

127021

**UÇUŞ MÜRETTEBATI İÇİN ARIZA TEŞHİS UZMAN
SİSTEMİ**

YÜKSEK LİSANS TEZİ
Uzay. Müh. Sedef ÖZÇELİK
(514981014)

Tezin Enstitüye Verildiği Tarih : 2 Ocak 2002
Tezin Savunulduğu Tarih : 16 Ocak 2002

Y.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Tez Danışmanı : Doç.Dr. Alpaslan FIĞLALI *Alpaslan 8 Şubat 2002*
Diğer Jüri Üyeleri Doç.Dr. Ziya ULUKAN (G.Ü) *Ziye*
Dr. Y. İlker TOPÇU *Y. İlker*

OCAK 2002

127021

ÖNSÖZ

Günümüzde hızla gelişmekte olan alanlardan birisi de havacılıktır. Bu alandaki çalışmalar da diğerlerinde olduğu gibi bilgi-işlem teknolojileri ile desteklenmektedir. Bu tez çalışmasında öncelikle, son yıllarda giderek daha da başvurulmuş bilgi teknolojilerinden uzman sistemler (US) incelenmiştir. US'ler bilgiyi bir insanın yaptığı gibi işlemeye çalışan yapay zeka (YZ) programları olarak, zamanın bilgiye olan açlığına karşı kullanılacak değerli araçlardır.

Arıza giderme US'leri ise kendilerine duyulan ihtiyacın hiçbir zaman sona ermeyeceği US'ler olarak bütün endüstri alanlarında yaygın olarak kullanmaya başlamışlardır.

Çalışmalarında bana yardımcı olan ve destek olan, bilimsel araştırmalarda yol gösteren değerli hocam ve danışmanım Doç. Dr. Alpaslan FIĞLALI'ya sonsuz teşekkürler sunarım.

Yetişmemin her kademesinde bana büyük emeği geçen sevgili anneme, babama ve değerli hocalarıma teşekkürlerimi sunarım.

Ocak 2002

SEDEF ÖZÇELİK

İÇİNDEKİLER

KISALTMALAR	V
TABLO LİSTESİ	VI
ŞEKİL LİSTESİ	VII
ÖZET	VIII
SUMMARY	IX
1. GİRİŞ	1
2. UZMAN SİSTEMLER	3
2.1. Giriş	3
2.2. US Tanımı	4
2.3. US'ler	4
2.4. US'lerin Yapısı	6
2.5. US'in Öğeleri	6
2.6. Çıkarım Yöntemleri	7
2.7. Temel Bileşenleri	7
2.8. US'in Faydaları	11
2.9. US'in Sınırları	12
2.10. US'in Uygulama Alanları	13
3. BİLGİ GÖSTERİM YÖNTEMLERİ	16
3.1. Kural Tabanlı Gösterim	16
3.2. Çerçeve Tabanlı Gösterim	17
3.2.1. Çerçeve Tabanlı Bilgi Gösterim Dilleri	20
3.3. Mantık Tabanlı Gösterim	20
3.4. Anlamsal Ağlar	21
3.5. Sınıflandırıcılarda Bilgi Gösterimi	22
3.6. Genetik Algoritmalar	22
3.7. Sinirsel Ağlarda Bilgi Gösterimi	22
4. PROLOG UZMAN SİSTEM ARACI	25
4.1. Giriş	25
4.2. ALS Prolog Özellikleri	27
4.3. Prolog Olguları	27
4.3.1. Basit Olgular	28
4.3.2. Değillemeler	30
4.3.3. Yapılar	30
4.3.4. Bileşik Olgular	30

4.4. Prolog Kuralları	32
4.4.1. Basit Kurallar	32
4.4.2. Bileşik Kurallar	33
4.5. Tekrarlılık	35
4.6. Listeleme Yöntemi	35
4.7. Prolog Fonksiyonları	37
5. UYGULAMA	43
5.1. Problemin Tanımı	43
5.2. Problemin Alanı	44
5.3. Boeing 737-400/500 İçin US İhtiyacı	44
5.4. Boeing 734-400/500 İçin Arıza Giderme İşlemi	45
5.5. Uçuş Mürettebatının Arıza Teşhis Kontrol Listesinin Yapısı	45
5.6. Anormal Durum Kontrol Listesi Çalışması	49
5.7. Yeni Nesil Uçuş Yönetim Sistemleri	49
5.8. Uçuş Ekibi Faaliyetleri İçin Yeni Boeing Yazılımı	52
5.9. Modelin Özellikleri	54
5.10. Modelin Yapısı	54
5.11. Programın Çalışması ve Örnek Sorgular	57
6. SONUÇLAR VE ÖNERİLER	63
KAYNAKLAR	65
EKLER	68
EK A Örnek Arıza Giderme Kullanım Kılavuzu	68
EK B Anormal Durumlar Kontrol Listesi	71
EK C Bilgisayar Programı	75
ÖZGEÇMİŞ	76

KISALTMALAR

ES	: Expert Systems
YZ	: Yapay Zeka
US	: Uzman Sistem
GA	: Genetik Algoritmalar
SA	: Sinir Ağları
BC	: Bus Controller
CDU	: Control Display Unit
MFD	: Multi-Function Display
FMS	: Flight Management System

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. : Jeoloji, Kimya, Elektronik ve Eğitim Alanında Bazı US'ler	14
Tablo 2.2. : Tıp Alanında Bazı US'ler	14
Tablo 2.3. : Tarım Alanında Bazı US'ler	15
Tablo 2.4. : Havacılık ve Uzay Alanında Bazı US'ler	15
Tablo 3.1. : Bazı Çerçeve Tabanlı US paketleri ve Geliştiricileri	20
Tablo 3.2. : Veri Seti	23
Tablo 5.1. : Boeing Anormal Durum Kontrol Listesi Ana Başlıkları	46
Tablo 5.2. : Anti-İce, Rain Anormal Durumları Tablosu	48
Tablo B. : Anormal Durumlar Kontrol Listesi	71

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1. : US'nin Temel Bileşenleri	10
Şekil 3.1. : Çerçeve Tabanlı Gösterim	19
Şekil 3.2. : Veri Seti Sinir Ağı	24
Şekil 5.1. : Uçuş Ekibi İçin Arıza Teşhis Uzman Sistemi	56
Şekil A. : Örnek Arıza Giderme Kullanım Kılavuzu	68



UÇUŞ MÜRETTEBATI İÇİN ARIZA TEŞHİS UZMAN SİSTEMİ

ÖZET

Bu tezde güdülen birinci amaç tamamen bilgisayara dayalı uçuş mürettebatı için bilgi bir arıza teşhis ve giderme US'i geliştirmektir.

US'ler bilgiyi bir insanın yaptığı gibi işlemeye çalışan YZ programları olarak, zamanın bilgiye olan açlığına karşı kullanılacak değerli araçlardır. Arıza giderme ve teşhis US'leri ise kendilerine duyulan ihtiyacın hiçbir zaman sona ermeyeceği US'ler olarak bütün endüstri alanlarında yaygın olarak kullanılmaya başlanmıştır.

Tezde US kavramı tanımlanmış, yapısı, sınırları, yararları, uygulama alanları ve bilgi gösterim metotları konularına yer verilmiştir. Tezin amacı, problemin tanım, problemin alanı, anormal durumlarda uçuş mürettebatının görev yönetimi, uçuş mürettebatının arıza teşhis el kitabının yapısı, bilgi tabanlı pilot kabini yardımcı sistemi incelenmiştir.

Tezin ikinci aşamasında geliştirilen arıza giderme US programı kullanılarak gerçek hayattan bir uygulama ortaya çıkarmak ve böylelikle bu geliştirilen programın bir çok ticari uçakta uçuş mürettebatı için kullanılabileceğini kanıtlamaktadır. Böylece pilot iş yükünün azaltılması ve operasyon etkinliğinin artırılması hedeflenmiştir. Günümüzde yeni nesil uçak tasarımı ve mevcut uçak modifikasyonlarında bu tür çözümlere gidilmektedir.

US kurma araçlarından yazılım dili olarak, önerilen sistemin bilgi tabanlı olması gerektiğinden bir kural dili olan Prolog seçilmiştir. Bu amaca ALS Prolog US aracı kullanılarak hazırlanan bir programla ulaşılmıştır.

FAULT DIAGNOSIS EXPERT SYSTEM FOR FLIGHT CREW

SUMMARY

Expert systems (ES) are computerised program tools that mimic the judgement of human experts in a well defined domain. ES are one area of artificial intelligence that is being developed and used many diagnostic tasks.

The fault diagnosis of machine is a complicated engineering problem. It is highly related to the manufacture surrounding. The diagnosis ability of an ES is decided by the actual experiences and knowledges of its designer. For the diversity of machine and the difference of the manufacture surrounding, it is impossible for the designers to know all the facts of the machine that ES will be used to. So we can't expect that an ES has the diagnosis ability of a real expert at birth.

Expert systems typically have a number of distinguishing components. The knowledge base is the component that contains the knowledge obtained from the domain expert. The most common way of representing knowledge is using rules. The inference engine is the component that manipulates the knowledge found in the knowledge base as needed to arrive at a result or solution. The user interface is the component that allows the user to query the system and receive the results of those queries. Many ES also have an explanation facility which explains why a question was asked or how a result or solution was obtained.

There are three different mains of building an ES knowledge base, namely rules, frames and semantic networks.

Frames method is used mostly when the variables in the knowledge base have hierarchical structure. Since frames allow the structured representation of descriptive knowledge, they also facilitate analogical reasoning. Frame systems provide no direct methods for describing how the knowledge stored in frames is to be used.

Semantic networks method is preferred when the variables of the knowledge base have a lot of different interrelations such as "is, can, has, etc.". Semantic networks (also called associative networks), are another major form of knowledge representation. In such systems, knowledge is represented in the form of nodes and arcs, where nodes represent objects, classes, and events while arcs represent relationships that hold between the concept nodes.

Production rules is preferred when the rules of the knowledge base can be expressed in the form of "IF,..., THEN". Most of the developed ES uses production rules.

In logic based systems, knowledge is represented as simple and complex predicate

statements. Logic based systems rely on resolution theorem proving for their inference mechanisms. In these systems prescriptive knowledge is represented in the form of condition-action rules or procedures.

Classifiers and classifier systems based on genetic algorithms are inductive learning systems. In these systems knowledge is represented as vectors or matrices of values to a set of parameters or a set of message lists.

Genetic algorithms represent knowledge as vectors of values to a set of parameters or a set of message lists. GA based classifiers These systems support, besides inductive learning, conflict resolution, classification and learning control rules.

Neural nets are composed of a large number of interconnected processing units. In these systems knowledge is represented by the totality of the connections. Neural networks support inductive learning, classification and conflict resolution.

Prolog is a unique programming language. While other languages like BASIC, FORTRAN and PASCAL are classified as procedural languages, Prolog is a declarative language.

In other languages the programmer must write the procedures of how the problem is to be solved in order to get a program working, whereas in Prolog computing procedures are built-in in the language. All the programmer has to do is give the necessary information about the problem domain and let Prolog find the answers to the questions asked by the programmer.

Prolog is based on First Order Predicate Logic and the name Prolog derives from PROgramming in LOGic.

First Order Predicate Logic is a well developed part of Symbolic Logic. A Prolog programmer is not a programmer but a logician, or in fact any one who can represent information in a logical way.

Advanced in electronics and computer technology have a profound effect on modern aircraft. On flight deck electronic displays and computer controlled avionic devices are common features. Powerful computer technology is involved in the signal processing to generate display formats and to transmit signals to and from control devices and the remaining avionic components.

However, the crew station as we know it today is under-going an even more profound change, which will allow to make use of abstract human like knowledge and reasoning.

Non-normal situations require the flight crew to devote more time and attention to the function related to the cause of the non-normal condition (for example, systems management if an engine has failed, or flight management if a storm is encountered). However, it is rare that this function is the only one affected. Tasm management has a large role to play in how the crew's attention is dispersed among the different functions.

Checklists are used by the flight crew to properly configure an aircraft for safe flight and to ensure a high level of safety throughout the duration of the flight. In addition, the checklist provides a sequential framework to meet cockpit operational requirements, and it fosters cross-checking of the flight deck configuration among crew members.

This thesis examined expert system instead of checklists for non-normal procedures for a typical transport aircraft. For the purposes of this thesis, a typical transport aircraft (Boeing 737-400/500) checklists is used for application which is programmed in Prolog.



1. GİRİŞ

Günümüzdeki hızlı teknolojik gelişmeler, gündelik yaşantımızı olduğu kadar işletmeleri de büyük ölçüde etkilemektedir. İşletmeler bu gelişmenin ortaya çıkardığı ürünleri, üretim, planlama, kontrol vb. alanlarda kullanmaktadırlar. Çoğu zaman işletmeler, teknolojik gelişmeler karşısındaki hıza ayak uyduramamakta ve ileri teknolojileri kullanan rakipleri karşısında zorlanmaktadırlar. Özellikle son yıllarda bilgisayar bilimlerinde, yazılım ve donanım alanında çok hızlı gelişmeler yaşanmaktadır. Bilgisayar kullanımının hızla yaygınlaşması ve yeni ortaya çıkan yazılımların, daha üst seviyelerde donanıma ihtiyaç göstermesi ile yeni ürünlere olan talep de artmaktadır.

Bilgisayar bilimlerindeki bu ilerleme, insan gibi düşünen ve davranan sistemlerin geliştirilmesine yönelik olarak, 1950'li yıllardan beri sürmektedir. Yapay Zeka (YZ) olarak isimlendirilen bu alan, insan düşünme ve davranışlarını taklide yönelik olduğundan, nöroloji, psikoloji ve mühendislik gibi farklı disiplinleri kapsayan geniş bir alana yayılmıştır.

İnsan gibi düşünebilen ve davranabilen sistemlerin geliştirilmesi için yapılan çalışmalarda bugün için gelinen nokta, henüz YZ'nin tam olarak geliştirilememiş olmasıdır. YZ'nin yapılabirliği üzerinde yapılan felsefi tartışmalar bir yana, düşüncenin salt fiziksel süreçlere indirgenebildiği kabul edilse bile, henüz beynin tüm fonksiyonları tam olarak çözülemediğinden, bugün için yapılabilmesi henüz mümkün gözükmemektedir. Fakat konu üzerinde yapılan çalışmalar farklı alanlarda hızla devam etmektedir. Burada şunu da belirtmek gerekir ki, YZ'nin yapılamayacağını savunanlar, konu üzerinde karşıt görüşlü araştırmacılar ile aynı araştırma ve geliştirme çalışmalarını yürütmektedirler. Çünkü her iki tür

arařtırmacının yapmaya alıřtıkları řey gzlemlenebilen nesnel olayların benzerini yapabilmektedir.

YZ disiplini altında onu destekleyen farklı alanlar bulunmaktadır. Teorik olarak YZ yapılsa, onun fayda saėlayabilmek iin gerek dnya ile iletiřim iinde olması gerekir, aynı insanın beř duyu organına sahip olduėu gibi. İřte US'ler, bulanık mantık ve sinirsel aėlar zerinde yapılan alıřmalar, YZ disiplinini bu alanlarda desteklemektedirler.

US'ler bilgiyi bir insanın yaptıėı gibi iřlemeye alıřan YZ programları olarak, zamanın bilgiye olan alıėına karřı kullanılacak deėerli aralardır. Arıza giderme ve teřhis US'leri ise kendilerine duyulan ihtiyaın hibir zaman sona ermeyeceėi US'ler olarak btn endstri alanlarında yaygın olarak kullanılmaya bařlanmıřtır.

Bu farklı alanlarda yapılan alıřmaların ortaya ıkardıėı teknolojik rnler, iřletmelerde de sıklıkla kullanılmaktadır. nk ortaya ıkan rnler, insan zelliklerinin sınırlı da olsa belli bir kısmına sahip olabildiėinden, belirli iřlerde insanların yerine onlardan daha verimli olarak kullanılmaktadır. Bu ise ynetim aısından bakıldıėında, iřletme de verimlilik artıřı ve hata oranları ve birtakım diėer masraflarda azalmalara sebep olduėundan olduka nemlidir. Ynetim aısından aynı oranda dikkat gsterilmesi gereken nokta, yeni teknolojilerin kullanılabilmesine ynelik, alıřanların eėitilmeleri ve alıřanların geliřen teknolojiye uyumunun saėlanması konusudur.

2. UZMAN SİSTEMLER

2.1. Giriş

Yapay Zeka (YZ), çok geniş bir kavram olup verilen bir problemi insanın yaptığı gibi yapabilen bilgisayar sistemi veya bilgisayar programı olarak tanımlanabilir. YZ, bilgisayar bilimlerinin bir alt dalıdır. İnceleme sahası ise insan zekasını ve davranışlarını taklit edebilen bilgisayar sistemlerinin ve programlarının tasarlanması ile ilgilidir. YZ'nın çağdaş bir bilim dalı olarak gelişmesi, 1956 yılında "YZ" teriminin ilk kullanılmasından sonra yaşanan gelişmeler, C. Shannon, M. Minsky ve J. MacCarthy'nin çabaları ve katkıları ile başlamıştır (Topçuoğlu, 2001).

Günümüzde büyük bir hızla gelişmekte gösteren YZ, robotik, sinir ağları (SA), uzman sistemler (US) vs. gibi çeşitli alt dallara ayrılır. Yatırımcılık, bankacılık, muhasebe, maliye ve diğer benzer programlar için YZ ve US metotları geliştirilmiş ve kullanılmıştır. Bu tür işlerle uğraşan bir çok şirket YZ ve US metotlarını geniş olarak kullanmaktadır. Dünyaca ünlü IBM, General Electric, General Motors, Boeing gibi şirketler vardır.

YZ ürünleri dünyada araştırma safhasını aşmış ve gelişmiş ülkelerde ticari bir araç olarak kullanılmaya başlanmıştır. Örneğin tüm dünyada 1995 yılında üretilen YZ ürünlerinin değeri 1,1 milyar dolardır. Bunun 700 milyon doları ABD'nin payına düşmektedir. YZ'nın en eski dallarından biri olan US ürünlerinin aynı yılda ABD'de 275 milyon dolar satışı gerçekleştirilmiştir. YZ ürünleri satışı analiz ederken, bunların devamlı artış gösterdiği gözlemlenmiştir. Bu artışta bilgiye dayalı sistemler, yani US'ler genel satış hacminin %70'ni kapsamaktadır (Hall, 1996).

2.2. US Tanımı

Bir US, bir uzmanlık yorumunu gerektiren karmaşık gerçek dünya problemlerini ele alır. Problemleri, bir uzmanın benzer bir problemle karşı karşıya kaldığında ulaştığı sonuçlara ulaşarak ve uzmanın akıl yürütme mantığının bir bilgisayar esaslı modelini kurarak çözerler (Vidin, 1999).

US'ler verilen bir uygulama sahasındaki karmaşık problemleri çözmek için bir uzmanın düşünme işlemlerine benzer tarzda hareket eden yazılım ve donanımdan ibaret bilgisayar sistemleridir.

İngiliz Bilgisayar Topluluğu'nun US grubu tarafından benimsenen tanım aşağıdaki gibidir.

Bir US'in işlevsel bir fonksiyon hakkında zeki bir karar alabileceği veya zeki bir tavsiyede bulunabileceği biçiminde, bilgi tabanlı program sınırları içinde bir uzman yeteneğinden somut bir örnek olarak görülebilir. Birçoklarının temel olarak kabul ettikleri özellik, sistemin soruyu soranın direkt olarak kolayca anlayabileceği tarzda kendi akıl yürütmesini doğrulama yeteneğidir. Bu özelliklere erişmek için benimsenen yol, bilgi tabanlı programlardır.

Diğer bir tanıma göre ise US, çözümleri için önemli derecede uzmanlık gerektiren zor problemleri çözmek için, bilgi, gerçekler ve sorgulama teknikleri kullanan bilgisayar tabanlı sistemlerdir. US'ler özel yazılım ve donanım konfigürasyonları için oluşturulabileceği gibi, genel amaçlı bir bilgisayarda çalışmak üzere tasarlanan yazılımlarda olabilirler (Feigenbaum, 1981).

2.3. US'ler

US kavramı, YZ terimi ile birlikte ortaya çıkmış ve YZ sistemlerinin esasını teşkil etmektedir. US'e herhangi bir karmaşık sistemde, uzman bir kişinin yaptığı işleri yapan bir bilgisayar programı gibi bakılabilir yada belirli konularda özelleşmiş ve yalnızca bu konuda bir "zeka" ya sahip programlarda US'ler olarak kabul edilebilir (Topçuoğlu, 2001).

1970'li yıllardan beri değişik alanlarda başarı ile uygulanmakta olan çeşitli US'ler mevcuttur. Bunlara örnek olarak MYCIN ve INTERNIST (Tıbbi teşhislerde),

DENDRAL ve SECS (Kimyasal analizler için), XCON (Bilgisayar sistemlerinin konfigürasyonunu belirlemek için) vs. göstermek mümkündür (Feigenbaum, 1992).

US'lerin uygulama alanlarına incelendiğinde bunlardan en önemli yeri müşahedeye dayalı sistem arızalarının tespiti gibi teşhis alanının tutmakta olduğu görülmektedir. Bu alanda en çok uygulama bulan hastalıkların teşhisidir. US'ler tıp alanından sonra en yaygın olarak elektronik ve elektrik sistemlerinde kullanılmaktadır. Bundan başka diğer teknik sistemler için arıza teşhisi yapabilen US'ler de geliştirilmiştir. Mesela, ACE US'i, telefon kablolarında meydana gelen hataların tespitinde, CATS US'i, lokomotiflerde meydana gelen hataları tespitinde, DART ve IDT US'leri, bilgisayar sistemlerinde meydana gelen hataları tespit etmek üzere tasarlanmıştır (Allahverdi ve diğ., 1995). Ayrıca başka bazı teşhis problemlerinde kullanılmak üzere içeriği boş araç ve kabuklar da geliştirilmiş ve bunlardan faydalanılmıştır. Buna örnek olarak, DEM aracı tam bilgi içermeyen veri tabanlarında sebep-sonuç ilişkilerini ortaya çıkarabilmek için hazırlanmış ve bu araç yardımı ile farmakoloji, ekoloji ve teknik teşhis gibi alanlarda çeşitli US'ler geliştirilmiştir. BEST aracı Bayer yöntemine dayanarak teşhis US'leri geliştirilmek üzere hazırlanmıştır. ESTA US'i gaz türbin motorlarında ortaya çıkabilecek kazaların önlenmesi için bunların teşhis edilmesi için tasarlanmıştır. ESPLAN kabuğu bulanık mantık yöntemini kullanmakla hem kalp-damar, hem de teknik sistemlerde durum ve arıza tespiti yapmak için uygulanmıştır (Regers et all., 1984). Kompresyonlu soğutma sistemlerinde genel şikayet ve arızaların teşhisini yapan US, arızanın sebebini bulduktan sonra yapılacak işlemleri tavsiye etmektedir. Elektrik motorlarında ve CNC tezgahlarında arıza tespiti yapan US'ler zamanı çok pahalı olan uzman kişinin yerini alabilmekte ve gereken tamir işlerini gösterebilmektedir (Ünsaçar ve diğ, 1998).

Arıza tespiti ve hastalık teşhisi yapabilen, US'lerin büyük çoğunluğu aynı zamanda eğitim amacı ile de kullanılabilir. Bu sistemler özellikle az eğitilmiş personel için çok faydalı olmakta ve çoğu zaman pahalı uzmanları kullanmaksızın çözüm getirebilmektedir (Yaldız ve diğ, 1998).

2.4. US'lerin Yapısı

Bir US genel olarak sonuç çıkarma mekanizması, bilgi tabanı ve veri tabanı, açıklama kısmı kullanıcı arabirimi ve bilgi edinme birimi gibi program modüllerinden meydana gelir. US'lerin bilgi tabanını oluşturmak için üç değişik metot mevcuttur. Bunlardan bazıları kural tabanlı sistemler, çerçeve tabanlı sistemler ve semantik ağlardır. Çerçeveler genellikle bilgi tabanındaki veriler hiyerarşik yapıda (baba-evlat) olduğunda, semantik ağlar bilgi tabanı değişkenleri olmak, sahip olmak ve muktedir olmak gibi birçok iç ilişkiye sahip olduğunda, kural tabanlı sistemler ise bilgi tabanı değişkenleri Eğer....Öyleyse yapısına getirilebildiği takdirde kullanılır. Geliştirilmiş US'lerin büyük çoğunluğunda kural tabanlı sistemler uygulanmaktadır. Bu kuralların sayısı 50'den 3 bine kadar değişmektedir.

US'lerin tasarımı karmaşık ve çok zaman alan bir iştir. Bir US'in tasarlanması genelde bir grup çalışması gerektirmektedir. US tasarımcılarına göre bir US'i tasarlayan grupta en az iki bilgi mühendisi (program için uzman kişiden gereken bilgileri alan ve programcıya ileten kişi), bir uzman ve bir programcı bulunmalıdır. Üretim kurallarına dayalı bir US prototipini tasarlamada bilgi tabanı 50'den 100'e kadar kural içerdiğinde sistemin tasarımı bir kaç aydan 1-2 yıla kadar süre almaktadır. Kuralların sayısı 1000 ile 1500 arasında olduğunda prototipin sanayi veya uygun alanda kullanılabilir duruma getirilmesi 2 yıldan 4 yıla kadar süre alabilmektedir. Tasarlanmış US'in genelleştirilerek daha geniş alanda kullanılması ve kuralların sayısının 2 bin ile 3 bin arasında olduğu durumda ise sistemin ticari ürüne dönüştürülmesi ortalama 3 ile 6 yıl zaman alabilmektedir. Bu açıklamaların ışığında kapsamlı bir US'nin tasarımının ne kadar bilgi ve zaman gerektirdiği daha iyi anlaşılmaktadır (Feigenbaum, 1992).

2.5. US'in Öğeleri

- Uzman

US'in kullandığı bilgiyi sağlayan, bir konuda uzmanlaşmış kişidir (mühendis, teknisyen, öğretmen vb.).

- Bilgi

US'in kullandığı verilerin ve yöntemlerin bütünüdür.

- Bilgi Mühendisi

Uzman ile etkileşim gerçekleştirecek, onun bu bilgilerini bilgisayarın işlem yapabileceği formatla düzenleyen kişidir.

- Kullanıcı

2.6. Çıkarım Yöntemleri

İleri doğru çıkarım: Problem çözümüne başlanırken, doğruluğu veya yanlışlığından emin olduğumuz gerçeklerin sayısı az ise tercih edilir. Bu gerçekler, kuralların koşul kısımları ile karşılaştırılır. Koşul gerçekleştiğinde, kuralın sonucu da artık bir gerçektir.

Geri doğru çıkarım: Fikir yürütmeye amaçtan başlanarak amaca kılavuzluk eden yollardan geri doğru iz sürülür. Bu yollar üzerindeki tüm koşullar sağlanıyor ise amaç da sağlanıyor demektir.

2.7. Temel Bileşenleri

Bir US iki ana parçanın birleşiminden oluşur. Geliştirme çevresi ve görüşme çevresi. Geliştirme çevresi sistemin bileşenlerini kurmak ve uzman insan bilgilerini bilgi tabanına girmek için US'i kuranlar tarafından kullanılır. Görüşme çevresi ise uzman bilgiye ulaşabilmek için uzman olmayanlar tarafından kullanılır. Şekil 2.1'de US'in temel bileşenleri görülmektedir. Bir US'de aşağıdaki bileşenler mevcuttur (Parsaye ve Chignell, 1988).

- Bilgi kazanma

Bazı bilgi kaynaklarından bir bilgisayar programına problem çözümü için bilgi aktarma ve dönüştürme işlemleri yapılır. Potansiyel bilgi kaynakları uzman insanlar, kitaplar, veri tabanları, özel araştırma raporları ve kullanıcının kendi deneyimleri olabilir.

- Bilgi tabanı

Bilgi tabanı problemlerin anlaşılması, formülasyonu ve çözümü için gerekli olan tüm bilgileri içerir. Örneğin olaylar ve durumlar hakkında bilgi ve bunlar

arasındaki mantıksal ilişki yapılarını ihtiva eder. Ayrıca standart çözüm ve karar alma modellerini de içerir. İnsan Uzmanından alınan bilgilerin, bilgisayarın işleyebileceği formlarda saklandığı kısımdır.

- **Çıkarım mekanizması**

US'in beynidir. Bilgi tabanı ve çalışma alanında bulunan bilgiler üzerine düşünmek için bir metodoloji sunan ve sonuçları biçimlendiren bir bilgisayar programıdır. Bir başka deyişle problemlere çözümler üreten bir mekanizmadır. Burada sistem bilgisinin nasıl kullanılacağı hakkında karar alınır.

US'in, sahip olduğu bilgisini kullanarak yeni sonuçlara varmasını sağlayan kısımdır. Çıkarım Mekanizması biraz sonra değinilecek çıkarım yöntemlerini kullanarak bir anlamda insan zekasının işlevini üstlenir

- **Çalışma alanı**

Giriş verileri tarafından belirlenmiş problem tanımları için hafızanın bir köşesinde bulunan çalışma alanıdır. Bu alan işlemlerin ara seviyelerindeki sonuçları kaydetmek için de kullanılır.

- **Kullanıcı arabirimi**

US'ler, kullanıcı ile bilgisayar arasında probleme yönelik iletişimin sağlanması için bir dil işleyici içerir. Bu iletişim, en sağlıklı doğal dil ile yapılır. Kısaca kullanıcı ara birimi kullanıcı ile bilgisayar arasında bir çevirmen rolünü üstlenmiştir. Kullanıcının US, US'in kullanıcı ile etkileşimini sağlar.

- **Açıklama**

US'leri diğer sistemlerden farklı yapan bir özelliği de açıklama modülünün olmasıdır. Açıklama modülünden kasıt, kullanıcıya çeşitli yardımların verilmesi ve soruların açıklanması olduğu kadar, US'in çıkardığı sonucu nasıl ve neden çıkardığını açıklayabilmesidir. Burada US karşılıklı soru cevap şeklinde davranışlarını açıklar.

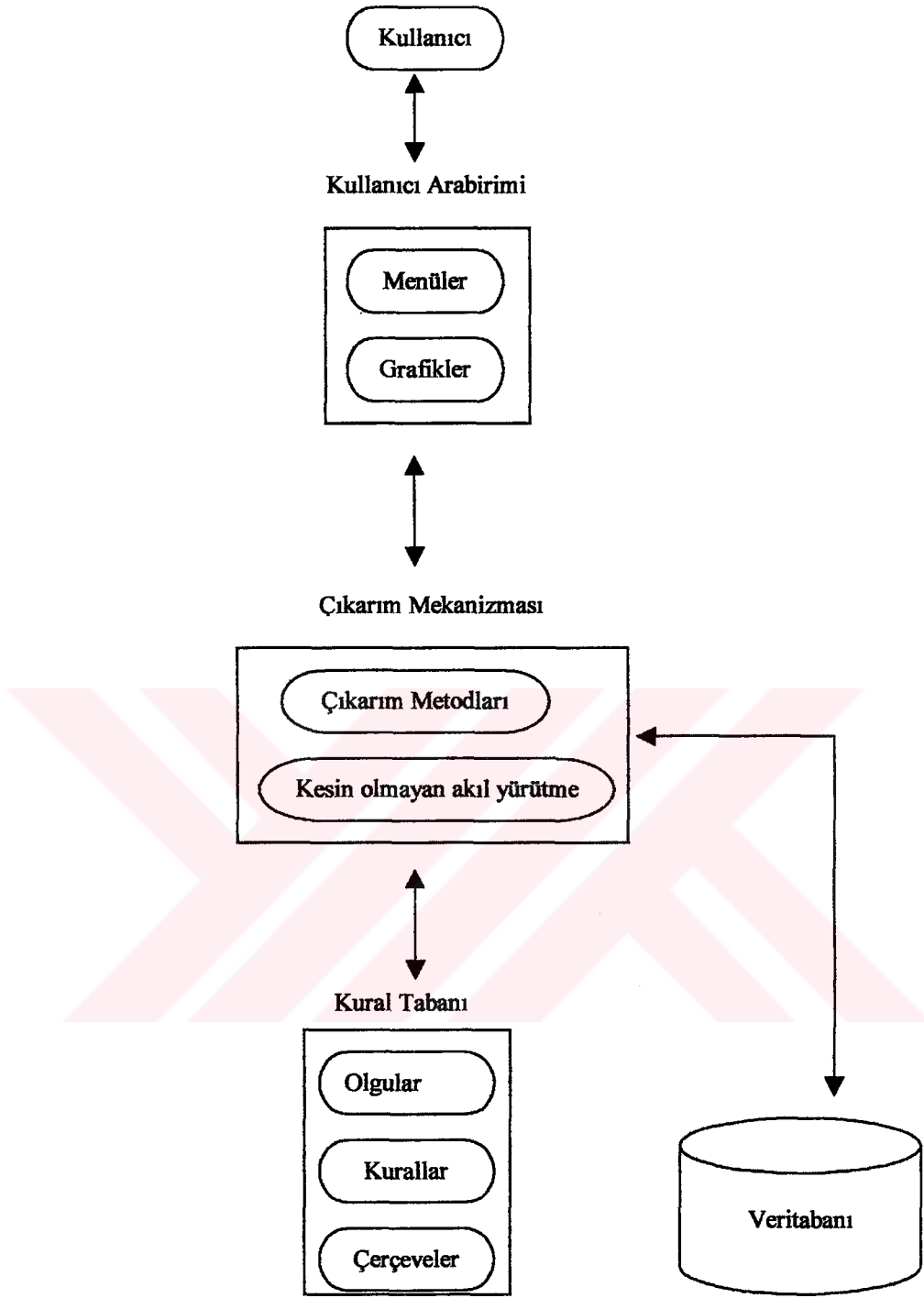
- **Düşünme kapasitesini iyileştirme**

Bir uzman insan kendi performansını analiz edebilir, öğrenebilir ve gelecekteki kullanım için onu iyileştirebilir. Sistemlerin de bu tip davranışlar göstermeye ihtiyacı vardır. Sistemin kendini iyileştirmesi öğrenme ile ilgili bir konudur. Sistemlerin bir uzman insan gibi öğrenebilmelerine yönelik çalışmalar sinirsel ağlar üzerinde sürdürülen araştırmalarla devam etmektedir. Amaç bir insan beyni gibi çalışan YZ geliştirebilmektir.

- Kural tabanı kısmı

Sistemin sorgulama sırasında kullandığı kuralları, gerçekler kısmı ise doğruluğu veya yanlışlığından emin olunan gerçekleri barındırır.





Şekil 2.1. US'nin Temel Bileşenleri

2.8. US'in Faydaları

- **Maliyet azalması**
US kullanımı ile karşılaştırıldığında insanların incelemeleri daha pahalı görülmektedir.
- **Verimlilik artışı**
US'ler insanlardan daha hızlı çalışır. Artan çıktının anlamı, daha az sayıda insan ve daha düşük maliyettir.
- **Kalite iyileştirmesi**
US'ler tutarlı ve uygun bilgiler vererek ve hata oranını düşürerek kalitenin iyileştirilmesini temin ederler.
- **İşleyiş hatalarını azaltma**
Bir çok US hatalı işlemleri tespit etmek ve onarım için tavsiyelerde bulunması için kullanılır. US ile bozulma sürelerinde önemli bir azalmanın sağlanması mümkündür.
- **Esneklik**
US'lerin kullanımı üretim aşaması ve servis sunulması sırasında esneklik sağlar.
- **Daha ucuz cihaz kullanımı**
İzleme ve kontrol için insanların pahalı cihazlara bağlı kaldığı durumlar vardır. Fakat US'ler ile aynı görevler daha ucuz cihazlarla yerine getirilebilir.
- **Tehlikeli çevrelerde işlem**
Bazı insanlar tehlikeli çevrelerde çalışırlar. US'ler ise insanların tehlikeli çevrelerin dışında kalmasına imkan sağlar.
- **Güvenilirlik**
US güvenilirirdir. US bilgilere ve potansiyel çözümlere üstün körü bakmaz, tüm detayları yorulmadan ve sıkılmadan dikkatlice gözden geçirir.
- **Cevap verme süresi**
US'ler, özellikle verilerin büyük bir kısmının gözden geçirilmesi gerektiğinde bir insandan çok daha hızlı cevap verecektir.

- Tam ve kesin olmayan bilgi ile çalışma
Basma kalıp bilgisayarlar ile karşılaştırıldığında, US'lerin insanlar gibi tam olmayan bilgi ile çalışabildiği görülmektedir. Bir görüşme sırasında sistemin bir sorusuna kullanıcı bilmiyorum veya emin değilim şeklinde bir cevap verdiğinde, US kesin olmasa bile bir cevap üretebilecektir.
- Eğitim
US'in açıklayabilme özelliği bir öğretim cihazı gibi kullanılarak eğitim sağlanabilir.
- Problem çözme kabiliyeti
US'ler, uzmanların yargılarını bütünlüğe imkân sağlayarak problem çözme kabiliyetlerini yükseltirler. Bu sistemler bilgileri nümerikten ziyade sembolik olarak işledikleri için bir çok yöneticinin karar alma stilleri ile uyumludur.
- Sınırlı bir sahada karışık problemlerin çözümü
US'ler insan yeteneklerini aşan karışık problemlerin çözümünde kullanılabilir (Patterson, 1990).

2.9. US'in Sınırları

US'lerin ticari olarak yayılmasının önündeki bazı problemler şunlardır (Patterson, 1990):

- Bilgi her zaman okunabilir uygunlukta değildir.
- İnsanlardan bilgi almak zordur.
- US'ler ancak sınırlı sahalarda, bazı durumlarda ise çok sınırlı sahalarda iyi çalışabilirler.
- Yardım için bilgi mühendisine ihtiyaç gösterir. Bilgi mühendisi az bulunur ve pahalı olduğu için sistemin maliyetini yüksektir.
- Sistemin maliyeti ve geliştirme süresi engelleyici bir faktördür.
- Her hangi bir uzmanın durum değerlendirmesi için yaklaşımı farklı bile olsa doğru olmalı.

- Çok tecrübeli bir uzman bile olsa, zaman baskısı altında olduğu zaman iyi bir durumsal değerlendirme yapması zordur.

2.10. US'in Uygulama Alanları

En yaygın US uygulama alanları (Kökyıldız, 1999):

- Mühendislik
- Askeri
- Uzay Teknolojisi
- Tıp
- Elektrik ve Elektronik
- Havacılık ve Uzay
- Tarım
- Kimya
- Hukuk
- Üretim
- Eğitim
- Karar ve Destek
- Teşhis
- Tamir ve İyileştirme

Çeşitli alanlarda kullanılan US'ler ve uygulamaları takip eden tablolarda görülmektedir.

Tablo 2.1. Jeoloji, Kimya, Elektronik ve Eğitim Alanında Bazı US'ler

Jeoloji	SPERIL	Depremden sonra binalardaki hasarları hesaplar (Lucas, 1997)
Kimya	SECS	Kimyasal analizler için
Elektronik	CRIBS	Bilgisayar donanım hatalarını teşhis eder(Lucas,1997)
Eğitim	SCHOLAR	Oyunla öğrenme (Ünsaçar ve diğ, 1998)
	EXCHECK	Matematik (Ünsaçar ve diğ, 1998)
	WHY	Yağmurun yağış nedenleri (Ünsaçar ve diğ, 1998)
	STEAMER	Mühendislik (Ünsaçar ve diğ, 1998)
	WEST	Oyunla öğrenme (Ünsaçar ve diğ, 1998)
	WUMPUS	Matematik (Ünsaçar ve diğ, 1998)
	BUGGY	Matematik(Ünsaçar ve diğ, 1998)

Tablo 2.2. Tıp Alanında Bazı US'ler (Regers et all.,1984)

Tıp	PUFF	Akciğer hastalıkları için teşhis
	PIP	Böbrek hastalıkları için teşhis
	MYCIN	Tıbbi teşhis
	INTERNIST	Tıbbi teşhis
	INTERNIST-II	Tıbbi teşhis
	EXPERT	Tıbbi danışma modellerinin etkilerini kontrol eden ve araştırmak için araştırmacılara yardım eder
	LEDI-2	Tıbbi teşhis
	UMDES	Ülser hastaları için teşhis ve tedavi
	MODIS-2	Stres bozuklukları için tedavi
	JOSEPH	Tıp öğrencilerinin eğitimi için yardım
	ONCO-HELP	Tümör hastaları için tedavi etmede yardımcı olma
	PHARM-2	Uyuşturucu madde tedavisi için yardımcı olma
	QUAWDS	Felçli hastalar için teşhis ve tedavi için yardımcı olma
	XDIS	Tıbbi teşhis
	SETH	İlaç zehirlenmelerinde yardımcı olma

Tablo 2.3. Tarım Alanında Bazı US'ler

Tarım	CUPTE	Salatahık üretimi (McClure ve diğ., 1996)
	CITEX	Portakal üretilmesi için (Allahverdi, 1998)
	MAIZE	Uçakla bitkilere pestisid verildiğinde hava akımının ürüne verdiği zararı ölçmek ve bunun karşısını alabilmekten ötürü gerekenleri tavsiye etmek için tasarlanmıştır (Allahverdi, 1998)
	PLANTING	Bitkilerle uğraşanlar, tohum üretenler ve bu alandaki talimcilere danışma yapmak için tasarlanmıştır (McClure ve diğ., 1996)
	SEPA	Çiftlikleri planlamak, yaz ve kış tahıl ürünlerini hayvan yemi otları ile kombineli olarak ekmek ve benzeri işler için tavsiyelerde bulunmaktadır (Delgado, 1996)
	MAIZE	Tarım işletmecilerine kendi müşterilerine pestisidler konusunda yardım etmek amacı ile tasarlanmıştır (Allahverdi, 1998)
	MAIZE	Ürün mevsimini planlar (McClure ve diğ., 1996)
	PNUTGRO	Uygun sulama kararı verebilmek (Parmar et al, 1997)
	TOMATEX	Domates bitkisinin hastahıklarını teşhis ve tedavi (McClure ve diğ., 1996)

Tablo 2.4. Havacılık ve Uzay Alanında Bazı US'ler (Hakima, 1996)

Havacılık ve Uzay	INSIDE	Uçaklarda navigasyon teşhisi yapar
	FICX	Uzay aracının ana motoruna yüksek basınçlı pompalamayı kontrol eder
		Türk Hava Yolları 310 Airbus arıza teşhisi
		Kokpit radarında hedef seçimine yardım eder
	FDI	Uçaklarda arıza teşhisi
	EDIS	Uzay Mekiği Ana Motor görevlerini düzenler.
	RECMS	F-15 ve F-16 simülasyonu
		F-16 sistem kontrol simülasyonu
	CEPS	Uçak teşhis araç takımı
	Scotty	Uzay Mekiği Ana Motor Ateşlemesine yardım eder
	FCMDS	F-16A Uçak kontrol sistemi
	DITFED	CF-5 Uçak motoru arıza teşhisi

3. US'DE BİLGİ GÖSTERİM YÖNTEMLERİ

3.1. Kural Tabanlı Gösterim

Kural tabanlı sistem, uygulanacak kurallar için 'eğer(şart)-öyleyse(sonuç)' (if (condition)-then(action)) formunda gösterir (Waterman, 1985). Bu kurallar, koşullardan oluşan eğer cümleci ile sonuçlardan oluşan öyleyse cümleciğinden oluşur.

Şayet: Şart-1 ve Şart-2 ve Şart-3

Öyleyse: Sonuç

ŞAYET:Hasta kronik bir rahatsızlığı var, ve
Hastanın cinsiyeti kadın, ve
Hastanın yaşı 30'dan küçük, ve
Hasta A durumunu gösteriyorsa, ve
Test B, biyokimyasında C durumunu ortaya çıkarıyorsa.

ÖYLEYSE: Sonuç:Hastanın teşhisi hepatitdir.

Prolog'da kural tabanlı bilgi gösterimi (Patterson, 1990):

sonuç(hasta, teşhis, hepatit):-

benzer(hasta, rahatsızlık, kronik),
benzer(hasta, cinsiyet, kadın),
daha_küçük(hasta, yaş, 30),
benzer(hasta, semptom_a, değer_a),
benzer(hasta, biyokimya, değer_c).

Kural taban sistemler çok kuralı gösterime izin verirler. Bir gösterim, veri girişi için ve bir başka gösterim açıklamalar için olabilir. Açıklamalar, doğal dil içine karara katılan kuralları çevirerek genellikle üretilirler. Kural tabanlı sistemlerde, bilgi yalın, bileşik önermeler ve şart-sonuç kuralları gibi temsil edilir. Bu sistemler hem ileriye hem de geriye zincirleme kullanırlar.

Kural tabanlı sistemler, varolan bilgiyi sadeleştirmekte ve açıklamakta faydalı olurlar. Pratik anlamlarda karmaşık bilgi esası içindeki yetersiz anlaşılan bilgiyi çevirmenin paha biçilmez olduğunu ispatlarlar.

Geleneksel programlamanın aksine, kural tabanlı programlama yorumsaldan daha çok analitiksel düşünmek gerekir.

Kural tabanlı sistemler, başarılarını uzmanlaşmış yoğun bilgi sisteminin bilgiyi anlaşılır bir şekilde göstermesine ve bilgiye kolayca ulaşılmasına borçludur.

3.2. Çerçeve Tabanlı Gösterim

Çerçeve gösterimi, Minsky tarafından geliştirilen çerçeveler teorisine dayanır. Bilgi, slotlar olarak adlandırılan özellik tanımlarının setlerini birleştirebilen çerçeveler adı verilen veri yapılarında temsil edilir. Nesneleri organize etmek için, bilgi cinsine göre sınıflandırılmış yapılarda gösterilirler ve türetilirler.

Çerçeve, bütün olarak sınıfın tanımları olduğu gibi, sınıfın üyelerinin prototip tanımlarını içerir. Bu yola, çerçeveler, nesnelerin yapısal gösterimlerini veya nesnelerin sınıflarını oluştururlar. Bu ilk örnekler, nesnelerin varsayılan tanımlarını yaratmakta kullanılabilirler.

Çerçeve sistemleri çıkarım tabanlı türetmede özellikle güçlüdürler, çünkü çerçeveler arasında cinsine göre sınıflandırılmış ilişkiler, türetme tarafından çoklu çerçeveler arasında bölüşülmek için tanımlayıcı bilgiyi seçilir kılarlar.

Çerçevede veriyi kullanma işlemlerin esasen üç tipi vardır. Bunlar kısıtlar (constraints), hayaletler (demons) ve denetçilerdir (watchdogs). Bir slot güncelleştirildiğinde kısıtlar harekete geçirilirler. Kısıtlar, yanlış veya konu dışı bilgi slot konumuna getirilmediğinden emin olmak için harekete geçerler. Hayaletler, bir slotda bir bilginin güncellenmesinden sonra meydana gelirler ve komut satırlarını

çalıştırırlar. Denetçiler slotdaki değere erişilmeye çalışıldığı zaman harekete geçerler. Kısıtlar, hayaletler ve denetçiler etkileri uygulamaya bağlıdır. Her bir prosedür o slotdaki veri ile ilişki kurulan özel slotla bağlanmıştır. Bu prosedürler, diğer çerçeveler tarafından türetilirler. Şekil 3.1.'de çerçeve tabanlı gösterim slotlarla gösterilmektedir (Parsaye ve Chignell, 1988).

Çerçeve : Çerçeve İsim

Sınıf : Çerçeve Sınıf-1

.....

Sınıf : Çerçeve Sınıf-2

Slot : Slot-1

Slot : Slot-2

.....

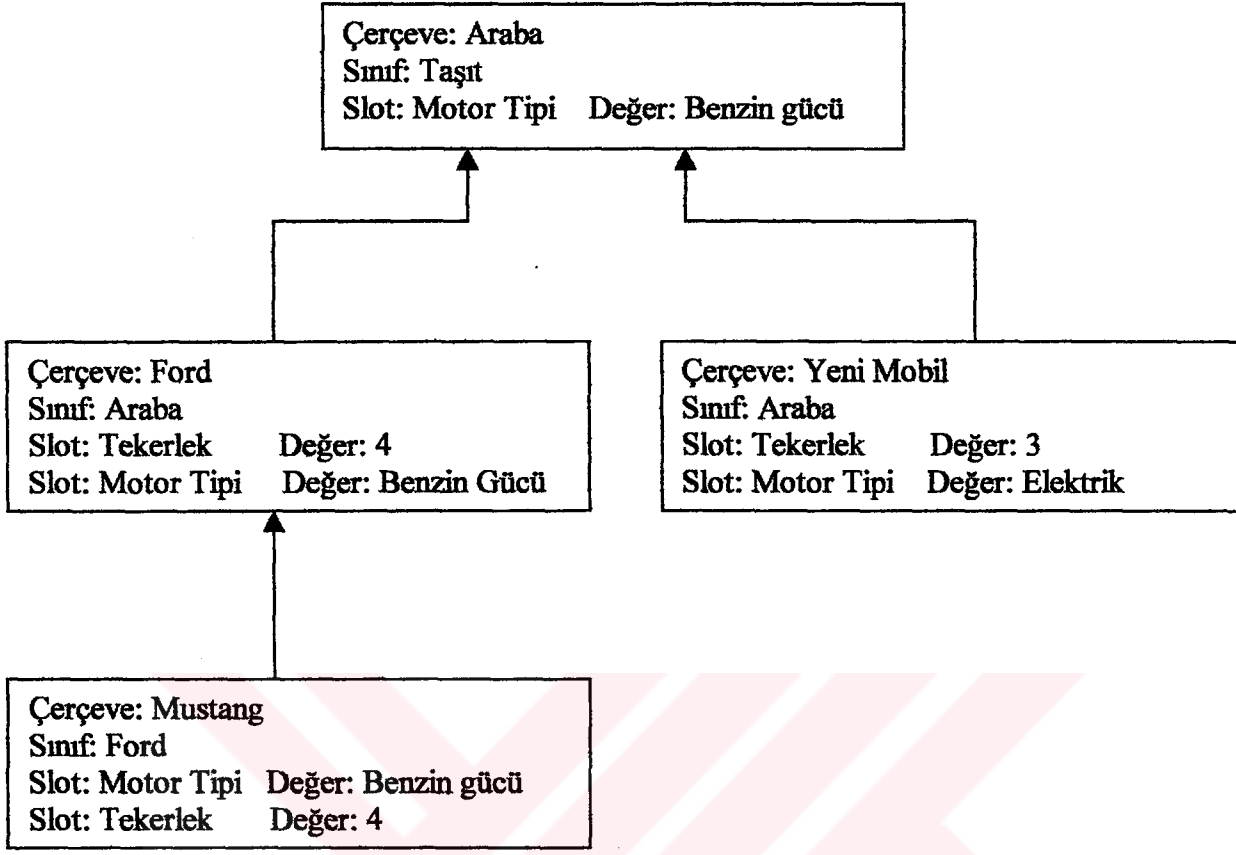
Slot : Slot-N

Bir Ford Escort bir Ford'dur.

Bir Ford bir otomobildir.

Bir otomobile dört tekerleğe sahiptir.

Bir otomobil bir motora sahiptir.



Şekil 3.1. Çerçeve Tabanlı Gösterim

3.2.1. Çerçeve Tabanlı Bilgi Gösterim Dilleri

US paketleri kullanıcının sadece bilgi tabanını oluşturarak çeşitli uygulamalar yapabilmesi için tasarlanmış özel amaçlı paketlerdir. Tablo 3.1.'de, bazı çerçeve tabanlı US paketleri ve bunların geliştiricileri görülmektedir (Karp, 1993).

Tablo 3.1. Bazı Çerçeve Tabanlı US Paketleri ve Geliştiricileri

İSİM	GELİŞTİRİCİ
KRL	Bobrow ve Winograd, 1997
FRL	Roberts ve Goldstein, 1977
RLL	Greiner and Lenat, 1980
KL-ONE	Brachman and Schmolze, 1985
KRYPTON	Brachman, 1985
NIKL	Kaczmarek, 1986
CYCL	Lenat and Guha, 1990
THEO	Mitchell, 1989
FRAMEKIT	Nyberg, 1988
KEE	Fikes, 1985
OBJLOG	Chouraki, 1987

3.3. Mantık Tabanlı Gösterim

Mantık tabanlı sistemlerde, bilgi, basit ve karmaşık yüklem deyimleri ile temsil edilir. Böyle sistemler, ilk-sıra yüklem hesabına dayanırlar. Bu hesap, anlamlı güç ve iyi geliştirilmiş biçimcilik sağlar. Çözüm etkinliği olgulardan elde eden bir işlemdir ve uygun standart forma çevirilen deyimler üzerinde çalışır. Mantık tabanlı sistemler, çıkarım mekanizmaları için ispatlayan çözüm teoremine güvenirlir. Bu sistemlerde bilgi, şayet-öyleyse kuralları veya prosedürler şeklinde temsil edilir.

Çekirdek dil olarak Japon beşinci kuşak hesaplama sistemler projesi yoluyla mantık programa dili olarak Prolog'un kabul edilmesi, bilimsel ve teknik programlamada bu dilin sağlamlaşmasına katkıda bulundu. Aşağıdaki mantıksal gösterimde;

- '∀' 'her bir',
- '→' 'eğer- ise',
- 'V' 'veya' ve
- 'Λ' 've' anlamında kullanılmaktadır.

$$\forall x : \text{evcil_hayvan}(x) \wedge \text{küçük}(x) \rightarrow \text{ev_hayvanı}(x)$$

$\forall x : \text{kedi}(x) \vee \text{köpek}(x) \rightarrow \text{evcil_hayvan}(x)$
 $\forall x : \text{fino_köpeği}(x) \rightarrow \text{köpek}(x) \wedge \text{küçük}(x)$
 $\text{fino_köpeği}(\text{sadık})$

Prolog'daki mantıksak gösterimin aşağıdaki gibidir (Rich ve Knight, 1991).

$\text{ev_hayvanı}(X) :- \text{evcil_hayvan}(x), \text{küçük}(x).$
 $\text{evcil_hayvan}(x) :- \text{kedi}(x).$
 $\text{evcil_hayvan}(x) :- \text{köpek}(x).$
 $\text{köpek}(x) :- \text{fino_köpeği}(x).$
 $\text{fino_köpeği}(\text{sadık})$

Yüklemler mantığında karmaşık yapıli tanımlamalar yapmak kolay değildir. Bu yüzden, çerçeve sistemler mantık tabanlı sistemler daha uygundur.

Benzer şekilde, türetme tabanlı ve sınıflandırma tabanlı çıkarım çerçeve tabanlı sistemde daha kolay uygulanırlar. Çerçevelerde temsil edilen tanımlayıcı bilgi, yüklemlier mantığı deyimlerine çevirilebilir. Ama, bu çevirilerde çerçevenin tarafından temsil edilen bilginin dış yapısı genellikle kaybedilir (Rich ve Knight, 1991).

3.4. Anlamsal Ağlar

Anlamsal ağlar (Semantic Networks) bilgi gösteriminin önemli diğeri şeklidir. Bu gibi sistemlerde, bilgi düğümlier ve yayların gösterilir. Her düğüm bir nesneyi ya da bu nesnenin nitelik veya değeriini gösterir. Yaylarda düğümlier arasındaki bağlantıyı gösterirler (Topçu, 1995).

Anlamsal ağ ile yüklemlier mantığı gösterimi arasında kuvvetli yapısal paralellik vardır. Türetme özellikleri ve ilişkileri taşımak için kullanılabilen anlamsal ağlarda, tanımlayıcı bilgi kavram hiyerarşileri içinde organize edilebilir. Böyle hiyerarşilerin örnekleri 'is-a' ve 'instance-of' hiyerarşileridirler. Instance-of : Bir nesne ile sınıfı arasındaki ilişkiyi gösterir. Is-a: Bir sınıf ile bir üst sınıf arasındaki ilişkiyi gösterir (Kocabaş, 2001).

3.5. Sınıflandırıcılarda Bilgi Gösterimi

Genetik algoritmalar dayanan sınıflandırıcılar ve sınıflandırıcı sistemleri, endüktif öğrenme sistemleridirler. Bu sistemler, bilgiyi, bir takım parametreler veya bir takım bildiri listeleri olan vektörler veya değerler matrisi olarak temsil edilir.

3.6. Genetik Algoritmalar

Genetik algoritmalar (GA), bir takım parametreler veya bir takım bildiri listelerine değerlerin vektörleri olarak bilgiyi temsil ederler. GA tümevarım öğrenme, çakışma çözümü, sınıflandırma ve kontrol kurallarının öğrenilmesini desteklerler. GA geniş, karmaşık, sayılamayan, çok boyutlu ses (gürültü) problemleri ve özellikle optimumlaştırma problemleri için geliştirilmiş ihtimali bir arama tekniğidir. Genelde GA bir alfabe (semboller takımı) üzerinde belirlenmiş karakterler dizini (zinciri) ile çalışır. Bu dizinler moleküler biyolojideki kromozomlara benzetilir. GA veriler üzerinde türeme, çaprazlaşma ve mutasyon gibi üç çeşit işlem yapabilme özelliğini kullanmaktadırlar.

3.7. Sinirsel Ağlarda Bilgi Gösterimi

Sinirsel ağlar büyük sayıda birbirine bağlı işleme birimlerin oluşmuşlardır. Bu sistemlerde, bilgi bağlantıların toplamıyla temsil edilir.

Sinir ağları (SA) girişleri ve çıkışları olan basit ve birbirleri ile sıkı bir şekilde ilişkilmiş işlem elemanları olup insan beynindeki hücrelerin çalışma prensibini modelleyen bir bilgisayar sistemidir.

Geleneksel programlama yöntemleri ile çözilemeyen bir dizi problem SA ile çözüm bulabilmektedir. İki temel durumda SA'nın kullanımı efektif olmaktadır:

- 1) Geniş veri takımlarının yorumlanması istenen incelemelerde ve
- 2) Giriş ve çıkış verileri belli, fakat bu veriler arasındaki ilişkilerin iyi bilinmemesi durumunda.

Bütün SA'nda olduğu gibi bu durumda da sonuç iki aşamadan geçilerek elde edilmiştir. Bu aşamalardan birincisi ağı eğitilme aşamasıdır. Bu aşamada girişlere rasgele ağırlık verilir ve bu ağırlıklar uygun giriş değeri ile toplanır. Sonra her bir

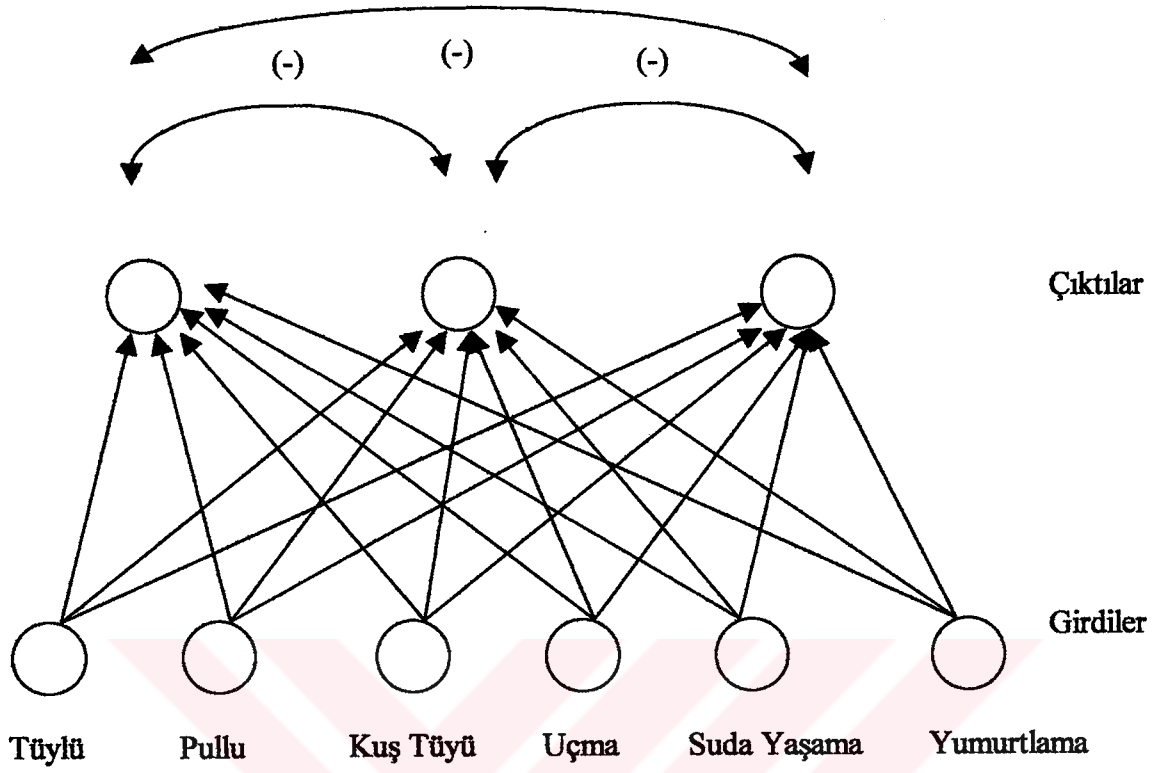
düğümüne dahil olan bu çarpımlar toplanır ve toplam belli bir kriter değerini (threshold) aşıyorsa çıkışa iletilir, aksi durumda çıkış değişmemektedir. Her bir düğüm bu işlemi yaptıktan sonra çıkış katmanındaki düğümden alınmış değer elde olan belli değerle kıyaslanır ve hata hesaplanır. Hata büyük olduğu takdirde yeniden geriye döndülür ve ağırlıklar değiştirilerek yeniden hesaplamalar yapılır. Ağırlıkların değiştirilmesi için çeşitli yöntemler mevcuttur. Bu süreç programlanarak bilgisayar yardımıyla ağız eğitimi yapılır, yani gereken ağırlıklar belirlenir (Allahverdi, 1998).

Tablo 3.2.'deki veri takımları ile memeliler, sürüngenler ve kuşlardan dokuz tane hayvanın özellik seti verilmiştir. Bu veri seti sayesinde belirli bir hayvanı gruplandırmak için bir ağ kurulur. Şu ana kadar görülmemiş bir hayvanı tanımlanmak için de bu ağ kullanılır.

Şekil 3.2.'de Girdiler doğrudan doğruya çıkışlarla bağlantılıdır. Çıkışlar kendi aralarında da bağlantılıdır. Altı tane girdi ve üç tane çıkışla ağ oluşturulur. Ağda girişlere ağırlık verilir ve çıkışlar gözlenir. Bir düğüm bu işlemi yaptıktan sonra çıkış katmanındaki düğümden alınmış değer elde olan belli değerle kıyaslanır ve hata hesaplanır. Hata büyük olduğu takdirde yeniden geriye döndülür ve ağırlıklar değiştirilerek yeniden hesaplamalar yapılır (Rich ve Knight, 1991).

Tablo 3.2. Veri Seti

	Tüylü	Pullu	Kuş Tüylü	Uçma	Suda Yaşama	Yumurtlama
Köpek	1	0	0	0	0	0
Kedi	1	0	0	0	0	0
Yarasa	1	0	0	1	0	0
Balina	1	0	0	0	1	0
Kanarya	0	0	1	1	0	1
Devekuşu	0	0	1	0	0	1
Yılan	0	1	0	0	0	1
Kertenkele	0	1	0	0	0	1
Timsah	0	1	0	0	1	1



Şekil 3.2. Veri Seti Sinir Ağı

4. PROLOG UZMAN SİSTEM ARACI

4.1. Giriş

YZ dilleri alışkın olduğumuz programlama dillerinden oldukça farklıdır. Bu farklılık YZ dillerinde program ve veri ayrımının olmamasından ileri gelir. YZ dillerinde program da veriler de semboller bütünü olarak algılanır. Bunların yanı sıra geleneksel programlama dilleri sıralı, yordamlara bağlı bir yapı gösterirken, YZ dilleri semboller (program ve veriler) arası ilişkilere dayalı olduğundan bu yapıya uymazlar. Bu ise daha dinamik bir yapı sağlar. Bir anlamda YZ programları çalıştırılmaları sırasında değiştirilebilirler. YZ dilleri aşağıdaki gibi sınıflandırılabilirler.

LISP (List Processing): ABD'de geliştirilmiş ve genellikle ABD'de kullanılan YZ dilidir. Liste işlemlerine dayalı olarak çalışır.

Prolog (Programming in Logic): Genellikle Avrupa'da kullanılan ve Japonların Beşinci kuşak bilgisayarları için seçilmiş YZ dilidir. Teorem ispatlayabilen bir yapısı olduğundan LISP'ten çok farklıdır.

Nesneye yönelik programlama dilleri : YZ çalışmalarına kullanımı kolay editörler, bit-map grafik olanakları, debug ve trace kullanımı sağlarlar(Smalltalk gibi).

Prolog 1970-1972 yılları arasında Marsilya, Fransa'da Alain Colmerauer başkanlığındaki bir grup tarafından geliştirildi. Endüstriyel amaçlı büyük YZ uygulamaları için kullanılmaya başlandı. Daha sonraları Japonların Beşinci Kuşak Bilgisayar sistemleri projesinde kullanılmaya başlandı. Çeşitli Prolog derleyicileri (Prolog-86, Micro-Prolog, Turbo Prolog, Arity Prolog, Als Prolog) bulunmakla birlikte Linux işletim sisteminde Xpce, Xpb Prolog derleyicileri bulunmaktadır. Bu

derleyiciler Linux altında çalıştıkları için grafik destekleri de standart Prolog derleyicilerinden oldukça fazladır. Ayrıca diğer Prolog programlarını da neredeyse sorunsuz şekilde çalıştırabilmektedirler. Tüm Prolog derleyicilerinin ortak özelliği yüklem teorisi sembolize edilmiş olarak içermeleri ve kendi teorem kanıtlayıcılarının bulunmasıdır. Kullanımda ise her komut işareti ile ve her oturum "halt" komutu ile sonlandırılmalıdır.

- Prolog programları bir bilgi tabanı içerirler.
- Bu bilgi tabanı gerçekler, kurallar ve çerçevelerden oluşabilir. Programlar ise bilgi tabanının sorgulanarak bir takım sonuçlar üretilmesi olduğundan, programın sonucu programın bilgi tabanındaki verilere bağlıdır. Bilgi tabanında yanlış veya tutarsız olan veriler varsa programın doğru sonuç üretmesi olası değildir.
- Bilgi tabanının sorgulanması sırasında Prolog'un geriye doğru iz sürme (backtracking) özelliğinin devreye girmesi aynı işlemin daha az komut kullanılarak yapılmasını sağlar.
- Prolog yüklem mantığını kullandığından bu durum sabitler ve değişkenlerin ayrılmasını gerektirir.
- Bunu sağlamak üzere Prolog'da sabitler, kurallar, gerçekler küçük harfle gösterilirken değişken adları mutlaka büyük harfle başlar.
- Prolog bu özellikleri nedeniyle liste işlemlerini ve yinelemeleri kolayca yapabilir; yüklem mantığına dayanması nedeniyle de sonuç çıkarımı için özel yapılar gerektirmez.
- Bu iyi yanlarına karşılık Prolog'un bazı olumsuzlukları da yok değildir. Örneğin iyi bir kullanıcı arabirimi olmadığı gibi grafik desteği de çoğunlukla yoktur. Ayrıca programların etkinliği içerdiği kuralların sırasına bağlıdır.
- Prolog programlarının yapısına bakıldığında gerçekler, kurallar, sorular ve sistem komutlarından oluştuğu görülür. Gerçekler sabitleri veya sabitler arasındaki ilişkileri açıklamada kullanılırken, kurallar sabitler ve gerçeklerin burada kullanılmasıyla oluşturulur.

4.2. ALS Prolog Özellikleri

ALS Prolog, eşsiz programlama dilidir. BASIC, FORTRAN VE PASCAL benzeyen diğer diller yordamsal diller sınıflandırılır iken, ALS Prolog bir bildirim dilidir. 1986 yılında Applied Logic System tarafında geliştirilmiştir (ALS Inc, 1986).

Yüklem mantığı, Sembolik Mantık'ın iyi geliştirilmiş bir kısmıdır. Bir Prolog programcısı sadece programcı değildir aynı zamanda bir mantıkçıdır yada aslında bilgiyi mantıksak bir yolla gösterebilen herhangi biridir.

Bilgi olgular ve kurallar olarak girilir. Kullanıcı, bu gerçekleri ve kuralları ilgilendiren sorular sorduğunda, Prolog cevapları bulmak için gerekli hesapları yapar. Eğer cevaplar, bu gerçekler ve kurallardan anlaşılabilirlerse, terminalde gösterilirler. Eğer anlaşılamazsa, Prolog 'no' mesajı gösterir.

ALS Prolog YZ uygulamaları için bilgisayarla çeviri, mantık tabanlı problem çözme ve bilgi tabanlı sistemler gibi çok uygundur.

Basit bilgiyi esaslarını geliştirmek için Prolog'u nasıl uygulamak gerektiğini ve Prolog sorularının basit formlarını kullanarak böyle sistemlerden bilgi elde etmek için Prolog deyimlerini kullanırız.

Prolog deyimleri üç çeşittir (Kocabaş, 1986):

1. Olgular
2. Sorgular
3. Kurallar

4.3. Prolog Olguları

Dört tip Prolog olgusu bulunur (Kocabaş, 1986).

- 1) Basit Olgular
- 2) Değillemeler
- 3) Yapılar
- 4) Bileşik Olgular

İ.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

4.3.1. Basit Olgular

Aşağıdaki cümleler basit olgular sayılabilirler.

Paris is in France
Italy is a country in Europe
Spain is larger than Italy
A cat is an animal
A bear eats honey
A zebra has black stripes
5 is greater than 3
x is less than 7
1 is equal to 1
Oxygen is an element
Iron is a metal
Sulphur is yellow

Prolog olguları olarak ele alınan cümleler Prolog gösterimi ile yazılmalıdır.

Cümle	Prolog olgusu
Paris is in France	is_in(paris,france).
Italy is a country in Europe	country_in(italy,europe).
Spain is larger than Italy	larger_than(spain,italy).

Prolog olgusu kısmındaki kelimeler; paris, france, italy, europe, spain, italy **ÖZEL İSİMLER** olarak adlandırılırlar ve is_in, country_in, larger_than gibi kelimeler **İLİŞKİLİ İSİMLER** yada **ÖZELLİK İSİMLERİ** olarak adlandırılır. Prolog'da, özel isimler ve ilişki isimler tek kelime veya alt çizgi '_' ile bağlantılı kelime dizileri olabilirler. Özel isimler örnekleri:

paris, france, zebra, black_stripes, cat, oxygen, iron, sulphur.

Özel isimler küçük harfle başlamalıdır. Değişkenler dışındaki bütün kelimeler küçük harfle başlar ve "." nokta ile son bulur. İlişkili isimler de küçük harfle başlamalıdır. Benzer olarak, ilişkili isimler için örnekler:

is_in, is_a, larger_than, eats, greater_than, element, metal, yellow.

Ancak, Prolog'un var olan fonksiyon isimleri Prolog'da özel veya ilişkili isimleri olarak kullanılamazlar. Değişken isimleri alfabenin büyük harfleri olabilir.

A,B,C,...,X,Y,Z

Büyük harf veya alt çizgi ile başlayan değişkenler tanımlanabilir.

Plant, Animal, Planet, _plant, _animal, _planet.

Alt çizgi, aynı zamanda değişken belirtisidir ve anonim değişken olarak adlandırılır. Bir cümleyi bir Prolog olgusuna çevirmek için bazı faydalı kurallar vardır. Bu kurallar:

1. Cümleleri İngilizce yaz.
2. Özellik yada ilişkili isimleri sola yaz (bu bir kelime olmalı yada kelimeler ' ' "alt çizgi" ile birbirine bağlanmalı).
3. Sol parantezi koyduktan sonra yazılan ilk özel ismi virgül takip eder. Sonra, sonraki özel isimler soldan sağa doğru aralarına virgül koyularak yazılır. Parantezler kapatılır ve en sona '.' "nokta" koyulur.

Cümle

Italy is a country in Europe

Prolog Olgusu

country_in(italy,europe).

Basit olgunun genel şekli:

ilişkili_isim(1.özel isim,2.özel isim,...).

Prolog olguları değişkenleri de kapsar ve özel isimler hakkında bilgi ediniriz.

Cümle

X is a country in Europe

Prolog Olgusu:

country_in(X,europe).

Bunun anlamı Avrupa da bir ülke var. Prolog'da bu tip değişkenler çok kullanışlıdır. Prolog sorgularında yanıtı bulmakta kullanılır. Değişkenler ile Prolog olguları aşağıda yazılmış (Kocabaş, 1986).

Prolog Olgusu

larger_than(X,italy).

Anlamı

[X is larger than Italy]

less_than(0,_).

[0 is less than anything]

4.3.2. Değillemeler

Prolog olguları kümesine Prolog programı veri tabanı gibi başvurulur. Bir Prolog programının veri tabanına girilmeyen her bir olgu gerçek değilmiş gibi bir anlaşılır. Prolog'da soru sorulduğu zaman bundan dolayı dikkatli olmak gereklidir.

Cümle

Prolog

The Sun is larger than the Earth.

larger_than(sun,earth).

Bu olgu veri tabanında bulunmuyorsa, Prolog Güneş Dünyadan büyük olup olmadığını sorduğumuzda olumsuz cevap verir. Prolog'da bu 'Değilleme Hatası' olarak adlandırılır (Kocabaş, 1986).

4.3.3. Yapılar

Prolog, tanımları içeren olguları temsil edebilir. Tanımlamalar, isimlerini kullanmadan belirlenmiş nesnelerde kullanılırlar.

The satellite of the Earth has craters.

'The satellite of the Earth' ifadesi Moon'u belirten bir tanımlama olarak adlandırılır. Bu tanımlayıcı olgu, Prolog'da aşağıdaki gibi yazılabilir.

has_craters(satellite_of(earth)).

Benzer olarak, bu cümlede: 'Mars is larger than the satellite of the Earth,' aşağıdaki gibi yazılır.

larger_than(mars,satellite_of(earth)).

Böyle tanımları içeren Prolog olguları, yapılar olarak adlandırılırlar (Kocabaş, 1986).

4.3.4. Bileşik Olgular

İki yada daha fazla yantümce bağlaçlar ile bir araya getirildiği zaman, ',' virgül yada ';' "noktalı virgül" cümleleri bileşik olguları temsil eder.

And , (ve)

Or ; (veya)

The Moon is smaller than the Earth and the Earth is smaller than Jupiter.

Bu cümlelerin iki yantümcesi var:

- 1 The Moon is smaller than the Earth
- 2 The Earth is smaller than Jupiter

iki yada daha çok yantümce 've' bağlacı ile aşağıdaki gibi bileşik bir cümle yapılmak için birleştirildiyse, bu bileşik cümle 'birleşim' olarak adlandırılır. Birleşimler Prolog'da ','(ve) ile birleştirilmiş iki yada daha çok ayrı olgular gibi temsil edilirler.

Örnek olarak, bileşik cümle:

The Moon is smaller than the Earth and the Earth is smaller than Jupiter

Prolog'da iki ayrı yan tümcede aşağıdaki gibi yazılır.

smaller(moon,earth), smaller(earth,jupiter).

Bu iki birleşmiş yantümceler bir Prolog programına ayrı ayrıda girilebilir. Kesinlikle, bu iki bileşik yantümcede, aynı etki ile ayrı ayrı olarak Prolog programı giriş yapılabilirler.

smaller(moon,earth).
smaller(earth,jupiter).

Buna göre, bileşik cümle aşağıda gösterildiği gibi tümceler ayrılabilir.

Venus is nearer to the Sun than the Earth and Mars is smaller than the Earth
and Neptune is a planet.

Venus is nearer to the Sun than the Earth.

Mars is smaller than the Earth.

Neptune is a planet.

Bütün bileşik tümce, üç yantümce olarak Prolog'da ifade edilebilir (Kocabaş, 1986).

nearer_to(venus,sun,earth),
smaller_than(mars,earth),
planet(neptune).

4.4. Prolog Kuralları

Prolog kuralları iki çeşit koşullu önermeye sahiptir.

Aşağıdaki gibi ifade edilen:

Basit Kurallar

If - then.

If (- not-) then

Bileşik Kurallar

If.-and-and-then

If-or-or-then

If-and-(-or-) then

Prolog'da, kurallar, veritabanında olgular vasıtasıyla bireyler hakkında bilgiyi ortaya çıkarmak için kullanılırlar (Kocabaş, 1986).

Basit ve bileşik koşullu önermelerin bazı örnekleri:

If Venus is a planet, then it is a celestial body.

If something is not wet, then it is dry.

If an animal flies and has wings and feathers, then it is a bird.

If it is dark or your eyes closed, then you can not see.

If an animal has four legs or horns, then it is not a bird.

If an animal has two legs and can fly or swim, then it is a bird.

4.4.1. Basit Kurallar

Aşağıdaki olguların bulunduğu veritabanını düşünelim.

planet(mars). [Mars is a planet]

planet(mercury). [Mercury is a planet]

planet(venus). [Venus is a planet]

Bir Prolog kuralı olmadan, bu olgu veritabanında olmadığı için, Marsın bir gökcismi olduğunu ortaya çıkarmak mümkün değildir. Bunu yapabilmek için kural eklemek gereklidir:

If X is a planet then X is a celestial body.

Bu, aşağıda gösterildiği gibi Prolog'da yazılabilir.

```
celestial_body(X):- planet(X).
```

Prolog'da, 'then-kısım' daima sola, 'if-kısım' daima sağa yazılır ve ':' (iki nokta üst üste-tire) ile devam eder. Prolog kuralı, aşağıda gösterildiği gibi okunur. Bir başka deyişle, 'eğer X bir gezegen ise, X gökcismidir'.

```
celestial_body(X):- planet(X).  
then          if
```

İki yantümce olduğu bu örnekte:

```
celestial_body(X)    planet(X)  
then-kısım          if-kısım
```

Prolog kurallarına göre, ilk yantümce her zaman kuralın 'then-kısım'dır. Takip eden yantümce (yada tümceler) kuralın 'if-kısım'dır.

```
celestial_body(X):- is_a_planet(X).
```

'celestial_body(X)' 'then-kısım'dır. ':' anlam işaretidir. 'is_a_planet(X)' kuralın 'if-kısım'dır. Basit kuralların genel şekli aşağıdaki gibidir.

s(a):- q(a,b).	anlamı: If q(a,b) then s(a)
s(b,c):- q(c).	If q(c) then s(b,c)
t(a):- not(q(a,b)).	If not(q(a,b)) then t(a)
t(a,Y,Z):- s(a,Y,Z).	If s(a,Y,Z) then t(a,Y,Z)
s(X,Y):- q(X,Y).	If q(X,Y) then s(X,Y)

q,s,t ilişkili isimleri,a,b,c özel isimleri ve X,Y,Z değişken isimlerini gösterir.

4.4.2. Bileşik Kurallar

Bileşik Kurallar koşulu ifadelerdir.

```
If - and.- then  
If.- or - then  
If.- and (- or-) then
```

Birleşik kurallar (conjunctive rule) 'if-kısmı'da yan cümlecikleri 'and' (ve) ile birleştirilmesiyle tanımlanır. 'and' ifadesinin Prolog'daki karşılığı ',' virgüldür.

If - and - then

Örnek:

Bu kuralın anlamı : Şayet bir şey kanatlı ise ve tüylü ise, o kuştur.

bird(X):- has_wings(X),has_feathers(X).

Görüldüğü gibi ikinci ve üçüncü cümleler arasında virgül 'and' ifadesi 've' anlamına gelmektedir.

If has_wings(X) and has_feathers(X), then bird(X).

Başka bir deyişle, Prolog bir kuraldaki ilk yan cümlecik 'then-kısmı'dır ve virgül ile birleştirilmiş yan cümlecikler 'if-kısmı'dır.

Cümle

Prolog

If q(b,c) and s(a,c) and u(b,c), then t(a,c). t(a,c):- q(b,c),s(a,c),u(b,c).

Ayrık kurallar (disjunctive) tarif edilir. Çünkü yan cümleciklerin mantıksal ifade olan 'or' ile olgunun 'if-kısmı'nda birleşmesiyle tanımlanır.

Örnek:

If something has feathers or a beak, then it is a bird

Bir Prolog kuralının programda aşağıdaki şekilde ifade edilir.

bird(X):- has(X,feathers);has(X,beak).

'if-kısmı'nda, noktalı virgül ';' kullanılır. Prolog'da noktalı virgül 'or' yerine kullanılır.

Örnek:

Ayrık Prolog kuralını düşünelim:

is_a(X,animal):- is_a(X,dog);is_a(X,cat).

Anlamı: If something is a cat or dog, then it is an animal.

(Şayet bir şey bir kedi veya bir köpek ise, o bir hayvandır.)

4.5. Tekrarlılık

Şimdiye kadar sunulan basit ve karmaşık kuralların örnekleri, Prolog kural tabanının hacmini oluşturan kuralların tipindendir. Ancak daha çok karmaşık gerçekler modellemek bazen gereklidir (Kocabaş, 1986).

Örnek 'ancestor_of' bağıntısıdır. Prolog'da bu bağıntı, aşağıda gösterildiği gibi iki kuralda tanımlanabilir.

```
ancestor_of(X,Y):- parent_of(X,Y).  
ancestor_of(X,Z):- parent_of(Y,Z),  
                    ancestor_of(X,Y).
```

İlk kuralın anlamı : If X is a parent of Y, then X is an ancestor of Y.

İkinci kuralın anlamı : If Y is a parent of Z and X is an ancestor of Y,
then X is an ancestor of Z.

Çünkü 'ancestor_of' bağıntısı, bu bölgede bireyler arasında yinelenir, buna tekrarlamalı tanım denir. Tekrarlanan tanım, bir ya da birden çok karmaşık kurallardan ve bir ya da birden çok basit olgular veya kurallar oluşur. 'ancestor_of' ilişkisinin tanımında, basit kural:

```
ancestor_of(X,Y):- parent_of(X,Y).
```

ve bileşik kural:

```
ancestor_of(X,Z):- parent_of(Y,Z),  
                    ancestor_of(X,Y).
```

4.6. Listeleme Yöntemi

İsimlerin, tümceler ve işlevlerin listeleri, bilgiyi düzenlemenin vazgeçilmez ortalamalarıdır. Kitaplık da başlıkların listeleri, telefon rehberlerinde telefon numaraları ve sözlükte sözcükler yararlı listelerin örnekleridir. Listeler, çok daha kolaylıkla bilgiyi yerleştirebilmek için genellikle çeşitli düzenlerde tutulmak için kullanılırlar.

Listeler, kendileri listelenilebilen elementlerden oluşurlar. Mesela, kitabın bölümleri, liste sayılabilir. Bu listelerin elementleri bu bölümlerdeki tümcelerdir. Benzer biçimde, tümceler öğelerini kapsayan sözcükler de liste sayılabilirler.

Prolog'da, listeler Prolog yan cümleciklerinin argümanları olarak temsil edilirler ve köşeli parantez içinde yazılırlar.

```
planet([mercury,venus,earth,mars]).
satellites_of(jupiter,[io,europa,ganymede]).
satellites_of(venus,[]).
```

İlk Prolog olgusu 'planet' yüklemine argümanı olan gezegenlerin listesini gösterir. İkinci olsun Jupiter'in uydularının bazılarının listesini gösterir. Sonuncu olguda Venüs'ün uydularının listesini göstermektedir. Liste '[]' boş liste olarak adlandırılır.

Listeleme yönteminin iki önemli kavramı vardır.

Bunlar:

- 1) Listenin ilk elemanı
- 2) Listenin geri kalanı

Bu sınıflandırma, aşağıda dört Prolog listesine uygulanır.

[saturn]	İlk:	saturn
	Kalan:	[] (boş liste)
[larger,sun,jupiter]	İlk:	larger
	Kalan:	[sun,jupiter]
[]	İlk:	(İlk eleman yok)
	Kalan:	[]
[[a,b],c]	İlk:	[a,b]
	Kalan:	[c]

İlk-kalan ilişkisi Prolog'da, X'in ilk elemanı yada [X|Y] listenin başı ve Y kalan yada listenin kuyruğu olduğu [X|Y] gibi gösterilir.

Prolog Olgusu:

```
fruits([apple,pear,banana,melon]).
```

Bu olgunun yapısı, `fruits(X)` şeklinde gösterilebilir. 'fruits' özellik ismidir ve X olarak adlandırılan meyvelerin listesidir: `[apple,pear,banana,melon]`. Bu liste `[apple|Y]` gibi yazılabilir. Burda apple listenin başı ve Y listenin kalan kısmıdır.

Soru : ?- `fruits([apple|Y])`.

Prolog cevabı : `[pear,banana,melon]`

4.7. Prolog Fonksiyonları

Prolog'un kendi fonksiyonları tanıtıldı. Bu konuda bu fonksiyonlar açıklanır.

?- , ; not

Bunlardan bazıları aşağıdakilerdir:

`listing, clause, retract, asserta,`
`assertz, read, write, call,nl, fail, !,`
`functor, arg, =.., op, var, atom, integer,`
`=, \=, is, +, -, *, /`

`listing`: Bu konksiyon olguları ve kuralları ekrana listelemek için kullanılır. Örnek olarak, aşağıdaki olguları ve kuralları;

`satellite(titan).`
`planet(venus).`
`satellite(moon).`
`orbits(X,sun):- planet(X).`

Ekranda bu fonksiyonu kullanarak listeleriz.

?- `listing`.

Prolog'un cevabı:

`satellite(titan).`
`planet(venus).`
`satellite(moon).`
`orbits(X,sun):- planet(X).`

Ekranda aşağıdaki fonksiyon girildiğinde 'satellite' ile ilgili bu olgu ve kuralları listeler.

```
?- listing(satellite).
```

Cevap:

```
satellite(titan).
```

```
satellite(moon).
```

clause(X,Y): Bu fonksiyon bir Prolog programında bir olgunun yada kuralın bir cümle olup olmadığını kontrol eder. 'cause' iki argüman alır. İlk argüman kuralın 'then-kısmı' ile eşleşir ve ikinci argüman 'if-kısmı' ile eşleşir.

```
planet(venus).
```

```
satellite(miranda).
```

```
orbits(X,sun):- planet(X).
```

```
orbits(X,Y):- satellite(X), planet(Y).
```

Şayet programda kural 'orbits(X,sun):-planet(X).' ise kontrol eder.

```
?- clause(orbits(X,sun),planet(X)).
```

```
X=_0 [X remains uninstantiated]
```

```
yes
```

```
?- clause(planet(venus),true).
```

```
yes
```

```
?- clause(satellite(moon),true).
```

```
no
```

```
?- clause(planet(venus),_).
```

```
yes
```

'clause' Prolog olgularından 'if-kısmı' olmayanları da tanır. Ama iki argümanlı olması gerektiğinden 'true' ikinci argüman olarak yazılır yada true yerine geçen alt çizgi işareti '_' kullanılır.

retract(X): Kullanıcının girmiş olduğu kuralları yada olguları siler.

```
planet(saturn).
```

`satellite(miranda).`

`orbits(X,sun):- planet(X).`

'planet(saturn).' olgusunu veritabanından bu komutla siler.

?- retract(planet(saturn)).

`asserta(X)`: Prolog programına bir olgu ekler. Programda aynı ilişkili isimli bir olgu zaten varsa yeni olguyu en üste ekler. Örnek olarak, veri tabanında ilk olarak bu olgular var.

`satellite(moon).`

`planet(earth).`

`planet(mars).`

Planet hakkında var olan olgulara 'planet(venus).' olgusu eklenir. Bunun etkisi aşağıda görülür.

?- asserta(planet(venus)).

Olgular listelenir :

?- listing(planet).

Cevap:

`satellite(moon).`

`planet(venus).`

`planet(earth).`

`planet(mars).`

`assertz(X)`: Prolog programına bir olguyu ekler. Eğer aynı ilişkili isimi ile zaten bir olgu varsa, o zaman yeni olguyu var olan aynı ismin olguları ve kurallarından sonra ekler.

`fish(shark).`

`fish(salmon).`

`plant(parsley).`

Var olan olguya 'fish(plaice).' eklersek. Prolog komut satırı:

?- assertz(fish(plaice)).

Liste:

```
?- listing.  
    fish(shark).  
    fish(salmon).  
    fish(plaice).  
    plant(parsley).
```

read(X): Kullanıcının yanıtını okumak için kullanılan bir Prolog fonksiyonudur. 'read' sadece bir atomu (sayı, harf, isim yada tek tırnak içine yazılmış ifade) okur. Bu fonksiyon Prolog programının çalışması boyunca etkileşimi sağlar.

```
?- read(X),asserta(planet(X)).
```

Prolog komut satırında kullanıcı bir isim girer (örnek olarak 'venus') ve sonra ismi ilişkisel ismin 'planet' bir argümanı yapar ve bunu bir Prolog olgusu olarak veritabanında oluşturur.

write(X): Bu fonksiyon argümanları ekrana yazar.

```
?- write(cat).  
cat  
?- write(tom),write(cat).  
tomcat  
?- write('Aladdin had a magic lamp.').  
'Aladdin had a magic lamp.'
```

nl: Yeni satır yazar.

```
?- write('M'),nl,write(o),nl,write(o),nl,write(n),nl.
```

Cevap:

```
M  
o  
o  
n
```

call(X): Prolog programında kullandığında bu fonksiyon olgu veya kural olan argümanını yorumlar ve çalışmasını sağlar.

```
if(X,Y,Z):- call(X),  
             call(Y),!.
```

```
if(X,Y,Z):- call(Z).
```

‘call’ şayet fonksiyon X’i gerçekleştirirse, ‘if’ fonksiyonu Y’yi yerine getirir, aksi takdirde Z fonksiyonu yerine getirir.

fail: İki farklı amaç için Prolog programında kullanılır. İlk olarak bir fonksiyona tekrarlanmış çözümler bulmak için kullanılır. Bunu göstermek, küçük bilgi tabanını hesaba katılır.

```
planet(mercury).  
planet(venus).  
planet(X):- orbits(X,sun).  
satellite(miranda).  
orbits(saturn,sun).
```

Gezegenlerin gösterileceğini farz edelim. Bütün gezegenler gösterilene kadar her bir gezegenden sonra Prolog sorgusunu tekrardan uygular. Burası ‘fail’ kullanıldığı yerdir. Bütün gezegenler gösterilene kadar döngüye devam eder.

```
?- planet(X),write(X),nl,fail.
```

```
planet(mercury)  
planet(venus)  
planet(saturn)  
no
```

Son gezegen bulunduktan sonra, başka bir çözüm olmadığından ‘no’ döndürür. ‘fail’in ikinci kullanımı fonksiyon ‘!’ ile birliktedir. Buna ‘cut-fail’ kombinasyonu denir. Malum koşul, uygulandığında Prolog programının yürütmesini durdurmak için kullanılır. ‘fail’ Prolog’da kontrol fonksiyonudur. Diğer kontrol fonksiyonu da ‘!’ dir. Prolog programında bunun örneği aşağıdaki gibi görünür. q(X) şartı sağlandığında programın yürütülmesi durur.

```
p(X):- q(X),!,fail.
```

functor(X,Y,Z): Bu fonksiyon, ilişkili isimi ve Prolog olgusunun bireylerinin sayısını belirtmek kullanılır.

?- functor(planet(venus),X,1).

X=planet

?- functor(larger_than(horse,dog),Y,Z).

Y=larger_than

Z=2



5. UYGULAMA

5.1. Problemin Tanımı

Arıza giderme birçok endüstri alanında üretilen birçok üründe çeşitli nedenlerden dolayı ortaya çıkan aksaklıkların giderilmesi, anormalliklerin düzeltilmesi için gerekli bir işlemdir (Oktay, 1993).

Çok çeşitli ürünlerde çok farklı aksaklıklar ve anormalliklerle karşılaşılabilir. Ürün mekanik bir araç olabileceği gibi bir yazılım programı, veya bu tezde ele alınan bir bilgi tabanlı bir teşhis programı olabilir.

Üzerinde arıza giderme işlemi uygulanacak olan ürünün fiziksel boyutları o ürünün içerdiği teknoloji ve ürünün yapısının karmaşıklığında arıza giderme işleminin önemini yakından etkiler.

Gelişen elektronik ve bilgisayar teknolojilerinin aviyonik alanlardaki etkisi sonucu, uçuş yönetim sistemlerinin tek bir birimden yönetilerek, çok işlevli monitörlerde gösterime sunulması gerçekleşmiştir. Böylece pilot iş yükünün azaltılması ve operasyon etkinliğinin artırılması hedeflenmiştir. Uçuş sırasında uçakta oluşabilecek arızalarda, uçuş ekibi hızlı ve güvenli bir şekilde düşünebilmeli ve talimatları uygulamalıdır (Letourneau ve Famili, 1999). US sayesinde geliştirilebilecek yazılımlar bu hızı artırır. Günümüzde yeni nesil uçak tasarımı ve mevcut uçak modifikasyonlarında bu tür çözümlere gidilmektedir.

Kokpitlerin içinde bulunan aviyonik sistemler gün geçtikçe daha da kompleks hale gelmektedir. Uçuş paneli; motor, yakıt, hidrolik, elektrik ve pneumatik sistemlerin, radyo ve seyrüsefer ekipmanların kontrol göstergeleriyle dolmuştur. Pilot uçak kontrolünü, aynı zamanda da sözü edilen bu sistemlerin kontrolünü sağlamak

zorunda kalmıştır. Bu durum uçuş ekibinin, uçuş süresince sürekli olarak tüm gösterge ve sistemdeki elemanları tek tek izlemek zorunda kalmasına, bunun sonucunda da performansın düşmesine ve güvenli uçuş koşullarının zorlaşmasına neden olmuştur.

Tez konusu olarak ele alınan arıza giderme US kabuğunun problem alanı olarak seçilen anormal durumlar, pilot iş yükünün azaltılması, operasyon etkinliğinin artırılması ve güvenli uçuş koşullarının sağlanmada büyük önem taşır.

5.2. Problemin Alanı

Boeing 737-400 ve 737-500 uçakları, bir çok sistemi ve bunların teknolojilerini bünyesinde barındıran karmaşık bir elektro mekanik, hidrolik, elektronik, ve pnömatik sistemler kümesidir. Bu sistemleri kontrol eden, kokpitlerin içinde bulunan aviyonik sistemler gün geçtikçe daha da kompleks hale gelmektedir. Uçuş ekibinin bu sistemlerde meydana gelebilecek anormal durumları, hem uçuş yönetim sistemleri ile hem de kendi tecrübesiyle en kısa zamanda arızayı teşhis etmelidir. Her bir durumda motorun çalışmakta olduğu uçağın, güvenli uçuşabilmesi için en kısa zamanda arıza giderme işlemini yapmalıdır.

Sivil havacılıkta bütün kazaların %75 ve daha fazlası kararsız insan davranışları sonucunda meydana gelir. Bu insan hataları, daha çok pilotun sınırlarını zorlamasıyla meydana gelmektedir (Onken, 1995). Pilot iş yükünü azaltmaya yardım eden, verim oranı ve uçuşun güvenilirliğini artıran sistemler geliştirilmelidir.

Bu tipte bir uçağın fiyatı 50-200 milyon dolar civarında olup ve uçuş ekibinin arıza giderme işlemini yapamadığı durumlarda oluşabilecek maddi kayıplar büyüktür. Ayrıca maddi kayıpların yanı sıra, uçağta bulunan yolcuların hayatı, problemin önemi konusunda en çarpıcı unsurdur. Uçağın kullanımı yüksek derecede uzmanlık gerektiren bir iştir. Bu bağlamda, programın temel amaçları, uçuş ekibine nesnel bir bakış açısı ve hız kazandırmaktır (Boeing Co., 2001).

5.3. Boeing 737-400/500 için US İhtiyacı

Bir pilot uçuş sırasında ortaya çıkan problemleri çözebilmek için oldukça uzun ve pahalı bir eğitimden geçerler ve kokpitlerin içinde bulunan aviyonik sistemler gün

geçtikçe daha da kompleks hale gelmektedir. Uçuş paneli; motor, yakıt, hidrolik, elektrik ve pnömatik sistemlerin, radyo ve seyrüsefer ekipmanların kontrol göstergeleriyle dolmuştur. Pilot uçak kontrolünü, aynı zamanda da sözü edilen bu sistemlerin kontrolünü sağlamak zorunda kalmıştır. Bu durum uçuş ekibinin, uçuş süresince sürekli olarak tüm gösterge ve sistemdeki elemanları tek tek izlemek zorunda kalmasına, bunun sonucunda da performansın düşmesine ve güvenli uçuş koşullarının zorlaşmasına neden olmuştur.

Havacılık sanayiinde çok önemli bir yeri olan Boeing firmasının ulaştırma alanında sivil havacılıkta önemli bir yeri vardır. Bu şirketin ürettiği ve sattığı bir çok yolcu uçağı ülkemizde de kullanılmaktadır. Bu gün Türk Hava Yollarının toplam 68 tane yolcu uçağı bulunmaktadır. Bunlardan 16 tanesi Boeing 737-400, 2 tanesi Boeing 737-500, 24 tanesi Boeing 737-800, 7 tanesi Airbus 310-304, 9 tanesi RJ-100 ve 3 tanesi RJ-70'dır (www.thy.com.tr).

Boeing 737-400/500 uçak tiplerinde pilotlar anormal durumlarda kontrol listelerini kullanmak zorundadırlar. Bu modellerde bilgisayar ortamında yardımcı bir sistem yoktur (Özçelik, 2001). Bu sistem ihtiyacı düşünülerek, bu tezde, Boeing 737-400/500 uçaklarının anormal durumlar kontrol listesi referans alınarak bir US geliştirildi.

5.4. Boeing 734-400/500 İçin Arıza Giderme İşlemi

Arıza giderme işlemlerini yapan pilotlar üreticinin hazırlamış olduğu anormal durumlardaki kontrol listesini (Non-normal Checklist) kullanmaktadırlar. Bu kullanım kılavuzu, ortaya çıkması muhtemel olan her türlü arıza için en detaylı arıza giderme çözümlerini içerirler. Bir anormallik karşısında bu anormalliğin işaret edebileceği muhtemel tüm arızaları ve bunların giderilmesi için yerine getirilmesi gereken tüm talimatlar bu kullanım kılavuzunda detayları ile verilir. Bu kılavuzda mümkün olan tüm sebeplere göre yapılacak işlemler sıralanmıştır (Boeing Co., 2000).

5.5. Uçuş Mürettebatının Arıza Teşhis Kontrol Listesinin Yapısı

Kontrol listeleri uçuş boyunca yüksek seviyede güvenlikten emin olmak için ve uçuş güvenliği için bir uçağın ayarlarını yapmak için uçuş ekibi tarafından kullanılır

(Turner ve Huntley, 1990).

Buna ek olarak, kontrol listesi, kokpiti işlevsel gereksinimlerini karşılamak için sıralı sistem sağlar ve mürettebat üyeleri arasında uçuş güverte ayarlarının çapraz-kontrolüne yardım eder (Degani ve Weiner, 1990).

Bu bölüm, anormal durumların üstesinden gelmek için uçuş ekibi tarafından kullanılan kontrol listesini içerir. Kontrol listeleri her bir bölümde alfabetik sırayla düzenlenmişlerdir. Tablo 5.1'de görüldüğü gibi 15 ana başlık altında sınıflanmışlardır. Bu 15 ana sistemin bütün sorunları Tablo B.'de gösterilmiştir (Boeing Co., 2000).

Tablo 5.1. Boeing Anormal Durum Kontrol Listesi Ana Başlıkları

0 Unannounced Checklists.
1 Airplane General, Emergency Equipment, Doors, Windows.
2 Air Systems
3 Anti-Ice, Rain
4 Automatic Flight
5 Communications
6 Electrical
7 Engines, APU
8 Fire Protection
9 Flight Controls
10 Flight Instruments, Displays
11 Flight Management, Navigation
12 Fuel
13 Hydraulics
14 Landing Gear
15 Warning Systems

Anormal durumlar kontrol listesi 15 ana bölümden oluşmaktadır. Bu her 15 ana bölümde kendi içinde bölümlere ayrılır. Üçüncü bölümde (3 Anti-Ice, Rain) buzlanmayla ilgili pilotun karşılaşacağı problemler alfabetik sırada Tablo 5.2.'de ve Şekil A.'da orijinal halinde gösterilmiştir. Bunlar motorda, pencerelerde ve

kanatlarda buzlanma gibi anormal bir durumda pilota yapması gereken adımları sırasıyla tanımlar.

Aşağıdaki beş tane anormal durumdan her hangi biri ortaya çıktığında, pilot tecrübelerini de kullanarak anormal durumlar kontrol listesine baş vurur. Şekilde bu beş durum için pilotun yapması gereken işlemler görülmektedir. Pilot, açıklaması yukarıda yapılan anormal durum kontrol listesi el kitabından koşul (condition) kısmında yazan semptomu göre uygun arıza giderme işleminin anlatıldığı sayfaları bulur ve kontrol listesinde yazılmış prosedürleri yerine getirir. Bir birinden alakasız çoklu hata durumlarında, uçuş kokpit ekibi, eylemin en güvenli şekline karar vermek için bir kontrol listesi çoğunun öğelerini birleştirir ve hükmünü uygular (Boeing Co., 2000).

- Engine cowl anti-ice
- Engine cowl valve open
- Pitot static heat malfunctions
- Window overheat
- Wing anti-ice valve open

Tablo 5.2. Anti-Ice, Rain Anormal Durumları Tablosu

ENGINE COWL ANTI-ICE
Condition: An engine COWL ANTI-ICE light illuminated indicates an overtemperature or overpressure condition in the cowl anti-ice duct. Flight conditions permitting: : THRUST LEVER.....RETARD Reduce thrust until the COWL ANTI-ICE light extinguishes. AUTOTHROTTLE (If engaged).....DISENGAGE [Prevents undesired autothrottle activity.]
ENGINE COWL VALVE OPEN
Condition: An engine COWL VALVE OPEN light remaining illuminated bright blue in flight indicates the cowl anti-ice valve position disagrees with the engine anti-ice switch position. Valve Open: If total air temperature is above 10°C, limit thrust on the affected engine to 80% N1 if possible. Valve Closed: Avoid icing conditions.
PITOT STATIC HEAT MALFUNCTIONS
Condition: Pitot static light(s) illuminated indicate related pitot static malfunction(s). Avoid icing conditions. Note: Flight in icing conditions may result in erroneous flight instrument indications.
WINDOW OVERHEAT
Condition: A window OVERHEAT light illuminated indicates an overheat condition has been detected. WINDOW HEAT SWITCH (affected window)..... OFF [Extinguishes the OVERHEAT light and resets the system.] After 2-5 minutes: WINDOW HEAT SWITCH.....ON If the window OVERHEAT light re-illuminates: WINDOW HEAT SWITCH.....OFF Limit airspeed to 250 knots maximum below 10,000 feet. WINDSHIELD AIR CONTROLS.....PULL [Vents conditioned air to the inside of the windshield for defogging.]
WING ANTI-ICE VALVE OPEN
Condition: A WING ANTI-ICE L VALVE OPEN and/or R VALVE OPEN light remaining illuminated bright blue in flight indicates the related wing anti-ice valve position disagrees with the wing anti-ice switch position. Valve Open: If total air temperature is above 10°C or no visible moisture: ISOLATION VALVE SWITCH.....CLOSE PACK SWITCH (affected side).....OFF [Causes operating pack to regulate to high flow in flight with flaps up.] ENGINE BLEED AIR SWITCH (affected side).....OFF Valve Closed: Avoid icing conditions.

Bir şart ifadesi bütün anormal durum procedürler için sağlanır. Kontrol listeleri, hem hatırlatma hem de referans maddeleri kapsarlar. Hatırlatma maddeleri, bellekten yerine getirilmesi gereken kritik adımlardır. Kontrol listesini okuyor iken referans maddeleri, tamamlanması gereken hareketlerdir. Her bir anormal durum kontrol

listesi bölümü için içindekiler tablosunda, bellek yerine getirilmesi gereken kritik öğelerini kapsayan kontrol listelerinin başlıkları koyu tipinde yazılırlar.

Görev için sebep açık olmadığında bilgi, köşeli parantez içinde [] dahil edilir. Bu maddeler, normal yazı tipi karakterleri yazılırlar ve kalın(bold) olmazlar. Anormal durum kontrol listesi anormal durum kontrol listeleri, durum veya durumu düzeltme adımlar ile başlarlar (Boeing Co., 2000).

5.6. Anormal Durum Kontrol Listesi Çalışması

Anormal durum kontrol listeleri durum veya koşulu (condition) düzeltmek işlemleri ile başlarlar. Uçuşun kalanını planlamak için bilgi içermektedir. Özel durumlarda, iniş için uçağı ayarlamak gerektiğinde, görevler, anormal durum kontrol listesinin parçası gibi kabul edilir. Belirlenmiş anormal durumları için uçuş kalıbı, anormal durum manevra bölümünde belirtilir ve konfigürasyon değişikliğinin sırasını gösterir.

Bir çok çalışma, gerekli anormal durum kontrol listelerini belirlemek için yapılır. Fakat bütün olası durumlar için kontrol listeleri geliştirmek mümkün değildir.

Belli çoklu hata durumlarında, uçuş kokpit ekibi, eylemin en güvenli şekline karar vermek için bir kontrol listesi bir çoğunu öğelerini birleştirebilmeli ve/yada hükmünü uygulayabilmelidir (Boeing Co., 2000).

Kaptan duruma değerlendirmeli ve hareketin en güvenilir rotayı belirlemek için kendi yargısını kullanmalıdır. Pilotlar kontrol listelerinin bütün olası durumlar için yaratılmadığını ve doğru kararı yerine getirmeyi planlayamayacağını farkında olmalıdırlar. Bazı durumlarda, kaptanın sağduyusu ile kontrol listelerden sapma gerekli olabilir (Schuttle ve Trujillo, 1996).

5.7. Yeni Nesil Uçuş Yönetim Sistemleri

Gelişen elektronik ve bilgisayar teknolojilerinin aviyonik alanlardaki etkisi sonucu, uçuş yönetim sistemlerinin tek bir birimden yönetilerek, çok işlevli monitörlerde gösterime sunulması gerçekleşmiştir. Böylece pilot iş yükünün azaltılması ve operasyon etkinliğinin artırılması hedeflenmiştir. Günümüzde yeni nesil uçak tasarımı ve uçak helikopter modifikasyonlarında bu tür çözümlere gidilmektedir.

1960'lı yıllarda, uçuş yönetim sistemlerinin karmaşık hale gelmeye başlamasıyla sistem üniteleri arasında bulunan çok sayıdaki bağlantı aviyonik sistemde hem ağırlık hem de ölçü artışına neden olmuştur. Bu problemin anlaşılmasıyla 1960'ların sonunda sayısal, zaman bölmeli (time-division multiplex) veri yolu (data bus) geliştirilmesine başlanmıştır. Sistem üzerindeki çalışmalar 1978'de tamamlanmıştır. Uçakta bulunan aviyonik ekipmanların arasında yapılan bilgi değişiminin tek bir kablo ile değişik zamanlarda gerçekleştirilmesi prensibine dayanmaktadır.

Veri yolu kavramının geliştirilmesi ve buna bağlı olarak bilgi transferinin değişik aviyonik ekipmanlardan gelen birden çok sinyalle ortak bir yoldan gerçekleştirilmesi, hava platformlarında büyük avantaj sağlamıştır. Bu yöntem, her birimden gelen bilginin, bu veri yoluna bağlı tüm birimler tarafından erişilebilmesini, dolayısıyla sistem ağırlığının (kablo ağırlığı ve uzunluğu) azaltılmasını sağlamakta ve tüm birimlerin ortak veritabanı kullanmalarını garantilemektedir. Veri yolu kavramı değişik operasyon ve performans amacıyla yapılan uçak modifikasyonlarında da çok daha esnek sistem tasarımına imkan vermektedir.

Bu sistemle, askeri hava platformlarındaki sayısal veri yolu gereksinimlerini ve veri yolu teknolojisinin gelecekte üretilecek uçaklardaki kullanımını açıklamıştır. Bu standart sadece uçaklara yönelik olarak bilinse de silah sistemleri ve yer sistemlerinde de kullanılmaktadır.

Bu veri yolu yapısında, bir veri yolu yöneticisi (bus controller-BC) vardır, diğer cihazlar ise birer uzak terminal (remote terminal)'dir ve BC'nin komutuna bağlı olarak veri yoluna bilgi yollarlar ya da söylenen bilgiyi alırlar. Bu veri yolu yöneticisi genellikle CDU (Control Display Unit)'dur. CDU, tüm aviyoniklerin tek bir merkezden yönetilmesini, barındırdığı yazılım sayesinde de sistemdeki ekipmanlar arasında bilgi alışverişini kontrol eder. Bu sayede, sisteme yeni bir ekipman eklendiğinde bu yazılımda gerekli değişiklikler yapılarak sistem entegrasyonu sağlanmaktadır.

Hava platformlarının tarihi gelişimi incelendiğinde, kokpit tasarımındaki değişikliklerin uçakların yetenekleriyle doğru orantılı olarak arttığı gözlenebilir. Uçuş sürelerinin saniye ve dakikalarla sınırlı olduğu zamanlarda, kokpit içinde az sayıda cihaz vardı. Uçak performansının gelişen teknolojiyle birlikte artması kıtalar

arası uçuşu sağlamıştır. Böylece seyrüsefer, motor ekipmanları, radyo, hava radarı ve diğer uçuş sistemlerinin kokpitlerde yer alması zorunlu hale gelmiştir.

Kokpitlerin içinde bulunan aviyonik sistemler gün geçtikçe daha da kompleks hale gelmektedir. Uçuş paneli; motor, yakıt, hidrolik, elektrik ve pnömatik sistemlerin, radyo ve seyrüsefer ekipmanların kontrol göstergeleriyle dolmuştur. Pilot uçak kontrolünü, aynı zamanda da sözü edilen bu sistemlerin kontrolünü sağlamak zorunda kalmıştır. Bu durum uçuş ekibinin, uçuş süresince sürekli olarak tüm gösterge ve sistemdeki elemanları tek tek izlemek zorunda kalmasına, bunun sonucunda da performansın düşmesine ve güvenli uçuş koşullarının zorlaşmasına neden olmuştur.

Kokpit içindeki ağır, pahalı ve yer kaplayan, her aviyonik ekipman için kullanılan tek fonksiyonlu ölçüm aletleri ve monitörler günümüzde çok amaçlı göstergelerle (MFD) değiştirilerek pilot iş yükünün azaltılması hedeflenmiştir. Günümüz MFD (Multi-Function Display) teknolojisinde kullanılan renkli, Aktif Matriks Likid Kristal ekrana (AMLCD) sahip, yüksek çözünürlüklü, hem gündüz hem de gece koşullarında rahatlıkla okunabilir parlaklığa sahip camların kullanılması pilotun görme yeteneğini artırarak daha güvenli uçuş sağlamaktadır. MFD'ler bu yeni geliştirilen camın yanında çok gelişmiş elektronik donanım ve yazılıma sahiptir. Bu yazılım sayesinde video processing, computer işlemleri, analog/sayısal çevirimleri, uyarı-ikaz ve moving map fonksiyonlarını gerçekleştirebilmektedir.

Bilgisayar ve elektronik alanında süregelen gelişmelerin hava platformlarına taşınması ile tasarlanan MFD, CDU gibi ekipmanlar ve bunların içinde bulunan yazılımlar, uçuş yönetiminin daha kolay gerçekleştirilmesini, karmaşık uçuş manevralarının daha hızlı, verimli ve güvenli bir biçimde yapılmasını sağlamaktadır.

CDU, data girişine ve datanın monitör edilmesine imkan vermekte, uçaktaki diğer aviyonik ekipmanlarla birlikte çalışarak uçuş yönetiminin (FMS-Flight Management System) gerçekleştirilmesini sağlamaktadır. CDU, genellikle FMS yazılımını içermektedir. Bu yazılım uçaktaki seyrüsefer, yakıt, ağırlık, iletişim yönetimini sağlamaktadır. Pilot gerekli menüyü seçerek data girişini veya gerekli parametrelerin kontrolünü, monitör edilmesini CDU üzerinden yapabilmektedir. CDU, aviyonik ekipmanların ve tüm radyoların tek bir merkezden kontrolünü sağlamaktadır. Ayrıca uçuş planlarının yapılmasına imkan vermektedir. Pilot, uçuş

öncesinde sistem veri tabanını (eski görev bilgileri, sayısal haritaları, havaalanı bilgileri, vs) kullanarak ve varış noktaları gibi bilgileri içeren uçuş planını hazırlayarak CDU'ya aktarabilmekte, görev sırasında bu planın otomatik olarak uygulanmasını sağlayabilmektedir. CDU ayrıca uçuş planının uçuş sırasında kolay ve hızlı bir biçimde değiştirilebilmesine imkan vermektedir. Pilot, mevcut yüklenmiş uçuş planlarını ve varılacak noktaları silebilmekte, yenilerini ekleyebilmekte, veya yerlerini değiştirebilmektedir. CDU, uçuş planı aktif durumdayken bir sonraki varış noktasına gidiş hesabını yaparak önceden dönüş alarmı verebilmekte bu noktaya gitmek için gerekli manevra parametrelerini hesap etmekte, böylece görev daha verimli olarak gerçekleştirilebilmektedir.

Günümüzde endüstri öyle bir noktaya ulaşmıştır ki, hava platformlarındaki gelişmeler yeni uçak tasarımlarından çok, modernizasyon amaçlı yapılan tasarım ve değişikliklerden oluşmaktadır. MFD, CDU, FMS gibi sistemlerin tasarımıyla tüm dünyadaki silahlı kuvvetler, yeni uçaklar almak yerine mevcut eski platformları söz konusu yeni sistemlerle donatarak daha az maliyetle günümüz teknolojisini kullanmayı tercih etmektedirler en yeni modifikasyon çalışmaları, 'glass kokpit' konfigürasyonu olup, kokpit içindeki antika sayılabilecek sensörlerin tamamen kaldırılmasını ve yerlerine MFD ve CDU'larm yerleştirilmesini, böylece uçuşun pilot için çok daha elverişli, güvenilir ve kolay hale getirilmesini sağlama yönündedir.

Kokpit teknolojisinde son 10 yılda görülen gelişmelerin, önümüzdeki 20-25 yıl içinde daha da artarak devam etmesi beklenmektedir. Bu gelişmiş sistemlerin, lazer korumalı kokpitlerden penceresiz kokpitlere kadar uzanabileceği tahmin edilmektedir (Tokdemir, 1999).

5.8. Uçuş Ekibi Faaliyetleri İçin Yeni Boeing Yazılımı

Bu yeni nesil avyonik sistemlerin yazılımlarla destekleyen firmalardan biri olan, Boeing şirketi, uçak performans hesaplamalarını en iyi şekilde kullanan yeni yazılım ürününü haziran 2001'de tanıttı. Pilot iş yükünü azaltmaya yardım eden ve verim oranı ve uçuş çalışmalarının güvenilirliğini geliştirirler.

Yeni Boeing diz üstü bilgisayar aleti, havalanma ve iniş hesaplamaları optimize ederek işlevsel verim oranları artırmak tasarlanılan yazılım uygulamadır. Kullanılan diz üstü bilgisayarda, tipik olarak pilotlar uçaklarında taşıdıkları kağıt kullanım

kavuzunun dijital versiyonlarına yapılan hızlı erişiminde sağlayabilir. Diz üstü bilgisayara yüklenilebilen dijitalleştirilmiş referans kaynakları;

- Uçuş el kitabı (Flight Manual),
- Faaliyetler El Kitabı (Operations Manual),
- Hızlı Referans El Kitabı (Quick Reference Handbook) ve
- Uçuş Ekibi Eğitim El Kitabı (Flight Crew Training Manual).

Yeni yazılım uçuş mürettebatının prosedürlerini düzene koyar. Örnek olarak, el ile yapılan hesaplamalar ve farklı dokümanlardan dipnot bilgisini aramakta ihtiyacı ortadan kaldırmak için havalanma ve iniş için performans hesaplamalarını otomatikleştir. Başka avantajı, diz üstü bilgisayar aleti klasik kağıt kullanım kılavuzlarına karşılaştırılan, tek bir uçak için daha çok belli ve ihtiyaca göre düzenlenilen veriyi kapsadığıdır.

Yüksek doğruluğun düzeyi, herhangi bir hava koşulunda, herhangi bir uçak pistinde, uçuş mürettebatına en uygun verim oranı için roketin taşıdığı yük kapasitesini öğrenmesine imkan verir. Gerçekten, koşulların yerine geçmek için herhangi bir ayarlama, bu yazılım ile hesaplanabilir. Aynı zamanda diz üstü bilgisayar aracının hızı, doğruluğu ve kapasitesi uçak zaman çizelgesi güvenilirliğine arttırma da yardım eder.

Elle yapılmı veriyi geçersiz kılan geçici durum varsa, birisi sayıları yeniden hesaplamaya ihtiyaç duyar. Bazen zaman azalmasından dolayı, ana üst bilimsel uzmanlardan yardım alır. Diz üstü bilgisayar aracı ile, ancak, uçuş mürettebatı veya yerel hareket merkezi gecikmeden o hesaplamalar yapabilirler (The Boeing Co., 2001).

Sunum sistem özellikleri

- Çoklu altyapı
- Uzman/Çırak
- Esnek Gösterge Teknolojisi
- Teşhis US'i ve Boeing Dijital Kablo Veri Sistemi
- İlişkisel tablolardan veri alma yeteneği
- Bakım etkinliğinin günlüğünü denetleme

- Yer imi yeteneği
- Talebe bağlı baskı
- Geriye ve ileriye doğru dolaşım

5.9. Modelin Özellikleri

Tezin ikinci aşamasında geliştirilen arıza giderme US programı kullanılarak gerçek hayattan bir uygulama ortaya çıkarmak ve böylelikle bu geliştirilen programın bir çok ticari uçakta uçuş mürettebatı için kullanılabileceğini kanıtlamaktadır. Böylece pilot iş yükünün azaltılması ve operasyon etkinliğinin artırılması hedeflenmiştir. Günümüzde yeni nesil uçak tasarımı ve mevcut uçak modifikasyonlarında bu tür çözümlere gidilmektedir.

US kurma araçlarından yazılım dili olarak, önerilen sistemin bilgi tabanlı olması gerektiğinden bir kural dili olan Prolog seçilmiştir. Bu amaca ALS Prolog US aracı kullanılarak hazırlanan bir programla ulaşılmıştır.

Bu uygulamada, bilgi sunum türlerinden olan çerçeve tabanlı sunumdan yararlanılmıştır. Bilindiği gibi bu sistemde bilgiler çerçeveler ile gösterilir. Çerçeveler; veri, olgu, kural ve hipotez gruplarından oluşan bir kalıp çeşididir. Bilgi, slotlar olarak adlandırılan özellik tanımlarının setlerini birleştirebilen çerçeveler adı verilen veri yapılarında temsil edilir.

Uygulamada kullanılan US'nin oluşturulması sırasında çerçeve tabanlı bir bilgi gösteriminden yararlanmaya karara verilmesinin ana nedeni, çerçeve tabanlı sistemler türetme tabanlı ve sınıflandırma tabanlı çıkarımda güçlüdürler. Çünkü çerçeveler arasında cinsine göre sınıflandırılmış ilişkiler, türetme tarafından çoklu çerçeveler arasında bilgiye kolayca erişilmesini sağlarlar.

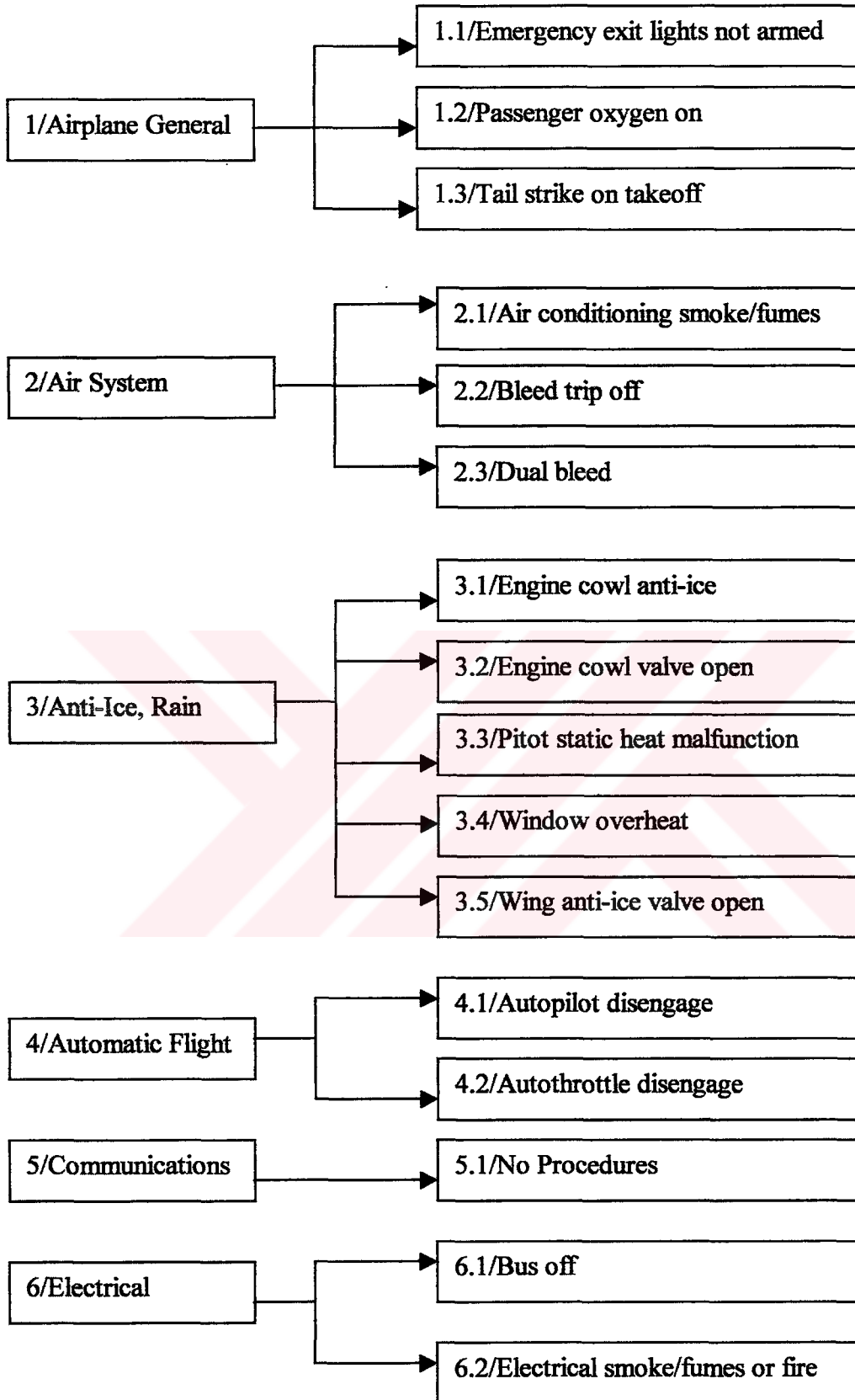
5.10. Modelin yapısı

Program içerisinde anormal durumlar kontrol listesinde bulunan 15 tane bölümden 6 tanesi kullanılmıştır. Bunlar aşağıda listelenmiştir. Bu 6 tane bölümden oluşmuş US hiyerarşik yapısı Şekil 5.1.' de görülmektedir.

- 1) Airplane General, Emergency Equipment, Doors, Windows.

- 2) Air Systems
- 3) Anti-Ice, Rain
- 4) Automatic Flight
- 5) Communications
- 6) Electrical





Şekil 5.1 Uçuş Ekibi İçin Arıza Teşhis Uzman Sistemi

5.11. Programın Çalışması ve Örnek Sorgular

Program içerisinde anormal durumlar kontrol listesinde bulunan 15 tane bölümden 6 tanesi kullanılmıştır. Çalışmanın bu bölümünde aşağıda görülen 6 bölüm için sorgulamalar örnek olarak alınmıştır. Tablo B.'de, uçuş ekibi için karşılaşılan bütün anormal durumlar gösterilmiştir.

Programın çalışması

Program iki şekilde çalıştırılır. Birincisi, EK C'deki disketten, PROLOG klasörü içinde "alspro.exe" dosyası çalıştırılarak "ALS-Prolog Personal Version 1.00" programına girilir. Daha sonra "consult" komutuyla "fcas.ari" dosyası yüklenir.

Prolog komutları ? işaretinden sonra yazılır ve sonuna kesinlikle "." (nokta) koymak gerekir.

?- consult(fcass.ari).

İkincisi, diskette bulunan PROLOG klasörü Windos'da 'C:>' altına kopyalanır. DOS komut istemine geçilerek, aşağıdaki komutla Prolog klasörü altına geçilir ('C:>', disk C: üzerinde çalışıldığı zaman, varsayılan DOS komut istemidir).

C:>cd Prolog

Bu klasör altında yüklemek istediğimiz dosyanın ismi aşağıdaki gibi yazılır.

C:\Prolog>alspro filename

C:\Prolog>alspro fcass.ari

Kaynak kod üzerinde düzenleme "x filename" ve "edit <filename>" komutuyla yapılır.

C:\Prolog >x filename

C:\Prolog >edit filename

C:\Prolog >x fcass.ari

C:\Prolog >edit fcass.ari

Prolog komut istemine geçilir. İlk aşmada program "go" komutu ile başlar.

?- go.

Programdan “halt.” komutuyla çıkılır.

?- halt.

Sorgulama 1:

?-consult(fcas.ari).

?- go.

1/Airplane General

2/Air System

3/Anti-Ice, Rain

4/Automatic Flight

5/Communications

6/Electrical

0/Exit

Hangi arıza var? Numarasini buraya yaziniz ->> 1.

1/Emergency exit lights not armed

2/Passenger oxygen on

3/Tail strike on takeoff

0/Exit

Hangi arıza var? Numarasini buraya yaziniz ->>1.

Durum: The NOT ARMED light illuminated indicates the EMER EXIT LIGHTS switch is not in the ARMED position.

[sorum,1.1]

[cozum,If the EMER EXIT LIGHTS switch is ON, individual emergency exit light batteries will provide a minimum of 10 minutes of lighting.

-If the EMER EXIT LIGHTS switch is OFF, emergency lighting is not available.]

Sorgulama 2:

?- go.

- 1/Airplane General
- 2/Air System
- 3/Anti-Ice, Rain
- 4/Automatic Flight
- 5/Communications
- 6/Electrical
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->> 1.

- 1/Emergency exit lights not armed
- 2/Passenger oxygen on
- 3/Tail strike on takeoff
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->>2.

Durum: The PASS OXY ON light illuminated indicates the passenger oxygen system is activated.

[sorun,1.2]

[cozum,No procedure]

Sorgulama 3:

?- go.

- 1/Airplane General
- 2/Air System
- 3/Anti-Ice, Rain
- 4/Automatic Flight
- 5/Communications
- 6/Electrical
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->> 3.

- 1/Engine cowl anti-ice

- 2/Engine cowl valve open
- 3/Pitot static heat malfunction
- 4/Window overheat
- 5/Wing anti-ice valve open
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->>1.

Durum: An engine cowl anti ice light illuminated indicates an overtemperature or over pressure condition in the cowl anti-ice duct

[sorun,3.1]

[cozum,Flight conditions permitting:

- THRUST LEVER.....RETAND
- Reduce thrust until the COWL ANTI-ICE light extinguishes.
- AUTOTHROTTLE(if engaged).....DISENGAGE]

Sorgulama 3:

?- go.

- 1/Airplane General
- 2/Air System
- 3/Anti-Ice, Rain
- 4/Automatic Flight
- 5/Communications
- 6/Electrical
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->> 3.

- 1/Engine cowl anti-ice
- 2/Engine cowl valve open
- 3/Pitot static heat malfunction
- 4/Window overheat
- 5/Wing anti-ice valve open
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->>2.

Durum: An engine cowl light remaining illuminated bright blue in flight indicates the cowl anti_ice valve position disagrees with the engine anti ice switch position.

[sorun,3.2]

[cozum,Valve Open: If total air temperature is above 10C, limit thrust on the -affected engine to 80% N1 if possible.
-Valve Closed: Avoid icing conditions.]

Sorgulama 4:

?- go.

- 1/Airplane General
- 2/Air System
- 3/Anti-Ice, Rain
- 4/Automatic Flight
- 5/Communications
- 6/Electrical
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->> 3.

- 1/Engine cowl anti-ice
- 2/Engine cowl valve open
- 3/Pitot static heat malfunction
- 4/Window overheat
- 5/Wing anti-ice valve open
- 0/Exit

Hangi ariza var? Numarasini buraya yaziniz ->>4.

Durum: A window OVERHEAT light illuminated indicates an overheat condition has been detected

[sorun,3.4]

[cozum,WINDOW HEAT SWITCH(affected window).....OFF
-After 2-5 minutes:

-WINDOW HEAT SWITCH.....ON.]



6. SONUÇLAR VE ÖNERİLER

Bilindiği gibi elektronik ve bilgisayar teknolojisini gelişmesiyle uçaklarda aviyonik sistemler daha da karmaşıklaşmaktadır.

Bu çalışmada önerilen yaklaşımda, uçuş ekibinin görev yönetim faaliyetlerinin temelini oluşturan anormal durumlarda uygulanacak işlemlerin bir YZ teknolojisi olan US tarafından desteklenebileceği savunulmaktadır.

Önerilen US'nin kurulmasında ilk aşamada Türk Hava Yollarından bir pilotla görüşülmüş ve bu sisteme ihtiyacı olan uçaklar belirlenmiştir. Ayrıca anormal durumlarda uçuş ekibinin kullandığı sistem incelenmiştir.

Geliştirilen US, kullanıcının sorulara verdiği yanıtlara bağlı olarak anormal durumun hangi bölümde olduğunu teşhis eder ve arızayı ortadan kaldıracak bir dizi işlemi yapması için talimatlara kolayca erişmesini sağlar.

YZ ve US teknolojilerinin dünyadaki gelişimine paralel olarak ülkemizde de daha sık kullanılabilir hale gelmesi için öncelikle bu teknolojinin tanınması sağlanmalıdır. Bu teknolojilere yapılacak yatırımların geri dönüş oranları uzun vadeli değerlendirmelerle reel olarak hesaplanmalı ve bu sayede yapılacak yatırımların ne denli yüksek olduğu net bir şekilde görülmektedir. Bu çalışma kapsamında yapılan araştırmalar, yazılım sektöründeki hızlı gelişim ve veri yolu kavramının geliştirilmesi ve buna bağlı olarak bilgi transferinin değişik aviyonik ekipmanlardan gelen birden çok sinyalle ortak bir yoldan gerçekleştirilmesi, hava platformlarında büyük avantaj sağlamıştır. Bu yöntem, her birimden gelen bilginin, bu veri yoluna bağlı tüm birimler tarafından erişilebilmesini, dolayısıyla sistem ağırlığının (kablo ağırlığı ve uzunluğu) azaltılmasını sağlamakta ve tüm birimlerin ortak veritabanı kullanmalarını garantilemektedir. Veri yolu kavramı değişik operasyon ve performans amacıyla yapılan uçak modifikasyonlarında da çok daha esnek sistem tasarımına imkan

vermektedir. Bu veri yolu kavramıyla, veritabanları sayesinde bilgiler yazılımlara kolayca aktırılabilir ve uçuş kontrol yazılımları geliştirilebilir.



KAYNAKLAR

- Allahverdi, N.**,1998. Metods of Artificial Intelligence and Its Applications in Agriculture, *In Proc. of the 2nd Symposium on Computer Applications in Agriculture*, Konya, Turkey, September 28-30, 123-131.
- Allahverdi, N. M., Yaldız S. ve Ünüvar A.**, 1995. Endüstride US Uygulamaları, 2. Endüstriyel Otomasyon '95 Sempozyumu Bildiriler Kitabı, 30-31 Mart, İstanbul, s.75-86.
- ALS Inc.**, 1986. ALS Prolog Technical Reference Manual Personal Version 1.0, University Syracuse, New York, USA.
- Davrak, Ö.**, 1997. Jeppesen Deyimler Elkitabı, Türk Hava Yolları Uçuş Eğitim ve Standartları Başkanlığı, İstanbul.
- Degani, A. and Wiener, E. L.**, 1990. Human Factors of Flight-deck Checklists-The Normal Checklist, NASA CR-177549.
- Delgado, R.**, 1996. An Expert System for Farm Planning, *Proceedings of 6th Intern. Conf. on Computers in Agriculture*, Mexico, pp.497-502.
- Feigenbaum, E. A.**,1992. Expert Systems: Principles and Practice, *Knowledge Systems Laboratory Technical Report, KSL-91-79*, Stanford University.
- Feigenbaum, E. A.**,1981. Knowledge Engineering for the 1980s, *Computing Science Dept.*, Stanford University.
- Hakima, A.S.**, 1996. Artificial Intelligence Techniques in Diagnosis: A Review of Approaches, Applications, and Issues, *Institute for Information Technology National Research Council of Canada*, NRC 37142, ERA-381.
- Hall, C.**,1996. The Intelligent Software Development Tools Market, Part 2, *Intelligent Software Strategies*, V.12, No:3, pp. 1-16.
- Karp, P. D.**, 1993. The Design Space of Frame Knowledge Representation Systems,

Artificial Intelligence Center SRI International, Menlo Park, CA, USA, SRI AI Center Technical Note #520.

Kocabaş, Ş., 2001. Artificial Intelligence I Ders Notları, İ.T.Ü Uçak-Uzay Fakültesi, İstanbul.

Kocabaş, Ş., 1986. Edinburg Syntax Prolog Tutor Reference Manual, *Arafah Systems LTD.*, London, UK.

Kökyıldız, D., 1999. Uçak Bakım Onarımı Üzerine Expert Sistem Hazırlanması, *Lisans Tezi*, İ.T.Ü. Uçak ve Uzay Bilimleri Fakültesi, İstanbul.

Letourneau, S. and Famili, F., 1999. Data mining to predict aircraft component replacement, *IEEE intelligent systems & their applications*, 14, 59-66.

Lucas, P., 1997. Symbolic Diagnosis and its Formalisation, *The Knowledge Engineering Review*, 12(2), pp.109-146.

McClure, J.C., Calvin, D.D. and Esh V., 1996. Pest Management Choices: Integrating Pesticide Toxicity Data with a Crop Management Expert System, *Proceedings of 6th Intern. Conf. on Computers in Agriculture*, Mexico, 520-527.

Oktay, O., 1993. Küçük boy telefon santrallerinin bakım ve onarımı için bir US tasarımı, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

Onken, R., 1995. Functional Development and Field Test of CASSY - A Knowledge Based Cockpit Assistant System, *AGARD*, Madrid, Spain, November 6-7, 4, 1-22.

Özçelik, İ., 2001. THY, Kişisel görüşme.

Parmar R.S. et al., 1997. Estimation of Aflatoxin Contamination in Preharvest Peanuts Using Neural Networks, *Transactions of the ASAE*, Vol.40(3), pp.809-813.

Parsaye, K. and Chignell, M., 1988. Expert Systems for Experts, pp. 29-60, John Wiley & Sons, New York, USA.

Patterson, D. W., 1990. Introduction to Artificial Intelligence and Expert systems, pp. 326-356, Prentice Hall International.

Regers W. et al., 1984. Computer Aided Medical Diagnosis; Literature Review, in Proc. 1.Conf. on Artificial Intelligence Applications-IEEE Computer Society, pp.178-186,

Rich, E. and Knight, K., 1991. Artificial Intelligence, McGraw-Hill, New York, pp. 105-128.

Schuttle, P.C. and Trujillo, A.C., 1996. Flight Crew Task Management in Non-Normal Situations, *40th Annual Meeting of the Human Factors and*

Ergonomics Society, Philadelphia, Pennsylvania, USA, September 2-6, pp. 6.

Vidin, Ö., 1999. Uzman Sistemler ve İnsan Kaynakları Yönetiminde Bir Uygulama, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

Tokdemir, G., 1999. Yeni Nesil Uçuş Yönetim Sistemleri, *Aselsan*, 5.

Topçu, Y.İ., 1995. Ulaştırma Yapılarında Uygun Su Geçişlerinin Önerilmesinde Uzman Sistem Yaklaşımı, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

Topçuoğlu, A., 2001. Yapay zeka, *Bilim ve Teknik*, 409, 38-47.

Turner, J. W. and Huntley, M. S., 1991. The Use and Design of Flightcrew Checklists and Manuals. DOT/FAA/AM-91/7, FAA.

Ünsaçar F., Allahverdi N. and Halkacı S., 1998. Knowledge Base System Rule Design for Fault Diagnosing in CNC Machine Tools, Accepted by Org. Committee of 2nd International Symposium on Intelligent Manufacturing Systems (IMS'98).

Yaldız, S., Allahverdi N. and Ögüt, Ö., 1998. Computerized Education And Application Fields In Professional And Technical Educating, Selçuk Üniversitesi Teknik Bilimler Meslek Yüksekokulu.

Waterman, D. A. A., 1985. Guide to Expert Systems, Boston, Addison-Wesley.

The Boeing Co., 2000. BOEING 737 Quick Reference HandBook, *The Boeing Company*.

The Boeing Co., 2001. Boeing Software Helps Streamline Flight Crew Operations June 7, <http://www.boeing.com/news/releases/>

İnternet Kaynakları

<http://www.boeing.com/commercial/prices/index.html>

<http://www.thy.com.tr>

EKLER

EK A. Örnek Arıza Giderme Kullanım Klavuzu



Non-Normal Checklists	Chapter NNC
Anti-Ice, Rain	Section 3

ENGINE COWL ANTI-ICE	NNC.3.1
ENGINE COWL VALVE OPEN	NNC.3.1
PITOT STATIC HEAT MALFUNCTIONS	NNC.3.1
WINDOW OVERHEAT	NNC.3.2
WING ANTI-ICE VALVE OPEN	NNC.3.2

ENGINE COWL ANTI-ICE

Condition: An engine COWL ANTI-ICE light illuminated indicates an overtemperature or overpressure condition in the cowl anti-ice duct.

Flight conditions permitting:

THRUST LEVER RETARD

Reduce thrust until the COWL ANTI-ICE light extinguishes.

AUTOTHROTTLE (If engaged) DISENGAGE

[Prevents undesired autothrottle activity.]

ENGINE COWL VALVE OPEN

Condition: An engine COWL VALVE OPEN light remaining illuminated bright blue in flight indicates the cowl anti-ice valve position disagrees with the engine anti-ice switch position.

Valve Open:

If total air temperature is above 10°C, limit thrust on the affected engine to 80% N1 if possible.

Valve Closed:

Avoid icing conditions.

PITOT STATIC HEAT MALFUNCTIONS

Condition: Pitot static light(s) illuminated indicate related pitot static malfunction(s).

Avoid icing conditions.

Note: Flight in icing conditions may result in erroneous flight instrument indications.

WINDOW OVERHEAT

Condition: A window OVERHEAT light illuminated indicates an overheat condition has been detected.

WINDOW HEAT SWITCH (affected window) OFF

[Extinguishes the OVERHEAT light and resets the system.]

After 2-5 minutes:

WINDOW HEAT SWITCH ON

If the window OVERHEAT light re-illuminates:

WINDOW HEAT SWITCH OFF

Limit airspeed to 250 knots maximum below 10,000 feet.

WINDSHIELD AIR CONTROLS PULL

[Vents conditioned air to the inside of the windshield for defogging.]

WING ANTI-ICE VALVE OPEN

Condition: A WING ANTI-ICE L VALVE OPEN and/or R VALVE OPEN light remaining illuminated bright blue in flight indicates the related wing anti-ice valve position disagrees with the wing anti-ice switch position.

Valve Open:

If total air temperature is above 10°C or no visible moisture:

ISOLATION VALVE SWITCH CLOSE

PACK SWITCH (affected side) OFF

[Causes operating pack to regulate to high flow in flight with flaps up.]

ENGINE BLEED AIR SWITCH (affected side) OFF

Valve Closed:

Avoid icing conditions.

EK B. Anormal Durumlar Kontrol Listesi

0-UNANNUNCIATED CHECKLISTS
ABORTED ENGINE START AIR CONDITIONING SMOKE/FUMES AIRSPEED UNRELIABLE CONFIGURATION WARNING DICHTING ELECTRICAL SMOKE/FUMES OR FIRE EMERGENCY DESCENT INFLIGHT ENGINE START JAMMED OR RESTRICTED ELEVATOR OR AILERON JAMMED STABILIZER LOSS OF TRUST ON BOTH ENGINES PASSENGER EVACUATION RAPID DEPRESSURIZATION SMOKE/FUMES REMOVAL TAILSTRIKE ON TAKEOFF UNCOMMANDED RUDDER UNCOMMANDED YAW OR ROLL UNSCHEDULED PRESSURIZATION CHANGE VOLCANIC ASH WINDOW DAMAGE
1-AIRPLANE GENERAL, EMERGENCY EQUIPMENT, DOORS, WINDOWS
DOOR ANNUNCIATOR EMERGENCY EXIT LIGHTS NOT ARMED PASSENGER OXYGEN ON TAILSTRIKE ON TAKEOFF WINDOW DAMAGE
2-AIR SYSTEM
AIR CONDITIONING SMOKE/FUMES AUTO FAIL BLEED TRIP OFF DUAL BLEED DUCT OVERHEAT EMERGENCY DESCENT EQUIPMENT COOLING OFF OFF SCHEDULE DESCENT PACK RAPID DEPRESSURIZATION UNSCHEDULED PRESSURIZATION CHANGE WING-BODY OVERHEAT ZONE TEMP

3-ANTI-ICE, RAIN
ENGINE COWL ANTI-ICE ENGINE COWL VALVE OPEN PITOT STATIC HEAT MALFUNCTIONS WINDOW OVERHEAT WING ANTI-ICE VALVE OPEN
4-AUTOMATIC FLIGHT
AUTOPILOT DISENGAGE AUTOTHROTTLE DISENGAGE
5-COMMUNICATIONS
NO PROCEDURES
6-ELECTRICAL
BUS OFF ELECTRICAL SMOKE/FUMES OR FIRE GENERATOR DRIVE LOW OIL PRESSURE OR HIGH OIL TEMPERATURE LOSS OF BOTH ENGINE DRIVEN GENERATORS STANDBY POWER OFF TRANSFER BUS OFF
7-ENGINE
ABORTED ENGINE START APU DET INOP APU FIRE APU HIGH OIL TEMPERATURE/FAULT APU LOW OIL PRESSURE APU OVERSPEED ENGINE FAILURE/SHUTDOWN ENGINE FIRE, SEVERE DAMAGE OR SEPARATOR ENGINE FIRE/OVERHEAT DETECTOR FAULT ENGINE HIGH OIL TEMPERATURE ENGINE LIMIT/SURGE/STALL ENGINE LOW OIL PRESSURE ENGINE OIL FILTER BYPASS ENGINE OVERHEAT ENGINE OVER TEMPERATURE DURING TAKE OFF OR INFLIGHT ENGINE TAIL PIPE FIRE HIGH ENGINE VIBRATOR INFLIGHT ENGINE START LOSS OF THRUST ON BOTH ENGINES LOW IDLE POWER MANAGEMENT CONTROL (PMC) INOPERATIVE REVERSER REVERSER UNLOCKED (IN FLIGHT) START VALVE OPEN

8-FIRE PROTECTION
AIR CONDITIONING SMOKE/FUMES APU DET INOP APU FIRE CARGO FIRE CARGO FIRE DETECTOR FAULT ELECTRICAL SMOKE/FUMES OR FIRE ENGINE FIRE/OVERHEAT DETECTOR FAULT ENGINE FIRE, SEVERE DAMADE OR SEPARATION ENGINE OVERHEAT ENGINE TAILPIPE FIRE SMOKE/FUMES REMOVAL WHEEL WELL FIRE
9-FLIGHT CONTROLS
ALL FLAPS UP LANDING ALTERNATE FLAPS OPERATION ASYMMETRICAL OR NOLEADING EDGE DEVICES ASYMMETRICAL TRAILING EDGE FLAPS AUTO SLAT FAIL FEEL DIFFERENTIAL PRESSURE FLAP LOAD RELIEF FLIGHT CONTROL LOW PRESSURE JAMMED OR RESTRICTED ELEVATOR OR AILATOR OR AILERON JAMMED STABILIZER LEADING EDGE FLAPS TRANSIT MACH TRIM FAIL RUNAWAY STABILIZER SPEED BRAKE SPEED BRAKE DO NOT ARM SPEED TRIM FAIL STABILIZER OUT OF TRIM TRAILING EDGE FLAPS UP LANDING UNCOMMANDED YAW OR ROLL YAW DAMPER
10-FLIGHT INSTRUMENTS
AIRSPEED UNRELIABLE FLIGHT RECORDER OFF
11-FLIGHT MANAGEMENT
FMC/CDU ALERTING MESSAGE FMC FAIL INSTRUMENT SWITCH IRS DC FAIL IRS FAULT IRS ON DC SG FAIL

12-FUEL
CROSSFEED SELECTOR INOPERATIVE FUEL FILTER BYPASS FUEL PUMP LOW PRESSURE FUEL QUANTITY INDICATOR INOPERATIVE FUEL TEMP LOW INADVERTENT TRANSFER OF FUEL INTO CENTER TANK MINIMUM FUEL OPERATION
13-HYDRAULICS
HYDRAULIC PUMP LOW PRESSURE HYDRAULIC PUMP OVER HEAT LOSS OF SYSTEM A LOSS OF SYSTEM B MANUAL REVERSION STANDBY HYDRAULIC LOW PRESSURE STANDBY HYDRAULIC LOW QUANTITY
14-LANDING GEAR
ANTI-SKID INOPERATIVE AUTO BRAKE DISARM BRAKE PRESSURE INDICATOR ZERO PSI GEAR LEVER WILL NOT MOVE UP AFTER TAKE OFF LANDING GEAR LEVER JAMMED IN THE UP POSITION MANUAL GEAR EXTENSION PARTIAL OR GEAR UP LANDING WHEEL WELL FIRE
15-WARNING SYSTEM
ALTITUDE ALERT CONFIGURATION WARNING GROUND PROXIMITY ALERT OVERSPEED PSEU

ÖZGEÇMİŞ

Sedef ÖZÇELİK 25 Eylül 1976 tarihinde Ankara'da doğdu. İlk öğrenimini Bandırma'da, Orta öğrenimini Eskişehir'de tamamlayarak 1993 yılında İstanbul Bahçelievler Lisesi'nden iyi dereceyle mezun oldu. Aynı yıl İTÜ Uçak Uzak Fakültesi Uzak Mühendisliğine girdi. 1998 yılında iyi dereceyle mezun oldu.

Mezun olduktan sonra 1998-2000 Login Bilgisayar Programları AŞ'de Program Geliştirici olarak çalıştı. 2000 yılında Fen Bilimleri Enstitüsü Savunma Teknolojileri Anabilim Dalında Strateji Geliştirme Teknolojileri master programına başladı.

İyi derecede İngilizce bilen ÖZÇELİK C, C++, Java, Visual Basic, SQL ve Prolog dillerini ile Ofis paketlerini kullanabilir.