. Bu endeksler, zamanla, üretim ortamının dinamizmine göre değişir.

4. Çoklu Perspektifler ile Kısıt Yönetimli Sonuçlandırma

Her kaynağın geçici kısıt ağacı, tek bir kaynak profili; toplam geçici kısıt ağacı, toplam kaynak profili sağlar. Bu profili kullanarak, her kaynağın kapasitesini izleyen ve bir kaynağın aşırı yüklendiğini bulunca uyarı mesajı gönderen bir sistem tasarlanmıştır ve bu sisteme kapasite izleyici adı verilmiştir

Gereklik fonksiyonundaki bütün istenen kaynaklar incelenerek, sipariş perspektifi programlamacıya verilir. Programlamacı gelen bir sipariş için en erken fizibil programı belirler. Ters olarak, FORCE süreklilik endekslerini, siparişin çok erken teslim edilmemesini sağlamak için kullanır. Sipariş perspektifini kullanarak, sipariş izleyicisi siparişi sürekli izler ve durumu güncelleştirir. Bir olay olunca reaktif faaliyet en ilgili seviyede harekete geçer ve olayın etkisi diğer seviyelere aktarılır. Böyle bir olay, yeni sipariş girişi, makina bozulması, tedarikçi gecikmesi, makinanın aşırı yüklenmesi olabilir. Örneğin, bir makina bozulması olayında, kapasite izleyici, etkilenen siparişlere dikkat edilmesi için sinyal gönderir, sipariş izleyici, siparişin durumunu güncelleştirir. Eğer siparişin durumu önemli oranda değişiyorsa, az öncelikli olan bazı siparişler geciktirilir veya askıya alınır.

4.4.1.2. REDS'NİN KAVRAMSAL YAPISI

ReDS programlama işini dört alt göreve böler.

1. Ön işlemci
2. Fizibilite analizi (FA)
3. Detaylı çizelgeleyici (DS)
4. Sıralayıcı (SEQ)

Kavramları daha iyi anlaşılır hale getirmek için, yeni bir sipariş sisteme girdiği zaman neler olacağını anlatan bir örnek verilecektir. ReDs'nin iki modülü vardır. Önleyici modül ve reaktif modül. Öncelikle yukarıdaki dört alt görevin önleyici modül altında nasıl çalıştığı anlatılacak, sonra da reaktif modül işlemleriyle ilgili örnek verilecektir.

1. Önleyici Modül

Sipariş geldiği zaman, ReDS yeni siparişi, programın stabilitesini mümkün olduğu kadar koruyarak, mevcut programa yerleştirmeye çalışır. Bir program, yeni sipariş için daha az değişiklik gerektiriyorsa daha stabildir. Ön işlemci modülü, geriye doğru programlama yapar. Önemli bazı kısıtları da kontrol eder. Ön işlemlerden sonra, fizibilite analizi modülü, siparişin esasını teşkil eder. Siparişin esas demek, tüm kritik kaynak ihtiyaçlarının - darboğaz oluşturan iş istasyonları, pahalı olan ve zor elde edilen envanter kalemleri, kullanılacak aletler - toplamıdır. Fizibilite analizi safhası boyunca sistem, kullanıcıyla dialog halinde, sipariş ihtiyaçlarını karşılamak için yollar bulmaya çalışır. Bu bazı kısıtları (örneğin, termin tarihleri, vardiya ilaveleri, miktarlardaki değişiklikler gibi) rahatlatarak yapar. Ayrıca ReDS en düşük maliyeti arayan maliyet prosedürlerine dayanan kararlar da verir. FORCE teslimat modülü, fizibilite analizi modülü içine yerleştirilmiştir. FORCE gelen siparişler için optimale yakın teslim tarihlerini bulmaya çalışır. FORCE şu raporlardan birini üretir:

- Önemli problem bulunmadı.
- Bazı problemler bulundu fakat bazı kısıtların rahatlatılması problemi çözecek. Gereken çözüm tavsiye edilir.
- Bulunan problem çok önemli ki bu siparişi tanımlamak akıllıca değil.

Fizibilite analizinden sonra, detaylı çizelgeleyici modülü fizibilite analizi tarafından verilen tavsiye ile çalışır ve daha detaylı bir programı fizibilite analizinin gözden kaçırdığı tüm alt seviyeleri kontrol ederek hazırlar. Daha detaylı bu program, kısa dönemli olarak yapılır. Örneğin fizibilite analizi, aylık program hazırlıyorsa, detaylı çizelgeleyici, programın bir kısmını gün veya vardiya seviyesinde tanımlar. Bu strateji, programı değiştirmeyi kolaylaştırır ve programın revize edilmesini gerektirecek beklenmedik, fakat kaçınılmaz bir olay ortaya çıktığında geriye dönme olasılığını azaltır. Ayrıca, üretimdeki durum değişikliklerine bağlı olarak eskiyecek detaylı programları hazırlamak için harcanacak çabayı önler.

Son olarak sipariş, programdaki diğer tüm siparişlerle karışarak sıralayıcı modülüne geçer. Bu detaylı program işlemler, kaynaklar ve zamanın bir kümesidir. Verilen bir zaman periyodu ve kaynak için atanmış işlemler vardır. Sıralayıcının görevi, işlemlerin, hangi kaynaklara, ne zaman atanacağını bulmaktır. Sıralayıcı, lot kombinasyonu, hazırlık zamanları, kullanılacak makina seçimi ve problemlerin önlenmesi gibi alana özgü bilgi içerir. Sıralayıcının amaçları, makina kullanımını yükseltmek, akış süresini ve yarı mamul stoklarını azaltmaktır. Dinamik sıralama kuralı, sıralama için etkili öncelik kurallarını belirlemek için geliştirilmiştir. Dinamik sıralama kuralını uygulayarak, sıralama kararları, üretim ortamının sürekli değişen şartlarına cevap vermek için gerçek zamanlı olarak yapılmıştır. Sıralayıcının ürettiği program, üretimdeki işlemleri dakika dakika içerir. Sıralayıcının çıktısı, direkt olarak üretim sahasındaki operatörlere gider.

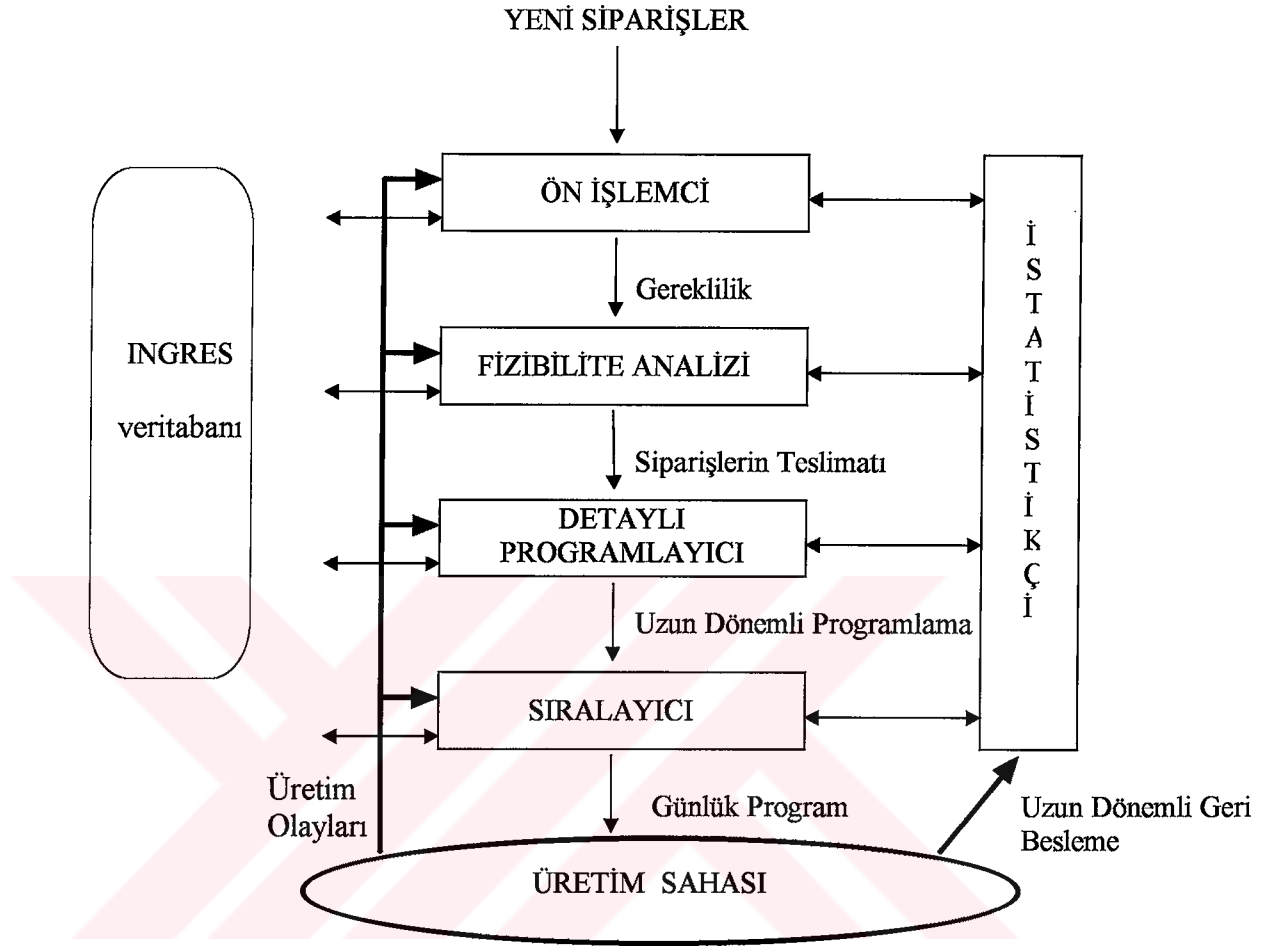
Özetle ReDS, siparişlerin ve kaynakların azaltılmasını ön işleme safhasında gerçekleştirir. Bu safhanın ana amacı teslim tarihlerini karşılamaktır. Fizibilite analizi, diğer idari amaçları (çevrim sürelerinin azaltılması, yarı mamul stoklarının ve bitmiş ürün envanterinin azaltılması gibi) karşılamak amacıyla teslimat kontrolü sağlar. Ayrıca en kritik kısıtlara dikkat eder ve diğerlerini detaylı çizelgeleyicinin kontrolüne bırakır. Detaylı çizelgeleyici, programın geçerliliğini sağlar. Sıralayıcı, dağıtım listesini gerçek zamanlı olarak alır ve optimal kararlar verir.

2. Reaktif Modül

Reaktif modülde, 4 modülün hepsi, üretim sahasından gelen şikayetleri dinler. Her biri, olayın karmaşıklığına bağlı olarak kendi yöntemine uygun olarak davranır. Örneğin makina bozulması olayında, sıralayıcı hemen lotları tekrar rotalamaya çalışır ve etkilenen lotların öncelik değerlerini değiştirir. Bu arada planlama modülü olan detaylı çizelgeleyici, bozulmanın etkilerini kontrol eder. Gerekli olursa, program detaylı çizelgeleyici tarafından düzenlenir. FORCE modülü, makina bozulmasının yarattığı problemin, ileriki siparişleri ne derece etkilediğini görmeye çalışır. Problem çözülünceye kadar yeni siparişlerin girişini sınırlandırır. Tüm bu faaliyetlerin amacı problemin kısa ve uzun dönemli sonuçlarını belirlemektir. Üretim sahasından gelen olayların ortalaması dakikada 50-100 arasındadır. Bu oran, günün belli zamanlarında 10-20 olaya kadar düşmektedir. Dağılım, üssel dağılıma uymaktadır. Üretim sahasından gelen bu olaylar arasında makina yükleme ve boşaltma, makina arızalanması, yeniden işleme, fire ve yeni siparişler yer almaktadır.

4.4.1.3. ReDS'İN MİMARİ YAPISI

Aynı çatıyı kullanan iki ReDS versiyonu uygulanmıştır. Her iki versiyonda da kullanılan yapı aynıdır. Geçici kaynak ayırma ağacının alt seviyesindeki şikayetler, hiyerarşinin alt seviyelerinde çözülür ve genellikle üst seviyelere yayılmaz. İlk versiyonun mimarisi, Requirement Driven Scheduling, Şekil 4.7'dedir.



Şekil 4.7. ReDS mimarisi

Bu yapı hiyerarşiktir, her modül, kavramsal çatıda tanımlanan çizelgeleme görevine karşılık gelir. Ancak sıkı bir hiyerarşi değildir, çünkü bilgi hiyerarşinin değişik seviyelerine direkt olarak geri beslenir. Karar vermenin en alt seviyesi olaylara ilk olarak müdahale eder ve mümkünse çare bulur. İstatistikçi modülünün amacı, uzun dönemli üretim problemlerini gözlemlemek, analiz etmek ve düzeltmektir. Problemin düzeltilmesi, istatistikçi tarafından iki yol ile tavsiye edilir.

1. ReDS'nin deęişik modüllerine bulunan bir darboğaz iş istasyonuyl ilgili sinyal vererek.
2. Yönetimi önemli kalite problemleri hakkında bilgilendirerek. Örneğın zamanın %60'ında çalışmayan bir makina raporlanacaktır.

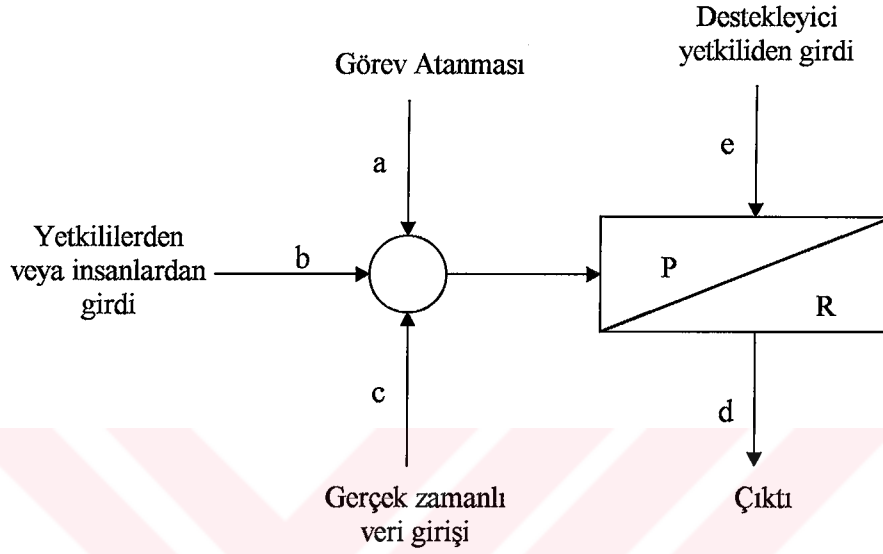
Bu yapı, uygulamalar için etkili olmasına rağmen, üretim bir çok açıdan aynı anda izlenemiyordu. Herhangi bir anda, ReDS tek bir alana dikkat edebiliyordu. Ortamın dinamiğı, daha fazla alana dikkat gerektirdiğı için ReDS² geliştirilmiştir.

4.4.2. ReDS²: REAL TIME DISTRIBUTED SCHEDULING

ReDS² 'de, planlama yetkilisi her modülü tasarlamak için jenerik yapı olarak kullanılmıştır. Her yetkilinin, kendine ait açıkça tanımlanmış görevi vardır. Sistemi tek bir mimari sınıfına kategorize etmek zor olduğı için, ReDS² , "katı hiyerarşı ile dağıtılmış sistem" olarak tanımlanabilir. Çoğı sistem gerçekten hibrittir, farklı görüşler incelendiğinde farklı yapılar sergilenir. Kapsam yapısı tipik bir örnektir. Sistem, kontrol için kullanıldığında hiyerarşik görünür, her kademe gerektiğinde bağımsız yetkili gibi davranır. Bazı durumlarda, bir kademe diğerleri üzerinde iletişimi engelleyerek kontrol koyar. Bu mimarideki ana zorluk, her kademenin diğerlerinin sorumluluğı ve otoritesi hakkında tam bilgiye sahip olması gerekliliğidir.

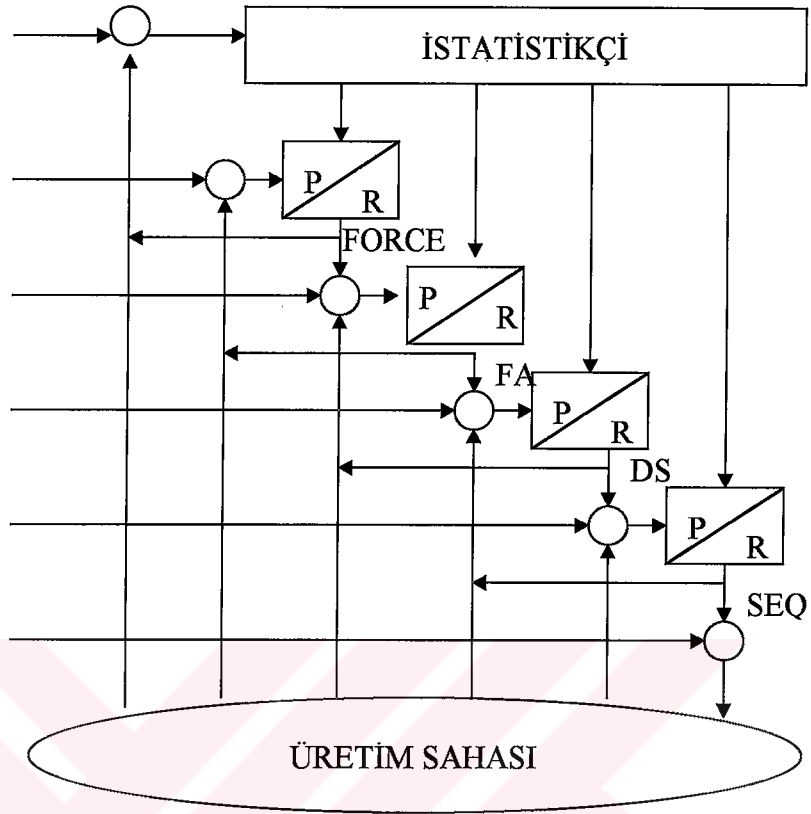
- **Planlama Yetkilisi:** Her planlama yetkilisi, çizelgeleme genlerinden, girdi ve çıktılarından oluşmuştur. Çizelgeleme geni iki bileşenden oluşmuştur. Önleyici eleman ve reaktif eleman. Önleyici eleman, rutin görevini yapar, reaktif eleman düzensiz gelen olaylara tepki gösterir. Her planlama yetkilisi, bağımsız ve özerk prostir. Görevleri, daha yüksek seviyelerdeki yetkililer tarafından atanır. Her planlama yetkilisi için girdi, insanlardan, diğer yetkililerden veya üretim sahasından gelen sinyallerdir. Her planlama yetkilisi, istatistikçi modülünden de girdi alır. Son olarak planlama yetkilisinin çıktısı alt seviye modülüne gönderilir. Bu çıktı, yüksek seviye modülü ve

istatistikçi modülü tarafından planlama yetkilisinin haberi olmadan gözlenir. Şekil 4.8 planlama yetkilisinin yapısını göstermektedir.



Şekil 4.8. Bir Planlama Yetkilisi

- **Tekrarlı Yapı:** Şekil 4.9, ReDS'nin kavramsal çatısına yerleştirilmiş planlama yetkilisinin tekrarlı yapısını göstermektedir. Her modül (ön işlemci, fizibilite analizi, detaylı programlayıcı, sıralayıcı) planlama yetkilisinin birer örneğidir ve özerk çalışır. Planlama yetkilileri arasında katı hiyerarşi vardır. Her modülün iyi tanımlanmış sorumluluğu olduğu için, kapsam mimarisinden daha iyi çalışır, gelen sinyallere bağımsız olarak tepki gösterir. Her modül diğerlerinin varlığından haberdar değildir. Bir yetkili diğerlerini kontrol etmediği zaman koordinasyon daha etkilidir.



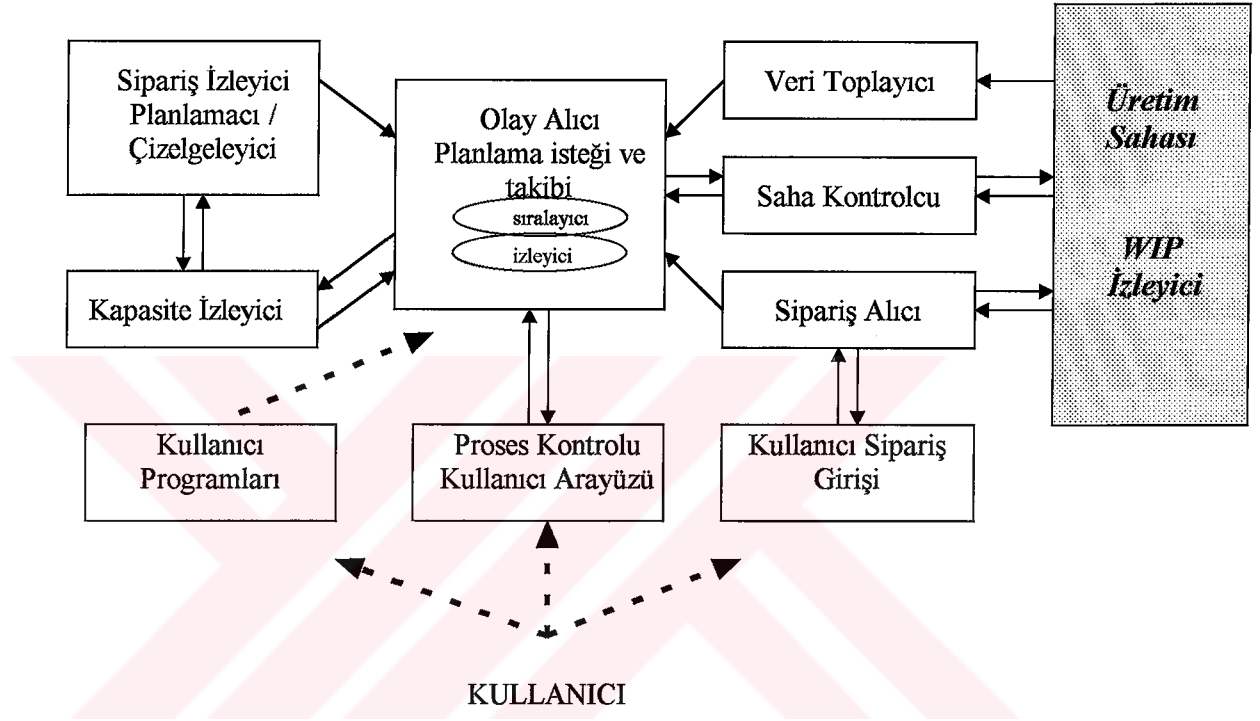
Şekil 4.9. Tekrarlı Yapı

4.4.2.1. ReDS² ÇEKİRDEĞİ

ReDS² açık bir sistem olarak uygulanır. Dış çevresiyle ticari bir veritabanı aracılığıyla iletişim kurar. Kullanıcılarla etkileşim içindedir ve etkileşimli çizelgelemeye izin verir. ReDS² ayrıca çalışma sırasında sistem yöneticisinin çağrısına göre davranışını değiştirebilir.

Bu bölümde, ReDS² sistem çekirdeğini ve dış çevre arayüzü anlatılacaktır. Sistem çekirdeği, altı sistem seviyesi yardımcısını kapsar: Üç çizelgeleme yardımcısı ve üç iletişim yardımcısı. Çizelgeleme yardımcılarını sipariş izleyici, kapasite izleyici ve olay

alıcıdır. İletişim yardımcıları ise veri toplayıcı, saha kontrolcu ve sipariş alıcıdır. Dış çevre ile olan arayüzler üç kullanıcı seviyesi arayüzleri ve yarı mamul (WIP) izleyici arayüzünden oluşur. COMETS, yarı mamul izleyici olarak kullanılan ticari bir yazılımdır. Şekil 4.10 ReDS²'nin fonksiyonel bir diyagramını göstermektedir (Hadavi et al., 1990).



Şekil 4.10. ReDS² Sistem Yapısı

4.4.2.2. ReDS² İŞLEMLERİ

Yeni bir sipariş, sipariş girişi kanalıyla sisteme ulaşır ve sipariş alıcıya yollanır. Sipariş alıcı ön işlemleri bu sipariş üzerinde yapar. Çıktı daha sonra olay alıcıya gönderilir. Bilginin bir kısmı (proses planı, ortalama gibi) ayrıca WIP izleyiciye de gönderilir. Olay alıcı, işlem öncesi bilgi ile siparişi alır ve bunları "iş havuzu" içinde biriktirir. Lot girişi veya çıkışı olduğunda, olay alıcı sipariş izleyiciyi çalıştırır.

Sıralayıcı olay alıcının bir parçasıdır. Dağıtım listesini her makina için ayrıntılı olarak üretir. Sıralayıcı ve kapasite izleyici, sipariş izleyiciye bir sinyal göndererek işi serbest bırakırlar. Sipariş izleyici, kapasite izleyici tarafından sağlanan kaynak kapasitelerini göz önünde bulundurarak iş havuzundan bir işi serbest bırakmaya çalışır. Serbest bırakma şartları sağlandığında, sipariş izleyici, işi sıralayıcıya bırakır; aksi halde bu şartlar sağlanıncaya kadar bekler.

Sipariş izleyici ve kapasite izleyici, planlama ve çizelgeleme için en önemli görevleri yerine getirirler. Sipariş izleyici diğer bütün siparişleri ve devamlılık endekslerinin izlerini takip eder. İşin ne zaman serbest bırakılacağını belirlediği gibi gerektiğinde ne zaman geri alınacağını da saptar. İki modülden oluşur: Planlama yardımcısı ve çizelgeleme yardımcısı. Planlama yardımcısı, geçici çizelge ağacının üst seviyeleriyle uğraşır. Çizelgeleme yardımcısı ise bu ağacın alt seviyeleriyle uğraşır. Lotun işlem rotası gibi detaylı kısıtlamalar, çizelgeleme yardımcısı tarafından ele alınır.

Kapasite izleyici her bir iş merkezinin kapasitesini güncelleştirir. Aşırı yüklenme veya az yüklenme kapasite izleyici tarafından belirlenirse, sipariş izleyicinin planlama modülüne yükleme dengesi için çağrı gönderilir. Kapasite izleyici ve sipariş izleyici beraberce kaynaklar ve siparişler için profiller sunarlar. ReDS² paralel olarak çalışmaları için birçok planlama yardımcısı ve kapasite izleyiciye izin verir. Bu, programın tam zamanında oluşması için iyi bir performans sağlar.

Olay alıcı, sistemin merkezidir. Diğer yardımcılarla ve üretim sahasıyla iletişim kurar. Lot takibini, sıralamayı, üretim sahasındaki harici olayları ele almayı, bütün üretim sahasına ait bilginin takibini ve veri tabanının güncelleştirilmesi işlemlerini yapar. Örneğin rotanın değişmesi gibi harici bir durum oluşursa, uygun yardımcıya gerekli bilgiyi ulaştırır.

Olay alıcının içine yerleştirilen sıralayıcı, lot dağıtım listesini isteğe göre oluşturur ve bunu üretim kontrolcu yoluyla üretime gönderir. Olay alıcı, herhangi bir

kullanıcının proses gereksinmelerini karşılamak üzere kullanılabilecek bir tercümana sahiptir. Bu tercüman fonksiyonel bir yazılımı girdi olarak alır ve önceden yüklenmiş bir kullanıcı kütüphanesi veya dış programlar kullanarak çevirme işlemini yapar. Bu yazılım LISP benzeri dilde kullanılan bir grup fonksiyondan oluşmuştur. Bu yaklaşım, sistem adaptasyonunu ve sürekliliğini sağlar. Çünkü:

- Her fonksiyon bir bilgi kümesini temsil eder. Bu bilgi, bir operasyonel strateji, sezgisel teknikler seti veya rota verisi olabilir. Fabrikanın kontrol bilgisi böylece ReDS çekirdeğinden ayrı olan bir yazılımın içinde tutulabilir.
- Tercüman IF ... THEN kuralının yapısına sahiptir. Bu yapı aşağıdaki durumlarda kullanılır.

1. Üretimde oluşacak olayları tanımlamak için.
2. Gerçek zamanlı işlemleri çizelgelemek için.

Veri toplayıcı, saha kontrolcu ve sipariş alıcı, kullanıcı arayüzü iletişim işlemleridir. Veri toplayıcı ReDS'ye dış çevreden bir giriş yoludur. Yarı mamul izleyici, veri toplayıcıya veri yollar. Sonra ReDS için veriyi formatlar. Saha kontrolcu, atölye faaliyetlerini veya atölyenin durumunu kontrol eder. Sipariş alıcı, kullanıcı, üretim sahası ve ReDS çekirdeği arasındaki arayüzleri korur. Sipariş bilgisinin üretim sahasına ve ReDS çekirdeğine gönderilmesini tasdik eder.

Kullanıcı seviyesi arayüz programları üç tiptir. Kullanıcı programları, üretim faaliyetleri monitörü ve sipariş girişi kullanıcı arayüzü. Üretim faaliyetleri monitörü sistem yönetimi için kullanılır. Kullanıcı, ReDS yazılım işlemlerini gözleyebilir ve bu işlemlerle onlara mesajlar yollayarak iletişim kurar.

Üretim faaliyet monitörü esasında amaçları geliştirmek için tasarlanmıştır. Ancak, kullanıcı isteklerini karşılayacak şekilde fonksiyonlar eklenmiştir.

Sipariş girişi kullanıcı arayüzü, kullanıcıya siparişleri yaratırken yardımcı olmak üzere tasarlanmıştır. Kullanıcı programları, çizelgeleme takvimleri oluşturmak gibi, destek görevleri için kullanılır. Başka bir kullanıcı programı, kullanıcıya siparişleri elle çizelgelemesine izin verir.

4.4.2.3. ReDS'NİN İTME ve ÇEKME DAVRANIŞI

ReDS'nin itme ve çekme davranışı, Just-In-Time sistemine çok benzer. İş siparişleri, işlenmeye hazır oldukları an üretim sahasına iletilir. ReDS², üretim sahası etkin olarak bir çok işi işlemeye hazır olduğunda siparişleri iş havuzundan çeker. Planlama seviyesinde, olay alıcı işi serbest bırakma isteklerini düzenli olarak yerine getirir. Bu "itme" hareketine benzerdir. Kapasite izleyici, kaynaklardan olduğundan az yararlanıldığını saptarsa, siparişi işlenmek üzere çeker. Çizelgeleme safhasında, sipariş eğer zamanında karşılanamayacaksa önceliği artar, böylece sipariş "itilir" ve diğer siparişlerin önüne alınır. Tersine, siparişin çizelgelenen zamandan önce bitirileceği tahmin ediliyorsa, önceliği azalır ve "çekilir", böylece kritik kaynaklar yüksek önceliği olan diğer siparişlere dağıtılabilir.

ReDS önceden küçük atölyelere göre tasarlanmasına rağmen, şu anda VLSI pilot hatları için kullanılmaktadır. Pilot hat, üretim kolaylığı için bir maskedir ve bir prototipi PCB montaj hattı için yapılmıştır. VLSI hattı iyi çizelgeler üretmek için en zor alandır. Böyle ortamlarda, proses planları ve işleme zamanları belli belirsiz tanımlanır, makina bozulmaları siktir, yeniden işleme çoktur ve ürünler makinalara karşı hassastır, bu yüzden hangi makinanın seçileceği dinamik olarak değişir. Bundan başka bu hatta, birçok darboğaz yaratan makina mevcuttur.

4.4.3. AERPLAN

Yalnızca insanların sonuçları görerek, öncelikli seçimleri yapmasını sağlayan sistemlere sahip stratejik üretim planlama destekleyicileri çoğu zaman yeterli değildir. İyi

bir destekleyici plan, düşük seviye kararları alabilmelidir. Bu çeşit karar alma işlemlerinin karmaşıklığı, sezgisel tekniklerin araştırılmasını gerektirir, çünkü kesin metodların izlenebilmesi mümkün değildir. Fakat, üretim planlamaya ve çizelgeleme problemlerine uygulanan sezgisel metodlar, genelde çok iyi işlev görseler de sık sık kötü davranışlar ortaya çıkarırlar. Etkileşimli planlama ve çizelgeleme teknikleri, problemi, otomatik problem çözümü sırasında insan operatörü de içeren en kötü durumları kontrol ederek temsil ederler. Bu yaklaşım, gelecek vaadedici görünse de, metodoloji yokluğu ve literatürdeki gerçek dünya örneklerinin kısıtlılığı, bu tekniklerin yaygınlığını engellemektedir. AERPLAN projesi, etkileşimli problem çözme prensibinin uygulanmasına işlerlik kazandırmıştır. Bu sistem, dört yıllık çalışma sonunda, Mart 1990 sonunda tamamlanmıştır ve AERITALIA adında, üretimi yedi fabrika arasında dağıtılmış bir uçak üretim şirketine yerleştirilmiştir. AERPLAN altında yatan gerçek düşünce, otomatik problem çözme becerilerinin, insan operatör fırsatından yararlanarak geliştirilmesidir (Bertolotti, 1990).

4.4.3.1. PROBLEM ALANININ ANALİZİ

AERITALIA, yedi ayrı fabrikaya sahiptir ve bunların her biri uçağın ayrı bölümlerini üretir. Planlama uzmanları, orta vadeli bir planı, pazarlama bölümünden her ay alınan satış rakamlarına göre güncelleştirirler. Yeni planı hazırlarken, mevcut iş gücünü gerekli iş yüküne dengeli bir biçimde dağıtmak zorundadırlar. Bu, gereğinden aşırı veya az iş yükleme problemlerinin son bulması demektir. Bu problemler, yeni aylık iş yükü dağıtımı ve bir fabrika için mevcut insan kaynakları arasındaki farktan kaynaklanmaktadır.

Az yükleme problemi, bir veya daha fazla üretim lotunun, fason üreticilerden fabrika içine alınmasıyla çözülebilir. Aşırı yükleme problemi ise, aynı hareketle fakat ters yönlü bir işlemle çözülebilir. Yani, üretim fazlalığı fasonda yaptırılabilir. İlk hareket operasyonu "üretmek" olarak, ikincisi ise "satın almak" olarak adlandırılır. Bir çok "üretmek" veya "satın almak" hipotezleri verilen bir probleme çözüm bulacak şekilde değerlendirilmelidirler. Her hipotez için bir çok hesap yapılmalıdır ki, iş yükü dağılımına

karşı insan kaynakları kapasitesi, üretimi gerçekleştirebilsin. Bu fabrikada yaklaşık 1500 farklı ürün çeşiti yedi değişik teknolojik alanı içermekteydi ve birkaç aydan bir yıla kadar değişen zaman aralığında üretilabiliyordu. Bir ürün fason üreticileri de içeren farklı fabrikalarda üretilmektedir.

Bir fabrika içi iş yükü dengesini ayarlamak, "üretmek" veya "satın almak" hareketleriyle oldukça karmaşıktır. Az veya aşırı yükleme problemlerini aşmak için planlayıcıların, hangi ürünlerin fabrika içerisinde üretilmesi gerektiğini (üretmek), hangilerinin dışarıda üretilbileceğini (satın almak) ve her ürün için hangi iş yükü dağılımlarının doğru olduğunu karşılaştırmaları gereklidir. Bunların dışında, planın iyiliğini ölçmek için bir çok çelişen amaçların varlığı sebebiyle kesin optimal bir karar setinin tanımlanması hemen hemen imkansızdır.

Üç kişiden oluşan bir takımın, bütün iş yükü kısıtlarını tatmin eden iyi bir planı hazırlaması üç ile beş gün sürmektedir. Bu insanlar, çeşitli alternatifler yaratıp seçim yapabilmek için, daha güçlü hesaplama araçlarına ihtiyaç olduğunu belirtmişlerdir. Bu ihtiyaçlar, yöneticileri, bilgi tabanlı bir sistem olan AERPLAN'ı planlamacılara işlerini yapmakta yardımcı olması için geliştirmeye sevk etmiştir.

İç iş yükü AERITALIA'nın yedi fabrikasından birinde üretilen iş yükünün bir kısmıdır, dış iş yükü ise, fason üreticiler tarafından üretilen iş yükünün bir bölümüdür. İnsan gücü eğilimleri ve iç iş yükü dağılımları arasındaki karşılaştırmalar, az veya aşırı yükleme problemlerini ortaya koyar. Planlama uzmanlarının amacı, bütün bu az veya aşırı yükleme problemlerini aşmaktır. Bir problem, eğer eldeki iş gücü ve gereken iş gücü arasındaki fark sabit olarak belirli bir değerin altındaysa çözülebilir kabul edilir. İnsan planlayıcılar tarafından uygulanan problem çözme prosedürü "üretmek" veya "satın almak" olarak adlandırılır ve problem içerisindeki üretim lotlarının iç ve dış iş yükü yüzdelerini ayarlanmasından oluşur.

İş ve dış üretim lotunun yüzdesini değiştirmek demek, lot iş yükünün aylık dağılımını günün ihtiyaçlarına uygun hale getirmek demektir. Bunun sonucu olarak bütün

fabrikanın aylık iş yükü dağılımı dağıtılır. Şekil 4.11 üç üretim lotunun iş gücü dağılımının yarattığı aşırı yükleme probleminin bir başlangıç örneğini göstermektedir. Teknolojik alan-1'deki aşırı yükleme problemi, üretim lotu-3'ün tamamen satın alınması ile çözülebilir. Lojistik sebeplerden dolayı, tüm lotun her iki teknolojik alanda da satın alınması tercih edilir. Bu son tercih teknolojik alan-2'de az yükleme problemi yaratır. Bu basit örnek bile farklı yerel çözümlerin diğerini nasıl etkilediği hakkında genel bir fikir vermektedir. Gerçek durumda, üretim çeşitliliğinin binlerle ifade edildiği ve problem çeşidinin on ile elli arasında değiştiği düşünülürse, planlamacıların karşılaştığı güçlükler anlaşılabilir.

Teknolojik Alan-1

	OCAK	ŞUBAT	MART	NİSAN	MAYIS	HAZİRAN	TEMMUZ
Üretim Lotu-1	---	---	0.6	0.6	0.7	0.7	0.7
Üretim Lotu-1	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Üretim Lotu-1	1.8	1.7	0.6	0.6	---	---	---
TOPLAM LOT	5.8	5.7	5.2	5.2	4.7	4.7	4.7
KAYNAKLAR	4.0	4.0	4.7	4.7	4.7	4.7	4.7
FARKLAR	-1.8	-1.7	-0.5	-0.5	0	0	0

Teknolojik Alan-2

	OCAK	ŞUBAT	MART	NİSAN	MAYIS	HAZİRAN	TEMMUZ
Üretim Lotu-1	---	---	0.5	0.5	0.7	0.7	0.7
Üretim Lotu-1	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Üretim Lotu-1	0.5	0.5	0.2	0.2	---	---	---
TOPLAM LOT	3.5	3.5	3.7	3.7	3.7	3.7	3.7
KAYNAKLAR	3.5	3.5	3.7	3.7	3.7	3.7	3.7
FARKLAR	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Şekil 4.11. Başlangıç İçin Bir Örnek

Basit bir örnekle alanı tanıttıktan sonra, planlama uzmanlarının "üretmek" veya "satın almak" işlemleri sırasında karşılaştıkları gerçek problemlere yakından bir göz atmak yerinde olacaktır.

1. "Üretmek" veya "satın almak" hareketleri için potansiyel adaylar olan ve az/aşırı yükleme problemlerinde yer alan üretim lotlarının sayısı yüzlercedir.
2. Problemin çözümü her zaman birden fazla hareketi gerektirir.
3. Belirli bir problemi çözmeye girişimi, başka az/aşırı yükleme problemleri ortaya çıkarabilir. Bunun nedenlerinden biri, yukarıdaki örnekte belirtildiği gibi, lojistik kısıtlamalardır.
4. Stratejik amaçlar (5b), çok sık olarak teknik amaçlarla (5a) çelişirler ve en iyi "üretmek" veya "satın almak" alternatifleri, bu iki amaç arasında en makul çelişkiye neden olacak şekilde üretilmelidirler.
5. Az/aşırı yükleme problemlerini çözecek hareketlere sahip lotları seçerken aşağıdaki teknik ve stratejik amaçlar dikkate alınmalıdır.
 - a. Önemli teknik amaçlar:
 - Bütün problemler mümkün olduğu kadar küçültülmelidir ve kısa vadeli problemler için daha iyi çalışacak çözümlere daha fazla önem verilmelidir.
 - Gerekli "üretmek/satın almak" hareketlerinin sayısı en aza indirilmelidir.
 - b. Önemli stratejik amaçlar:
 - Planlama uzmanları, üretim lotlarının şirketin stratejik olarak nitelediği teknolojik alanlara olan hareketlerini engellemelidir.
 - Değişik ürünler, değişik stratejik önceliklere sahiptirler ve en yüksek öncelikli olanlar mümkün olduğunca az hareket etmelidirler.

4.4.3.2. UZMAN SİSTEM NASIL YARDIM EDEBİLİR?

Problem, realistik iş yükü durumları için makul zamanlarda kesin olarak çözülememektedir ve sadece yaklaşık metodlar uygulanabilmektedir. Bir çok sezgisel

araştırma yaklaşımları içerisinde problem çözümü için, durum-yer-araştırma sistemi uygulanmıştır. Çünkü:

1. Durum-yer-araştırma sistemi, kısmi çözümlerin arındırılmasıyla global bir çözüm getirilmesine olanak sağlamaktadır. "Ya hep, ya hiç" yaklaşımlarının tersine, çözümler küçük küçük üretilmekte ve mümkün olan en kısa zamanda hazır hale getirilmektedir.
2. İnsan operatörle olan etkileşim, önceki noktada belirtilen karakteristiğe bağlı olarak basitleştirilmektedir.

Bu tür özellikler, çözüm üretme aşamasında araştırmayı yöneten insan tavsiyelerinden yararlanma avantajına sahip bir sistem geliştirmek için kullanılabilir. Durum-yer fikri, insan planlayıcının problem çözme sırasında izlenimlerini sunmasına olanak veren aktif bir sistem kurmayı sağlamıştır. Otomatik problem çözme mekanizması, kullanıcı ile üç ayrı seviyede iletişim kurabilir. Kullanıcı en uygun seviyeyi seçebilir. Diğer teknikler, sonlu kapasiteli üretim planlama problemlerine başarıyla uygulanmıştır. Yöneylem Araştırması, Dal ve Sınır Algoritması veya Tamsayılı Lineer Programlama metodları ile problem çözümlerine yardımcı olmaktadır. Bu örnekte, Yöneylem Araştırması tekniklerini takip etmek, gereken bütün ihtiyaçları tatmin etmekteki zorluktan dolayı gerçekten uzak görünmektedir.

Etkileşimli planlama ve çizelgeleme tekniği bir çok açıdan incelenmiştir. Etkileşimli çizelgeleme sistemleri kurmak için iki değişik yaklaşım vardır:

1. Etkileşimli yararlılık yaklaşımı: Bu alandaki çalışmaların çoğu, plan ve çizelgenin sürekliliği, gözlenmesi ve idaresine izin veren fayda ve aletlerin geliştirilmesi ile ilgilidir. Mesela plan, Bir Gannt diyagramı veya PERT grafiği aracılığıyla temsil edilebilir.
2. İşbirlikçi problem çözümü yaklaşımı: Bu alandaki çalışmaların amacı, insan programlayıcı ve makina programlayıcı arasındaki iş birliğini geliştirerek, her birinden

en iyi sonucu elde etmektir. Bu akımda, alandan bağımsız planlama sistemleri, amaçların yerine getirilmesi ve genişletilmesi için tasarlanmaktadır.

4.4.3.3. AERPLAN SİSTEMİNİN MİMARİSİ

Sistem mimarisinin tanımına geçmeden önce geçmiş bölümlerde geçen bazı terimlerin açıklamalarını vermek gerekecektir.

- Teknolojik alan: Her bir fabrikada farklı üretim seviyeleri uygulanır. Her seviye, homojen bir aktivite topluluğu olan ve eşit insan becerileri gerektiren teknolojik alana tekabül eder.
- Üretim lotu: Üretim lotu, bütün iş yükünü oluşturan temel üretim birimidir. Her bir üretim lotu aşağıdaki kriterlerle tanımlanır:
 - Üretilecek miktar
 - Başlama günü, bitiş günü
 - İş yükünün zamana dağılımı (her ay üretilecek miktar)
 - İç yüzde (AERITALIA'nın herhangi bir fabrikasında üretilecek yüzde)
 - Dış yüzde (Fason üreticilerin ürettiği yüzde)
- İnsan kaynakları: Verilen bir fabrikada ve teknolojik alanda mevcut iş gücüdür.
- Az/aşırı yükleme problemleri: Bir fabrikada iş yükünden kaynaklanan gerekli iş gücü ve mevcut iş gücü arasında fark varsa bir problem doğar. Problemin tanımlayıcıları:
 - Başlama günü, bitiş günü
 - Problem tipi (az yükleme, aşırı yükleme)
 - İçerilen teknolojik alanlar
 - Adam saati miktarıdır.
- "Üretmek" veya "satın almak" hareketleri: Bunlar bir veya daha fazla üretim lotunun iç ve dış yüzdelerini değiştiren operasyonlardır. Her bir hareket şöyle tanımlanır:
 - Yüzdelerinin değiştirilmesi gereken üretim lotlarının listesi
 - Üretim lotlarının ait olduğu uçak kısımları
 - Operasyonun uygulandığı teknolojik alanlar

- Yeni iç/dış iş yükü yüzdeleri

Her hareket aylık lot dağılımı değerlerinin değişmesine neden olur.

4.4.3.4. PROBLEM ÇÖZÜMÜ

"Üretmek" veya "satın almak" kararı safhasında, planlayıcının ana amacı, bütün az/aşırı yükleme problemlerinin aşılması ile gereken ve mevcut iş gücü dağılımı arasındaki farkları azaltmaktır. Bu, makul bir seviyeye ulaşıncaya kadar devam eder. AERPLAN bu görevde insan planlayıcıyı, iki farklı modülde destekler: Parça seviyesinde ve lot seviyesinde. Parça seviyesi modülü, verilen parçaya ait bütün üretim lotlarının yüzdelerini değiştirebilir. Lot seviyesi modülü bir veya birkaç üretim lotunun yüzdelerini değiştirebilir.

Tam bir problem çözüm dönemi genellikle parça seviyesi modülü ile başlar, bu modülde büyük ayarlamalar yapılır.

Parça seviyesinde "üretmek/satın almak" kararı: Planlama uzmanları, mevcut iş gücü ile gerekli iş gücü arasındaki büyük farklılıkları azaltmaya çalışırlar. Parça seviyesinde, mevcut iş gücü ve gerekli iş gücü toplamının her yılın sonunda her bir fabrikanın teknolojik alanları için dengelendiğini kontrol etmek yeterli olacaktır. Bunların dengelenmediği durumlarda sistem, planlayıcıya dengelenmemiş dağılımları gösterir ve "üretmek/satın almak" operasyonları ile hareket ettirilebilecek parçaları ve bunların bir sıralamasını tavsiye eder. Bu safha tamamlandığında bile, az/aşırı yükleme problemleri var olabilir, çünkü parça seviyesi süresince sadece yıllık toplamlar kontrol edilir. Oysa aylık dönemlerde problemler olabilir. Bu ayrıntılı problemler lot seviyesinde "üretmek/satın almak" kararları ile çözülecektir.

Lot seviyesinde "üretmek / satın almak" kararı: Lot seviyesinde problem çözümü, aşağıdaki ana kriterler izlenerek düzenlenmiştir.

1. Şu anda çözülmeye çalışılan az/aşırı yükleme probleminin tanımı (başlama günü, bitiş günü vb.)
2. Şu anda odaklanılan problemin neden olduğu işçilik saatindeki toplam fark
3. Hala var olan az/aşırı yükleme problemlerinin neden olduğu toplam fark
4. Şimdiye kadar uygulanan "üretmek" veya "satın almak" operasyonlarının sayısı

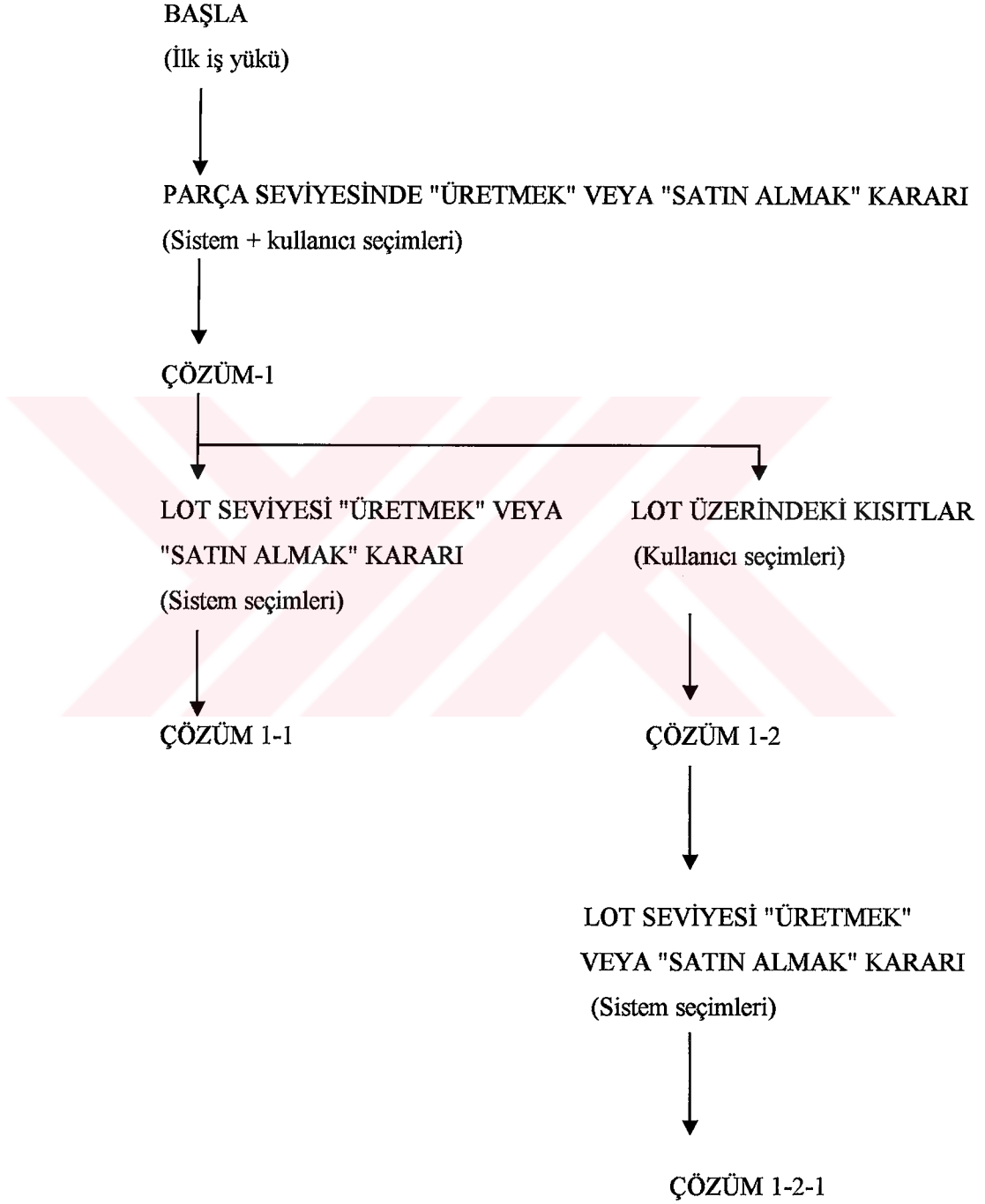
Şu ana kadar tam bir sonuca ulaşması için problem çözümünün alması gereken üç farklı karar grubu incelendi. İlk karar grubu, makul "üretmek/satın almak" hareketlerinin seçimi ile ilgilenir. İkinci grup kararlar, az/aşırı yükleme problemine uygulanan kısmi çözümlerin kabulü veya reddi ile ilgilenir. Üçüncü karar grubu, iş yükünde var olan az/aşırı yükleme problemlerine uygulanan iyi global çözümlerin seçimi ile ilgilidir. İnsan planlayıcı, sistemin kendi kendine alabileceği ve problem çözümü ile uğraşmadan önce kendisinin onaylamak isteyeceği karar grubunu seçer. İnsan planlayıcı, bütün işlem süresince karar alımında sahip olduğu gücü dinamik olarak değiştirebilir. Başka bir deyişle, planlayıcı, sistemin kendi kendini yönetme derecesini etkileyebilir. Her bir karar seti için kullanıcı etkileşiminin seviyesi belirlenmiştir.

Problem çözümü safhasında basitleştirilmiş ama gerçekçi bir problemden bahsedilecektir. Yedi fabrikadan sadece ikisi ele alınmıştır ve problemlerden de ikisi sunulacaktır. Son kullanıcıdan simülasyonu sürdürmesi istenmiştir. Burada AERPLAN sistemini kullanarak, insan operatörün bu iki problemi çözerken izlediği adımlar anlatılacaktır. Gerçek bir safha, yedi fabrikanın hepsini ve az/aşırı yükleme problemlerinin düzinelercesini içerecektir ve tamamlanması iki veya üç gün alacaktır.

Aşağıdakiler gerçek sistem çıktısının alındığı problem çözümü safhasının ana basamaklarıdır (Bertolotti, 1990):

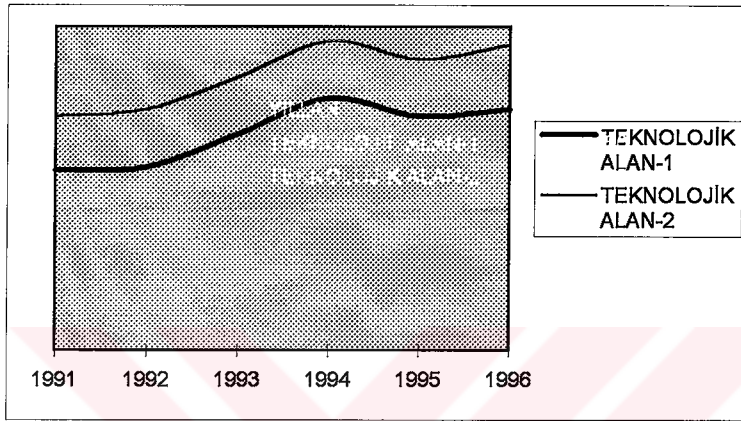
1. Adım: İnsan operatör, sistemden iş yükü eğilimini ve kaynak eğilimini grafiksel olarak göstermesini istemektedir. Bütün safha boyunca sistem, kullanıcının belirlediği önemli kısmi çözümlerin izi ışığında ilerleyecektir. Bu sonuçları ağaç dalları şeklinde

gösterecektir, çünkü her yeni sonuç hiyerarşik bir düzende öncekinden türetilmektedir.
Şekil 4.12 çözüm ağacını göstermektedir.

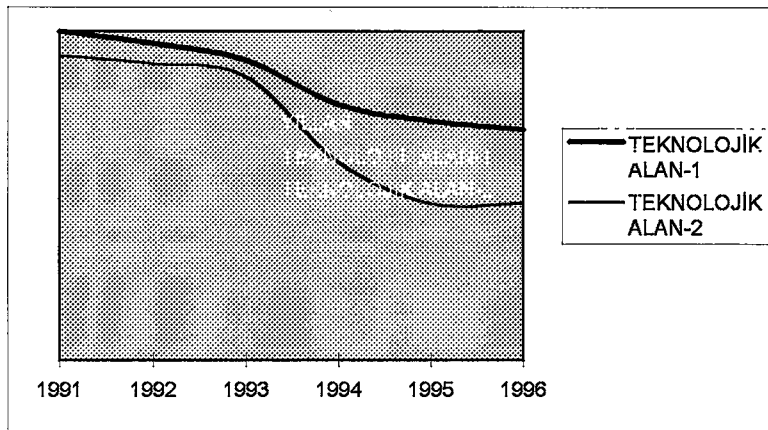


Şekil 4.12. Etkileşimli Sistem ve Kullanıcı Seçimleri

2. Adım: Sistem grafiksel olarak Pomigliano-Mont Fabrikasında var olan aşırı yükleme problemini, 1991 yılından 1996 yılına kadar, iki teknolojik alanda göstermektedir (Şekil 4.13). Sistem yine aynı yıllar için Pomigliano-Fabb Fabrikasının az yükleme problemini de göstermektedir (Şekil 4.14).



Şekil 4.13. Pomigliano-Mont Fabrikasındaki Aşırı Yükleme Probleminin Grafik Gösterimi



Şekil 4.14 . Pomigliano-Fabb Fabrikasındaki Az Yükleme Probleminin Grafik Gösterimi

3. Adım: Operatör, Pomigliano-Mont Fabrikası'ndan başlayan bir parça seviyesi problem çözümüne girmiştir, çünkü fabrikanın çok yükleme problemi stratejik önemi olan bir aktiviteyi içermektedir (montaj hattı). Operatör ayrıca Pomigliano-Fabb Fabrikası'nda oluşan problemleri de göz önünde bulundurmaya tercih etmektedir. Bu demektir ki, Pomigliano-Mont Fabrikası'ndaki problemleri çözerken, iki problemi de çözebilecek hareketlere sahip, iki fabrikada da üretilen aday parçalar tercih edilecektir.

4. Adım: Sistem, Pomigliano-Mont probleminde dahil edilen faaliyetleri gösteren ve bütün bu faaliyetlerde ihtiyaç duyulan işçilik saatlerinin ve mevcut işçilik saatlerinin toplamını gösteren bir tablo oluşturmaktadır (Şekil 4.15).

Fabrika: Pomigliano-Mont

Teknolojik alan: Montaj

	1991	1992	1993	1994	1995	1996
Boyama	145.486	151.371	169.878	190.416	160.589	169.858
	99.978	104.006	113.662	137.634	116.166	129.234
Komple Montaj	891.990	949.405	1.038.129	1 1.257.406	1.207.769	1.358.819
	646.524	676.039	737.132	1.063.386	1.030.209	1.196.953

Şekil 4.15. Pomigliano-Mont Fabrikası'ndaki Aşırı Yükleme Probleminin Tablo Gösterimi

5. Adım: Operatör, başlamak için bir yıl ve faaliyet seçmektedir.

6. Adım: Sistem, öncelik kurallarına göre seçilen aday parçalar üzerinde "satın almak" hareketlerini simüle etmeye başlamaktadır. Seçilen "satın almak" hareketlerini onaylamak veya reddetmek için gereken ara adımlardan sonra, sistem gerekli işçilik saatleri ve mevcut işçilik saatleri arasındaki farkın kabul edilebilir maksimum değerin altında olduğunu belirten kısmi bir çözüme ulaşmaktadır (Şekil 4.16). 7. adımda görüleceği gibi, iki fabrikada da yıllık toplamlar için çözüm bulunmasına rağmen, aylık seviyeye bakıldığında problemlerin devam ettiği görülecektir.

Fabrika: Pomigliano-Mont

Teknolojik alan: Montaj

	1991	1992	1993	1994	1995	1996
Boyama	99.438	103.582	113.170	138.169	116.563	129.953
	99.978	104.006	113.662	137.634	116.166	129.234
Komple Montaj	646.524	676.339	737.500	1.064.513	1.030.903	1.196.870
	646.524	676.039	737.132	1.063.386	1.030.209	1.196.953

Şekil 4.16. İş Yüğü ve İş Gücünün Dengelenmesi

7. Adım: Kullanıcı, ulaşılan kısmi çözümü, çözüm ağacında depolar. Sistem bu çözümü ÇÖZÜM-1 olarak niteler. Sonra kullanıcı, iş yüküne karşı kaynak eğilimlerini doğrulamak istemektedir. Parça seviyesi safhası Pomigliano-Fabb Fabrikası'nda bazı az/aşırı yükleme problemleri bırakmıştır. Bu problemler lot seviyesi çözümüne girerek daha iyi yerleştirilebilir.

8. Adım: Sistem, Pomigliano-Fabb Fabrikası'nın faaliyetlerini, sütunları plan döneminin aylarını gösteren bir tablo halinde göstermiştir (Şekil 4.17). Bu tablonun hücrelerindeki

değerler, mevcut işçilik saatleri ile herhangi bir ayda, herhangi bir faaliyet için dahil edilen üretim lotu için gereken işçilik saatlerinin toplamı arasındaki farkı göstermektedir. Farklar kabul edilebilir maksimum değeri aştığında, az/aşırı yükleme probleminin olduğu hücre, sistem tarafından işaretlenir. İlk faaliyet grubunda bir aşırı yükleme problemi hala mevcuttur. Lot seviyesi problem çözümüne başlamadan önce bir menü, iki değişik etkileşim seviyesinden birini seçmek için yardımcı olur.

- Düşük etkileşim: Problem çözümü, kullanıcıya her bir az/aşırı yükleme problemi için, çözümün kabulü veya reddi için soru soracaktır.
- Yüksek etkileşim: Sistem, kullanıcıya her bir makul "üretmek/satın almak" hareketinin kabulü veya reddi için soru soracaktır.

	OCA 1991	ŞUB 1991	MAR 1991	NİS 1991	MAY 1991	HAZ 1991	TEM 1991	AĞU 1991	EYL 1991	EKİ 1991	KAS 1991	ARA 1991
07-001	3658	2543	117	11	125	12	369	95	46	291	201	134
14-001	2529	2539	847	540	14	487	429	82	752	659	569	593
41-002	264	995	54	240	190	129	27	8	28	117	7	68
37-001	31	529	569	341	351	71	486	107	434	595	585	567
32-008	0	0	0	0	0	0	0	0	0	0	0	0
TOPLAM	6482	6606	1587	1132	680	699	1311	292	1260	1662	1362	1362

Şekil 4.17. İş Gücü ve İş Yüğü Arasındaki Aylık Farklar

9. Adım: Kullanıcı, başlamak için en önemli problemi seçer. Sistem, problemi çözmeye "satın almak" hareketleri uygulayarak başlar ve çözüm vaadeden hareket setleri geliştirir (Şekil 4.18). Bu safha boyunca kullanıcı, sisteme müdahale etme hakkına sahiptir, örneğin "satın almak" hareketlerini veya tüm çözüm şeklini reddedebilir. Kullanıcı, aday

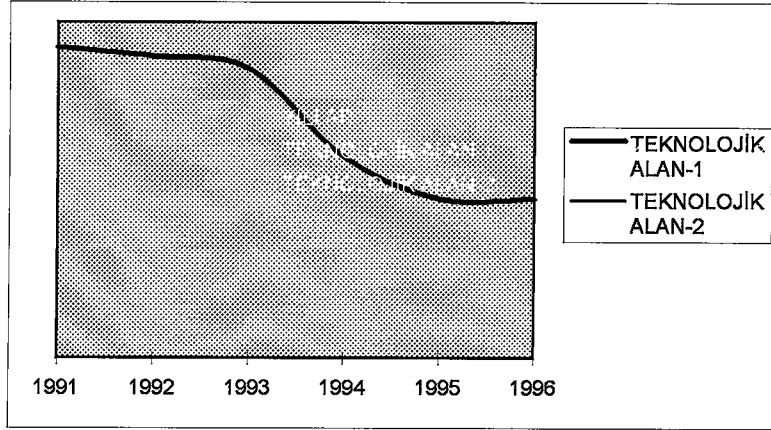
lotlar üzerindeki kısıtlamaları kaldırabilir veya bunlara yenilerini ekleyebilir. Kullanıcının bu adımda yapabileceği işlemler şunlardır:

- Aday bir üretim lotunun hareketini engellemek
- Bütün üretim parçasının hareketini engellemek
- Verilen teknolojik alanda belirli bir günden sonra yapılan hareketleri engellemek
- Verilen bir üretim lotu için iç/dış yüzdeleri kısıtlamak
- Verilen bir parça için iç/dış yüzdeleri kısıtlamak

--MEVCUT PROBLEM ÇÖZÜLDÜ (AYLIK SAPMA %5'TEN KÜÇÜK)		
Toplam Sapma: 185.857 saat		
Kalan sapma: 793 saat		
PARÇA NO	HAREKET TÜRÜ	HAREKET MİKTARI
07-001	SATIN ALMAK	54
14-001	SATIN ALMAK	15
41-002	SATIN ALMAK	2344
37-001	SATIN ALMAK	5797

Şekil 4.18. Sistemin Oluşturduğu "Satın Almak" Kombinasyonları

10. Adım: Operatörün ulaştığı sonuç, sistem tarafından ÇÖZÜM 1-1 olarak tanımlanmakta ve çözüm ağacına depolanmaktadır. Şekil 4.19 pratikte bütün problemlerin aşıldığını grafiksel olarak göstermektedir.



Şekil 4.19. İş Gücü ile İş Yükünün Dengelenmesi

11. Adım: Operatör, 41-002 nolu üretim lotunun lojistik nedenlerle hareket ettirilemeyeceğini belirlemektedir. Bu noktada operatör, kaydedilmiş ÇÖZÜM 1'e geri dönerek bu lota bir kısıtlama ekleyebilir. Bu kısıtlama bundan sonraki hareketleri engelleyecek bir karakterdedir. Burada elde edilen kısmi çözüm de sistem tarafından kaydedilmekte ve ÇÖZÜM 1-2 olarak adlandırılmaktadır. Operatör lot seviyesi probleminde bu ek kısıtlama ile yeniden başlar. Lot seviyesi problem çözümü bu lotu daha fazla hareket ettiremez ve böylece alternatif bir çözüm bulunabilir. Sistem bu çözümü de kaydetmekte ve bu çözüme ÇÖZÜM 1-2-1 adını vermektedir.

12. Adım: Bu safha sonucunda oluşturulan sonuçlar ağacı (Şekil 4.12) kullanıcı tarafından dikkatle incelendikten sonra, şirket için en iyi global plana karar verilecektir.

4.4.3.5. SİSTEMİN PERFORMANSI

a. AERPLAN'DAN ÖNCE "ÜRETMEK" VEYA "SATIN ALMAK" KARARI

Eski hesaplama imkanlarıyla, tek bir az/aşırı yükleme problemini çözmek üzere kullanılacak tam bir "üretmek" veya "satın almak" hipotezi oluşturmak bir saat kadar

sürmekteydi. Bu demektir ki, 36-48 saatlik bir çalışma süresince sadece az miktarda "üretmek/satın almak" hipotezi oluşturulabiliyordu. İnsan planlayıcılar, genellikle doğru hareketleri seçerken çok zekice davranıyorlardı. Ancak onlar için izlenimlerini analitik olarak haklı çıkarmak çok zordu ve farklı problemler arasındaki engelleri hesaba katmak karmaşık bir işlemdi.

b. AERPLAN KULLANARAK "ÜRETMEK" VEYA "SATIN ALMAK" KARARI

Sistem, 386 (Common Lisp) çevresinde çalışmaktadır ve tek bir problem için tam bir hipotez oluşturması sadece bir kaç dakika sürmektedir. Bu, operatörün AERPLAN'ı kullanarak, daha fazla hipotez formüle etmesi ve daha iyi global bir çözüme ulaşması anlamına gelmektedir.

Ortalama karmaşıklığa sahip bir aşırı yükleme problemi, normal olarak lot seviyesi safhasında çözülmektedir.

Problem Örneği:

Başlama tarihi: 4/1/1990

Bitiş tarihi: 8/1/1990

Problem tipi: Aşırı yükleme

İncelenen teknolojik alan: Montaj hatları

İncelenen fabrika: 1

İşçilik saati miktarı: 35000

İş yükü:

Fabrika sayısı: 7

Üretim lotları: 1500

Plan süresi: 6 yıl

Araştırma uzayı:

Genişletilmiş durumlar: 1200

Çözüm şekli: 10 "satın almak" kararı

Çözüm süresi: 10 dakika



5. BÖLÜM

SONUÇ

Otomasyonun yazılım yönü olarak görebileceğimiz uzman sistemler, bilgi ve tecrübe içeren bilgisayar uygulamalarıdır. Uzman sistemler bir insan uzman gibi kararlar verebilmekte, verdiği kararları açıklayabilmektedir. Diğer bilgisayar uygulamalarının aksine, nümerik hesaplamalardan çok sembolik işlemler gerektirirler.

Uzman sistemleri tasarlamada karşılaşılan sorunlardan bir tanesi, uzmandan uzmanlık bilgisinin alınıp, bu bilginin bilgisayarın anlayacağı şekilde temsil edilmesidir. Bilgi temsili, bilginin kısımlarını ve insan beynini temsil için kullanılan değişik kalıpları ile büyüleyici bir konudur. Uzman sistemler, çok basit bilgi temsil şemaları kullanılarak oluşturulabilirler. Bilgi temsil şemaları, esin kaynaklarını insan beyninin karmaşıklığı ve organizasyonundan alabilirler veya ince, biçimsel şemalar üreterek performanslarını arttırabilirler. Bilgi temsili üzerinde yapılan araştırmaların amacı, uzman sistemin performansının, üzerinde işlem yaptıkları alan hakkında daha iyi anlama kabiliyeti vererek geliştirilmesidir

Planlama prosesinin karmaşıklığı ile mücadelede, alternatif çözümler yaratma ve geliştirmede uzmanların tecrübesi ve hüküm verebilmesi anahtar rol oynamaktadır. Uzman sistem teknolojisi, insan - bilgisayar etkileşiminin tüm avantajlarından yararlanan bir çalışmayı sağlamaktadır. Kurulacak bir uzman sistem, ilk çözümlerin yaratılması, faaliyetlerin fizibilitelerinin doğrulanması, kararların sonuçlarının hesaplanması gibi temel modeller üzerine kurulmalıdır. Bir uzman sistem, her planlama safhasında kullanıcının ihtiyaçlarına en iyi şekilde uyan bilgi formatının seçilmesini sağlayacak ve değişik problem çözme taktiklerini yerleştirecek şekilde dizayn edilmelidir.

Üretim planlama alanında karar problemleriyle mücadele eden uzman sistemler açıklayıcıdır ve iki görevi yerine getirirler.

1. Çözümlerinin geçerliliğinin sağlanması
2. Verilen faaliyetin sonuçlarının hesaplanması

Bu çözümlerin değerlendirilmesi ve geliştirilmesi karar vericiye aittir.

Bütün bunlar dikkate alınarak, hazırlanacak olan uzman sistem ile aşağıdaki problemlere cevap aranacaktır:

1. Üretim sistemindeki mevcut verilere dayanarak, sistemin gücünü ortaya koymak. Bu sayede, üretimi düşünülen miktarların ne ölçüde gerçekleştirilebileceğini bulmak. Diğer yandan işletmenin verimli çalışıyor olması için gerçekte üretilebilecek rakamların ne olması gerektiğini tesbit ederek gerekli kişileri bilgilendirmek.
2. Planlamacılara işlerinde yardımcı olmak. Malzeme, işgücü, stok gibi alt dallarla uğraşanlara başlangıç verilerini sunmak.
3. Planlama sonuçlarını sonraki dönemler için saklamak. Çeşitli verilerde meydana gelebilecek değişiklik halinde, ki bu değişimler planlamacının isteği doğrultusunda bilerek yapılan değişiklikler olabileceği gibi, beklenmeyen değişimler de olabilir, planların nasıl değişeceğini kullanıcılara sunmak.

Tezde, üretim planlama faaliyetlerinde kullanılan bir uzman sistem olan ReDS'nin genel işlem çatısı anlatıldı. Çatının en önemli katkısı, planlamadan sonlu çizelgelemeye kadar bütün çizelgeleme faaliyetlerini kapsamasıdır. Ayrıca, ReDS bir teslimat kontrol stratejisi içerir. Planlama safhasında, bazı küçük çatışmalar ortaya çıkabilir ama çizelgeleme safhasında bütün çatışmalar çözülmektedir. ReDS, Just-In-Time sistemi ile beraber, ortak bir mimari çizelgeleme için çeşitli perspektifler yaratılmasını sağlar. İki uygulama da birçok fabrikada kullanılmış ve başarılı olmuştur.

Çizelgelemeye yardımcı olmak amacıyla geliştirilen diğer bir uzman sistem olan AERPLAN'da ise, etkileşimli bir problem çözüm mimarisi oluşturulmuştur. Sistem, kısmi çözüm basamaklarından geçerek sonuçlara varırken iki farklı avantaja sahiptir:

1. Arařtırma sırasında, herhangi bir anda yararlı sonuçlar çıkarabilmektedir.
2. Problem çözmeyi alana özel karakteristikleri kullanarak kolaylařtırmaktadır.

Uzman sistemlerin, insanlara ait bazı görevleri üstleneceęi bir gerçektir. Amaç, insanın yerini tam olarak almak olmayıp, karar vericinin daha iyi ve daha çabuk sonuca ulaşma kabiliyetini artırmak için bilgisayar desteęi sağlamaktır.



REFERANSLAR

- Baker, K.R., 1974. "Introduction to Sequencing and Scheduling", John Wiley and Sons Inc.
- Basden, Andrew, 1994. "Three Levels of Benefits In Expert Systems", Expert Systems, Vol.11, No.2,
- Beerel, Annabel C., 1987. "Expert System Stratejic Implications & Applications", Ellis Horwood Ltd.
- Bertolotti, Emilio, 1990. "Interactive Problem Solving For Production Scheduling", BULL HN R & D Division
- Bilginoğlu, Fahir, Prof. Dr., 1993. İ.Ü. İşletme Fakültesi Yönetim Dergisi, Yıl:4, Haziran
- Bonnet, A., Haton, J.P., & Ngoc, J. M., 1986. "Expert Systems Principles and Practise", Prentice Hall
- Chan., Christine W., 1995. "Knowledge Engineering For A Process Design Domain", International Journal of Expert Systems, Vol. 8 No:1, by JAI Press Inc.
- D'agapeyeff, A., 1989. "Expert Systems: A Vision Of The Future", Vol.85, April
- Efstathiou, Janet, 1989. "Expert Systems In Process Control", Longman In Assocation With The Institue of Measurement And Control, U.K.
- Frost, J., and Orciuch, E., 1984. "ISA: Intelligent Scheduling Assistant", IEEE, The First Conference On Artificial Intelligence Application, New York

- Feigenbaum, E. A., and McCorduck, P., 1984. "The Fifth Generation", Michael Joseph Inc.
- Graves, S., 1981. "A Review of Production Scheduling", Operations Research, Vol.29, No.4
- Grogono, P., Batarek, A., Preece, A., and Suen, C., 1991. "Expert System Evolution Techniques", Expert Systems, November, Vol.8, No.4
- Hadavi, Khosrow., Hsu, Wen-Ling., Chen, Tony., 1990. "An Architecture for Real Time Distributed Scheduling", Siemens Corporate Research
- Kanet, John, Adelsberger, Heimo, 1987. "Expert Systems in Production Scheduling", European Journal of Operational Research, Vol.29
- Madni, Azad M., 1988. "The Role Of Human Factors In Expert System Design And Acceptance", Human Factors, 30 (4), California, USA
- Moreira, N.A., and Oliveira, R.C., 1991. "DSS for Production Planning In An Industrial Unit", European Journal of Operational Research, North Holland
- Ntuen., A. Celestine., 1991. "Step-by-step Procedure In Developing An Expert System", Industrial Engineering, October
- Pollitzer, E., and Jenkins, J., 1985. "Expert Knowledge, Expert Systems and Commercial Interests", International Journal of Management Science, Vol. 13, No.5
- Simons, G.L., 1985. "Expert Systems & Micros", NCC Publications
- Turban, E., 1988. "Review of Expert System Technology", IEEE Transactions on Engineering Management, Vol.35, No.2

- Üntüvar, A., Allahverdi, M., Yıldız, S., 1995. "Endüstride Uzman Sistem Uygulamaları", Makina-Tek Dergisi
- Waterman, D.A., Hayes-Roth, F., 1983. "Building Expert Systems", Addison-Wesley, Cambridge



ÖZGEÇMİŞ

Doğum Tarihi	1 Nisan 1972
Doğum Yeri	ISPARTA
Eğitim	1978-1983 Isparta Ülkü İlkokulu 1983-1990 Isparta Anadolu Lisesi 1990-1994 Y.T.Ü. Endüstri Mühendisliği (Lisans) 1994- Y.T.Ü. Endüstri Mühendisliği (Y. Lisans)
İş	1996- Eczacıbaşı-Baxter H.Ü. San. Ü.P.K.Departmanı

