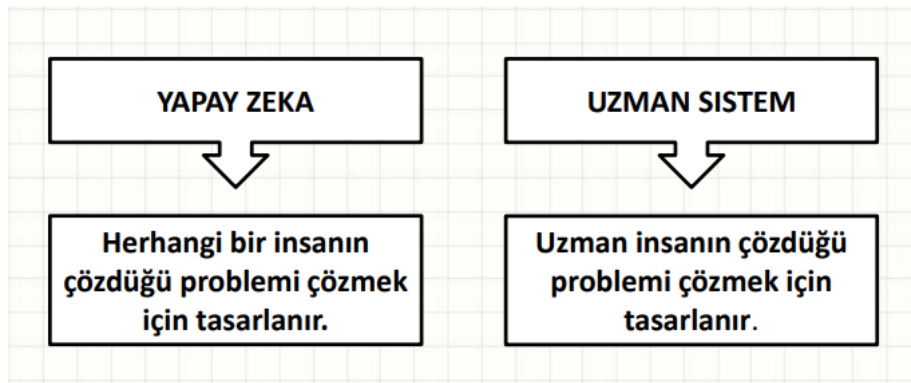


V.BÖLÜM : UZMAN SİSTEMLER

Bir problemi o problemin uzmanlarının çözdüğü gibi çözebilen bilgisayar programları geliştiren teknolojidir. Uzmanlar problemleri çözerken bilgilerini ve deneyimlerini kullanırlar. Bu bilgi ve deneyimlerin bilgisayar tarafından anlaşılabilir olması ve bilgisayarda saklanması gerekmektedir. Bilgi tabanında saklanan bu bilgileri kullanarak insan karar verme sürecine benzer bir süreç ile problemlere çözümler üretirler.

Yapay Zeka, herhangi bir insanın çözdüğü problemi çözmek için tasarlanır. Uzman Sistem ise uzman bir sistemin çözdüğü bir problemi çözmek için tasarlanır. Yani Uzman Sistem, yapay zeka kuramlarının insanı taklit etmesinden farklı olarak sadece UZMANI taklit eder (Şekil 5.1).



Şekil 5.1: Yapay Zeka ve Uzman Sistem.

Uzman sistemler (expert systems), bilgi tabanlı sistemler olup, problemleri daha geniş bir çerçevede inceleyip, çözümünde insan zekasını taklit etmeyi hedefleyen yapay zekanın bir uygulama alanıdır. Bu taklit içerisinde algoritma ve çıkarım mekanizmaları etkileşimde bulunarak işlemektedir. Bir başka tanım ifadesi ise, bir uzmandan alınan bilgilere dayanarak oluşturulan, karmaşık problemleri çözmek için olayları ve deneyimleri kullanan etkileşimli bilgisayar destekli karar aracıdır.

Uzman Sistem, algoritma yerine daha çok bilgi kullanır:

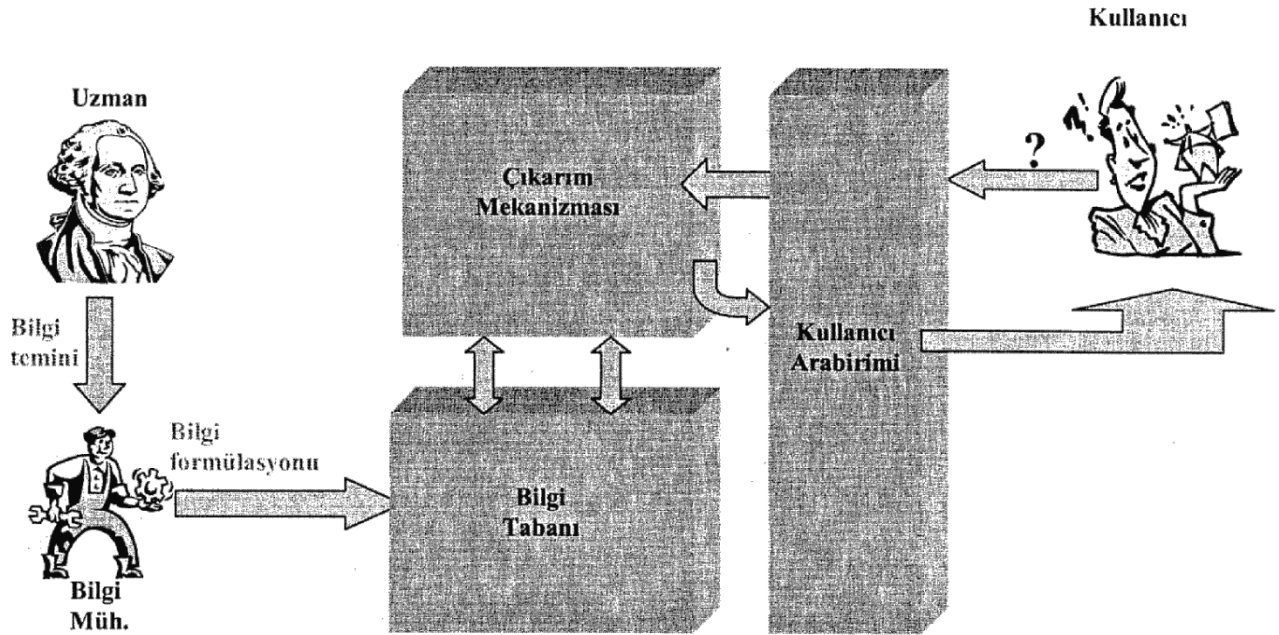
Geleneksel Programlar	→	Algoritma + Veritabanı
Uzman Sistemler	→	Çıkarım Mekanizması + Veritabanı
Çıkarım mekanizması tecrübeye bağlı olarak çıkarılmış kurallardır.		

Bu zeki sistemler bir alana ilişkin problemi aynen bir uzman insan gibi düşünerek çözmeye ve çözdüğü problemi öğrenerek bilgi deposunda depolayıp daha önceden öğrendiklerinin bir sonucu olan bilgilerinde yapısal bir değişikliğe gitmeyerek, uzman sistemin yapısında bir değişime gereksinim ihtiyacı duyulmamasını sağlamaktadır. Uzman sistemin oluşturulmasında alan uzmanından bilgi mühendisinin gerekli bilgileri alması ve bunları bilgi tabanında depolanacak bir şekilde çevirmesi gerekir.

Uzman sistemler kesin ve net algoritmalar yerine deneyime dayalı çıkarım yöntemleri kullanırlar bu nedenden ötürü uzman sistemlerin tasarımı karışık ve zaman alan bir işlemdir.

5.1. Uzman Sistemin Bileşenleri

Bir uzman sistemin 4 temel bileşeni ve çalışma ilkesi Şekil 5.1 'de gösterildiği gibidir:



Şekil 5.1: Uzman sistemin elemanları ve bilgi akışı

a) Bilginin temin edilmesi: Uzman sistemin uzmanlık alanı ile ilgili bilgilerin toplanması, derlenmesi ve bilgisayarın anlayacağı şekle dönüştürülmesi çalışmalarını kapsar.

Uzman sistemin geliştirilmesinde, uzman temel unsurdur. **Uzman**, özel bilgisi, yargısı, deneyimi, metoduyla birlikte, bu yetenekleri

sorunlara uygulayabilen ve tavsiyede bulunabilen nitelikteki kişi(ler)dir. Uzmanın görevi bilgi sisteminin yapacağı görevleri, nasıl yapacağını sisteme sunmaktır. Uzman, aynı zamanda hangi gerçeklerin önemli olduğunu ve bu gerçekler arasındaki ilişkilerin anlamını bilen kişi(ler)dir.

b) Bilgi tabanı: Uzman sistemin uzmanlık alanı ile ilgili toplanan bilgilerin saklandığı yerdir. Bilgiler genellikle kurallar (EĞER... ise O ZAMAN... şeklinde), bilgi çatıları (çerçeveleri), bilgi sınıfları ve prosedürlerden oluşur. Bu bilgiler ilgili uzmanlık alanı hakkında uzmanların bildiği ve belirlediği gerçeklere dayanmaktadır.

Bilgi mühendisi, uzmana problem alanlarını yapılandırmak için yardım eder. Bunu uzman kişi(ler)in cevaplarını yorumlayarak, bütünleştirerek, benzetmeler yaparak, ters örnekler vererek, bilginin formülasyonunu gerçekleştirir. Bu kişi aynı zamanda (çoğunlukla) sistemi tasarlayan kişidir. Uzman sistem yapımında bilgi mühendislerinin eksikliği (azlığı) önemli bir problemdir. Uzman sistemleri dizayn edenler bu güçlüğü yenmek için tasarım araçları kullanarak bilgi mühendisine olan ihtiyacı azaltmaktadırlar.

c) Çıkarım mekanizması: Bilgi tabanında bulunan bilgileri arayan, filtreleyen, yorumlayan ve sonuçlar çıkaran yani; çözüm üreten bir mekanizmadır. Genel olarak iki türlü çıkarım vardır.

i) İleri doğru zincirleme: Bu durumda, ilgili problem hakkındaki gerçeklerden hareket edilerek sonuca gidilir.

Çıkarım ünitesi, problemin en başından başlayarak (*IF cümlesinden*) sonuç kısmına (*THEN...*) ulaşmasıdır. Bu yöntem Tümevarım mantığı ile çalışır. Bütün kuralların şartı sağlayıp sağlamadığı gözönünde tutularak sonuca ulaşılır. Eğer şartları sağlıyor ise "Then" kısmında yer alan yargı cümlesi doğrudur. Bu cümle şartlara göre elde edilen sonuçtur.

ii) *Geri doğru zincirleme*: Bu durumda ise bir sonuç ele alınarak geriye doğru, o sonucu destekleyen gerçekler var mıdır? Sorusunun cevabı aranır.

Çıkarım ünitesi; problemi çözerken kuralın en sonu olan sonuç (*THEN...*) cümlesi ile başlar ve şart (*IF...*) cümleleri tatbik edilerek çözüm bulunur. Yani bu tür zincirleme Tümdengelim ilkesini temel olarak alır ve sonuç kısmını sağlayacak bütün kuralları tek tek inceler.

d) Kullanıcı ara birimi: Uzman sistemi kullanan kişiler ile uzman sistemin iletişimini sağlar. Problemlere üretilen sonuçların nasıl üretildiği ve niçin o sonuçlara varıldığını açıklar. Uzman sistemin bir uzman gibi görülmesi bu ara birimi ve açıklama yeteneğinin güçlü olmasına bağlıdır.

Uzman sistemlerin bir çok **kullanıcıları** vardır. Bunlar şu kimliklerle karşımıza çıkabilirler; Doğrudan danışmanlık isteyen ve uzman olmayan müşteri, öğrenmek isteyen bir öğrenci, bilgitabanını geliştirmek veya arttırmak isteyen uzman sitem tasarımcısı, uzman vb. Kullanıcıların bilgisayarlar hakkında bilgileri veya problemler hakkında derin bilgileri olmayabilir. Ama birçok kişi uzman sistemleri kullanarak daha çabuk ve muhtemelen daha az maliyeti olan kararlara ulaşmak isterler. Uzman sistemlerin kapasiteleri zaman ve çaba tasarrufu sağlamak için geliştirildiğinden, bunlar kullanıcılara geleneksel bilgisayar sistemlerinin aksine en kısa cevapları sağlarlar.

Görüldüğü gibi uzman sistem elemanlarından birisi de uzmanın kendisidir. Bir uzman sistemi geliştirmek için en az bir uzmana ihtiyaç vardır. Bilgi mühendisi bilgi temini yöntemlerini kullanarak uzmandan bilgileri elde eder. Topladığı bilgileri derleyerek gereksiz bilgileri uzmanın yardımıyla ayıklar ve bilgileri bilgisayarın anlayacağı biçime getirerek bilgi tabanına koyar. Kullanıcı uzman sisteme bir soru sorduğunda, çıkarım mekanizması bilgi tabanını arayarak sorulan sorunun cevabını arar. İlgili bilgileri belirleyip probleme

özüm ürettikten sonra kullanıcı arabirimi aracılığı ile kullanıcıya sorusunun cevabı verilir. Bazı durumlarda problemleri özecek bilgiler bilgi tabanında bulunmayabilir. Bu durumda bilgi tabanına yeni bilgi eklenmesi veya güncellenmesi olasıdır. Zaman içinde bilgi tabanında yeterli bilgileri toplamak söz konusu olabilir. Bilginin oğalması ise çıkarım mekanizmasının problemlere özümmler üretme gücünü attırmaktadır.

➤ **Uzman Sistemlerin Kısıtları:**

- Uzman Sistemlerin dış dünya ile bağlantıları yetersizdir.
- Tabandaki bilgi yüzeyseldir.
- Aşırı derecede uzman kişilerin bilgisine muhtaçtır.
- Hemen hemen hiç öğrenme becerisi yoktur.
- Kullanım alanı şimdilik sınırlı sayılabilir.
- Akıl yürütme metotları sınırlıdır.
- Bilgi sunumu metotları sınırlıdır.

➤ **Uzman Sistemlerin Avantajları:**

- Uzman sistemler, uzman insanlardan daha hızlı olduėu için bu hız farkı üretime artış olarak yansıyabilir. Bu hız, aynı zamanda üretim döngüsünün daha kısa sürede tamamlanmasını ve bunu yaparken de sistemin sürekliliğini korumasını sağlar. Sürekliliğin korunması için zaman içerisinde veri ve bilgi kaybı gerçekleşmez.
- Uzmanların yerine bir uzman sistem devreye geçirilebilirse, uzman insanlar için yapılan harcama ortadan kalkar ve böylece dolaylı olarak üretimde artar. Uzman insanların yerine uzman sistemlerin kullanılmasının bir diğer getirisi kalite düzeyinin yükselmesidir.
- Uzman sistemler tasarlanmasındaki amaca yönelik olarak, bir insandaki tüm özellikleri taşımasından dolayı “emin değilim”, “bilmiyorum” gibi kesinlik belirtmeyen ifadelere de yer verilmesi, değerlendirme ve kıstaslarını buna göre ayarlaması yani tam ve kesin olmayan bilgi ile çalışabilme gerçekleştirilebilmektedir.
- Uzman sistemler, ara sonuçların gösterilebilmesi açısından bu sistemlerin kullanıldığı bazı alanlarda ara sonuçları raporlar halinde

görmek ve ilerisine yönelik plan ve tasarı yapmak amacı ile istenildiği anda müdahale edilebilecek şekilde tasarlanabilir.

- Uzman sistemler geleceğin fabrikalarının gelişmesine katkıda bulunacak ve bu doğrultuda öngörülen insansız fabrikaların uzman sistemler ile dahada gelişmesi sağlanabilinecek. Özellikle 4. sanayi devrimi (Endüstri 4.0) ile bu konuda büyük bir atılım beklenmektedir.

➤ **Uzman Sistemlerin Dezavantajları:**

- Günümüz koşullarında her alanda yeteri kadar uzman bulunamamakta, bulunsa dahi yeteri zamanı ayıramamaktadır. Bu durum da uzmanlık bilgisinin olmaması yani bu bilgiye ulaşmanın zorluğunu ortaya koymaktadır. Ayrıca uzmanlardan bilgi edinilmesindeki güçlükler bir diğer sorun olarak karşımıza çıkmaktadır çünkü kimi uzmanlar bildiklerini aktaramamaktadır.
- Aynı konuya uzmanların farklı bakış açıları getirmesinin bir sonucu olarak, uygulanacak alana göre değişiklik göstermesiyle birlikte uzmanların benimsediği ve doğru olarak ortaya koyduğu bilgiler tutarsızlık arz ediyor olabilir. Yada, çok ekonomik ve pratik bir sonuç içermeyebilir. Bu gibi sorunların aşılması ancak dinamik bir uzman sistem tasarımının oluşturulması ve buna bağlı olarak öğrenme yeteneğinin kazandırılmasıyla, gerçek bilgiyi ortaya çıkarabilme imkanı olabilir.
- Maliyet yüksekliği ve zaman fazlalığı bir sorun olarak tezahür eder, bunun nedeni ise uzman sistem tasarlamamanın yüksek maliyeti ve buna ek olarak, bilgi mühendisi, uzman ve programcı ile 3-5 yıl arasında değişen sistemlerin maliyetleridir.
- Özgünlük, insana haiz bir beceri olması nedeni ile alışlagelmemiş bir durumla karşılaşıldığında, uzman insanın yeteneği ile yeni bir çözüm sunulabilirken uzman sistem bunu yapamamaktadır. Fakat uzman sistemlere öğrenme becerisi kazandırılabilir olmasına rağmen bu sistemi tasarlamak başlı başına bir problem teşkil etmektedir. Uzman sistemin aksine, uzman bir insanın öğrenme yeteneği çok kolay bir şekilde gerçekleşmektedir.

➤ **Uzman Sistemlerin Kullanım Alanları:**

- **Yorumlama:** Sensörlerden gelen durumların tanımlanması.
Burada, gözlemlerden durum yorumu çıkartılır. Keşif, konuşma, anlama, görüntü analizi, sinyal anlama gibi analizleri içerir. Gözlenen verilere sembolik anlamlar verilerek işlemler yapılır.
- **Tahmin:** Verilmiş durumlara benzer sonuçların çıkarılması.
Burada; hava tahminleri, trafik, nüfus tahminleri, tahıl üretim tahmin hesapları ve askeri alanlarda tahminler yapılabilir. Tahmin sistemi parametrik dinamik model kullanır. Bu modelde parametre değerler, verilen duruma göre değişir. Modelden elde edilen sonuçlar tahmin için esas teşkil eder.
- **Teşhis:** Gözlem neticelerine göre sistem bozukluklarının tespiti.
Gözlemlerden bozuklukları bulan sistemlerde kullanılır. Tıbbi, elektronik, mekanik ve yazılım tanıları bu sınıfa girer. Tanı sistemleri gözlenen mekanik bozuklukların altında yatan nedenleri bulmak ile ilgilidir. Bu amaçla iki teknikten biri kullanılır:
a) Davranış ve tanı arasındaki bağıntılara dayanan tablolar kullanılır.
b) Gözlemlerle, bozukluğa neden olma olasılığı olan tasarım, yapım yada bileşenlerdeki olası hata bilgileriyle sistem tasarımı bilgilerini birleştirir.
- **Tasarım:** Sınırlı şartlar altında nesne tasarımı.
Tasarlama sistemi, tasarlama kanıtlamalarını sağlayan nesne konfigürasyonlarını geliştirir. Bu sistem; devre planları, bina tasarımı ve bütçe gibi konulara uygulanmıştır. Tasarlama sistemleri nesnelerin birbiriyle çeşitli ilişkilerinin tanımını yapar. Ayrıca maliyet ve istenmeyen özellikleri ölçen amaç fonksiyonunu minimize etmeye çalışır.
- **Planlama:** İşlemlerin tasarımı.
Bu sistemler, fonksiyona sahip nesnelerin, tasarım problemleri üzerinde özelleşmişlerdir. Bunlar otomatik programlama, robotlar, projeler, yol, iletişim, deney ve askeri planlama problemleriyle ilgilidir. Planlama sistemleri, planlanan eylemlerin etkilerini görmek için eylem modellerinden yararlanır.

- İzleme: Hassaslıkları planlamak için gözlemlerin karşılaştırılması.

Bu sistemler, sistem davranışları gözlemleriyle, planlama sonuçlarının başarısı için, zorunlu görülen özellikleri karşılaştırır ve sistemin işleyişi konusunda bilgi verir. Nükleer Güç Santrallerinin Düzenlenmesi ve Maliyet Yönetimi gibi kullanım örnekleri verilebilir.

- Hata Ayıklama: Hatalara sebep olan bozuklukların sunulması

Örnek Kullanım Alanı: Bilgisayar Yazılımı

- Tamir: Belirlenmiş yönetim planının yürütülmesi

Örnek Kullanım Alanı: Otomobil, Bilgisayar

- Eğitim: Öğrenci davranışlarının tespiti ve düzeltilmesi

Örnek Kullanım Alanı: Danışma, Islah, Tedavi

- Kontrol: Sistem Davranışının yorumu, tahmini, tamiri ve izlenmesi

Örnek Kullanım Alanı: Hava Trafik Kontrolü, Savaş Kontrolü

Günümüzde bilginin kullanıldığı hemen her alanda Uzman Sistemler kullanılmaktadır. Bazı Uzman Sistemler araştırma aracı olarak kullanılırken, bazıları önemli iş ve endüstri alanlarında kullanılmaktadır. Yabancı para değerlerinin takibi ve tahmini, yatırım danışmanlığı, kredi yönetimi ve müşteri değerlendirme, sigorta risklerini değerlendirme ve yatırım fırsatlarını değerlendirme gibi alanlarda Uzman Sistem kullanımı yaygınlaşmaktadır. Literatürde verilen bilgilere dayanarak değişik alanlarda hazırlanmış olan Uzman Sistemlerden bazıları Tablo 5.1’de verilmiştir:

Tablo 5.1: Uzman Sistemler ve uygulama alanları

Alanlar	Uzman Sistem	Amacı
Biyoloji	CRYALIS	Proteinlerin 3 boyutlu yapılarını yorumlama
	DENDRAL	Moleküler Yapı Yorumlama
	CLONER	Yeni biyolojik yapı tasarımı
	MOLGEN	Gen kopyalama (klonlama) deneylerinin tasarımı
	SECS	Kompleks organik moleküllerin tasarımı
	SPEX	Moleküler biyoloji deneylerinin planlanması
Elektronik	ACE	Telefon Ağlarındaki arızaların teşhisi
	IN-ATE	Osiloskop hatalarının teşhisi
	NDS	Ulusal iletişim ağının teşhisi
	PALLADIO	Yeni VLSI devrelerinin tasarım ve testi
	CADHELP	Bilgisayar destekli tasarım yardımı
	SOPHIE	Devre arıza teşhisi yardımı
Tıp	PUFF	Akciğer hastalıklarının teşhisi
	VM	Yoğun bakım hastalıklarının incelenmesi
	AI/COAG	Kan hastalıklarının teşhisi
	CADUCEUS	Dahili hastalıkların teşhisi
	MYCIN	Bakteriyel enfeksiyonların teşhis ve tedavisi
	ONCOCIN	Kemoterapi hastalarının tedavi ve idaresi
	ATTENDING	Anestezi işlem talimatı
	GUIDON	Bakteriyel enfeksiyonlar için talimat
Termodinamik	REACTOR	Reaktör kazalarının teşhis ve çözümü
	DELTA	GE lokomotiflerinin teşhis ve çözümü
	STEAMER	Buhar santralinin çalışma talimatı
Madencilik	LITHO	Petrol kuyularının verilerinin yorumlanması
	MUD	Sondaj problemlerinin teşhis ve çözümü
	PROSPECTOR	Mineral arama çalışmaları için jeolojik verilerin yorumlanması
Bilişim	BDS	Şalterli ağlarda bozuk kısımların teşhisi
	YES/MVS	IBM MVS işletim sistemi için kontrol/izleme

5.2. Uzman Sistem Tasarımı

Bir Uzman Sistem tasarımına başlamadan önce, çözümü aranan problem için Uzman Sistemlerin en uygun yöntem olup olmadığına karar vermek gerekir. Örneğin geleneksel programlama ile çözüm bulunabilen bir problem için Uzman Sistemlere başvurmak gereksizdir. Bu konuda bir karara varabilmek için, göz önünde bulundurulması gereken bazı kriterler mevcuttur. Bunlardan bazıları şunlardır:

a) Problem, yordamsal programlama ile çözülebilir mi?

Eğer bu sorunun cevabı 'Evet' ise, bu durumda Uzman Sistemlerle uğraşmanın en iyi çözüm olmadığı barizdir. Örneğin arıza yapan bir cihazdaki hatayı tespit etmek için, bir arızaya yol açan bütün nedenler bilindikten sonra yapılması gereken tek şey, sadece bir hata ve çözüm tablosuna bakmak yeterli olur. Uzman Sistemler daha çok algoritmik çözümü olmayan, eldeki verileri kullanarak sonuç çıkarmak gereken problemler için uygun olur.

b) Uygulama yapılacak alanın sınırları kesin olarak belli mi?

Tasarlanacak Uzman Sistemin bilmesi gerekenler ve yeteneklerinin sınırları tam olarak belirli olmalıdır. Örneğin baş ağrılarına teşhis koymak için kullanılan bir Uzman Sistem düşünelim. Böyle bir Uzman Sistem için bir tıp doktorunun bilgisine başvurmak gerekir. Fakat baş ağrılarının nedenlerini çok ayrıntılı bir biçimde araştırmak, tavsiyede bulunmak için nörologun da bilgisine ihtiyaç duyulur. Daha ayrıntıya girildikçe biyokimya, kimya, moleküler biyofizik vs. gibi pek çok alan ile ilgili bilginin de bilgi tabanına eklenmesi gerekir. İşte böyle durumlara düşmemek için, Uzman Sistemin alanının mutlaka bir şekilde sınırlandırılması gerekir. Aksi takdirde Uzman Sistem çok karmaşık hale gelir.

c) Uzman Sisteme ihtiyaç var mı?

Pek çok uzmanın bulunduğu bir alanda Uzman Sistem hazırlamak pek de mantıklı olmaz. Çünkü böyle bir Uzman Sistemin dayanak noktası az bulunan uzman bilgisinden ziyade, çok sayıda uzmandan rahatlıkla alınabilecek bilgiye dayanır.

d) İşbirliği yapılabilecek en az bir uzman kişi mevcut mu?

Tasarlanan projeye ilgi duyan en az bir uzman kişinin bulunması gereklidir. Çünkü herkes, bilgisinin doğruluğunun test edilmesini kabul etmeyebilir. Bir projeye çok fazla sayıda uzman kişinin katılması da bazen sakıncalı olabilir. Bir problemin çözümü için iki uzman farklı testler ve çözümler önerebilir. Bazen, farklı sonuçlara da varabilirler.

e) Uzman kişi, bilgisini anlaşılabilir derecede açıklayabiliyor mu?

Bir Uzman Sistem hazırlanırken, bilgisine başvuru uzman kişinin kendisine net bir biçimde ifade edebilmesi çok önemlidir. Tasarım yapan mühendis, uzman kişinin söylediklerini net bir şekilde anlayamazsa, aldığı bilgiyi bilgisayar koduna dönüştürmesi çok zor olur.

➤ Tasarımda Bulunması Gereken Özellikler:

- **Yüksek Performans:** Bir Uzman Sistem programı, sorulan sorulara uzman bir insana denk veya daha iyi bir düzeyde cevap verebilmelidir.
- **Hızlı Cevap Verme:** Tasarlanan sistemin, sorulan sorulara yönelik bir sonuca makul bir sürede varabilmesi ve hatta uzman bir insandan daha çabuk karar verebilmesi gerekir. Örneğin, bir uzmanın bir saatte sonuca vardığı bir konuda, Uzman Sistemin bir yılda karar vermesi elbette işe yaramaz.
- **Güvenilirlik:** Hazırlanan Uzman Sistemin güvenilir olması, hata vermemesi gerekir.
- **Anlaşılabilirlik:** Tasarlanan sistemin, bir konuda vardığı sonucun aşamalarını tek tek açıklayabilmesi gerekir. Sonuca nasıl vardığı meçhul olan bir sistemden ziyade, tıpkı bir insan uzman gibi, gerektiğinde vardığı sonucun nedenlerini açıklayabilmelidir.
- **Esneklik:** Bir Uzman Sistemde kullanmak üzere büyük miktarda bilgi yüklemek gerekir. Bu yüzden bilgi ilave etmek, değiştirmek ve silmek için etkin bir mekanizmanın Uzman Sisteme eklenmesi gerekir. Kural-Tabanlı Sistemlerin (Rule-Based Systems) popüler olmasının önemli nedenlerinden biri, kuralların etkin ve modüler bir biçimde saklanabilme özelliğidir.

➤ **Uzman Sistemin Tasarım Adımları:**

1. Uzman Sistem tasarımı için bir araç seçilir.
2. Problemin tam olarak ne olduğu belirlenir sistemin içermesi gereken bilgi analiz edilir.
3. Sistem dizayn edilir. Yani akış diyagram ve matrisleri, sistemin kurallarının taslakları bu dizaynın içerisinde yer alır.
4. İlk adımda belirlediğimiz araç kullanılarak sistemin bir prototipi oluşturulur. Bu prototip, bilgi tabanının oluşmuş halini, test edilmesini ve birçok denemenin gerçekleştirilmesini içerir.
5. Sistem istediğimiz hale gelene kadar üzerinde genişletmeler ve testler yapılarak sistem gözden geçirilir.
6. Sistem son halini alır ve yeri geldikçe üzerinde yeni düzenlemeler yapılabilir.

➤ **Uzman Sistemlerin Programlanması:**

Geleneksel programlama dilleri (C, Java, vb.) prosedürel olarak bilginin (sayılar, diziler gibi) işlemsel açıdan değişikliğe uğratılabilmesi için geliştirilmiş ve optimize edilmiş dillerdir. Bununla birlikte genelde insanlar kompleks problemleri soyut ve sembolik yöntemlerle ve geleneksel diller kullanılarak, tam olarak ifade edilemeyecek yollardan çözerler. Soyut bilgi bu dillerde modellenemesine rağmen işlemsel programlama dizileri ile kullanılabilecek biçime bilgiyi dönüştürmek için hayli fazla bir programlama eforu kaydetmek gerekir. Yapay zeka alanında yapılan araştırmaların sonuçlarından biri de soyutlandırmanın yüksek seviyelerinde bilginin modellenmesini sağlayan tekniklerin geliştirilmesi olmuştur. Kullanılan araçlarla veya dillerle iyice somutlaştırılan bu teknikler programların insan mantığına daha benzer olarak oluşturulmasını ve daha kolay geliştirilip ilerletilmesini sağlarlar. İşte, iyi tanımlanmış problem alanlarındaki insan tecrübesini simüle eden programlara, Uzman Sistemler denir. Uzman sistem araçlarından kullanışlı olanlarının ortaya çıkması ile de uzman sistemlerin oluşturulmasında harcanan efor ve maliyeti azalmıştır.

Uzman sistemlerin genel tekniklerinden biriside karakter ve kelime eşleştirme tekniğidir. Geliştirilmiş bir sistemin bilgi tabanındaki herhangi bir

değişiklik sistemin tümünü etkilemez. Kendi kendilerine karar vermek için karar üniteleri vardır. Bu ve benzeri özellikler, uzman sistemleri diğer programlardan farklı kılmaktadır. Uzman sistem nesnelerini bu tip dillerle ifade etmek verimli bir yoldur. Bu nedenle uzman sistem yazılımlarını geliştirmek için kullanılan programlama dillerinden en önemli iki tanesi PROLOG ve LISP dir. Bu dillerle programlama ve hata ayıklama işlemleri hızlı bir şekilde yapılabilmektedir. Bundan başka CLIPS, OPS5, ART, ART-IM, ECLIPSE ve ILOG bu dillere örnek teşkil eder.

a) PROLOG (Programming Logic): Lojik üzerine temellendirilmiştir. Bir PROLOG programı olaylar ve kurallar denilen bir dizi ifadeden oluşmaktadır. Bu da PROLOG'u anlaşılır ve düzgün sentaksli bir hale getirir. Kural tabanlı bir dildir ve bir kuralın doğru olması için bilgisayarın bütün kuralların buna uyup uymadığını kontrol etmesi gerekir. Bir işin nasıl yapılması gerektiği değil, yapılması için neye ihtiyaç duyulduğu ve neyin doğru olduğu bilgisi çıkarılabilir.

Örnek:

"*sort([5,3,7,2], Answer)*" prosedürünün sonucunda "*Answer=[2,3,5,7]*" döndürülür.

PROLOG programlamada;

Nesneler hakkındaki olaylar ve aralarındaki ilişkiler tanımlanır.

Nesneler ve ilişkileri hakkındaki kurallar tanımlanır.

Nesneler ve ilişkileri hakkındaki sorular sorulur.

b) LISP (List Processing language): Bir çok yönüyle kolay ve fonksiyonel bir dildir. LISP aynı zamanda yorumlanabilen bir dildir. LISP uygulamaları etkileşimli, esnek ve yinelemelidir ve programları derlemeye yada link etmeye gerek yoktur. LISP programları liste yapısı ile temsil edilir ve temel işlemleri listeler üzerindeki işlemlerdir. LISP, programın çalışması sırasında otomatik bellek yönetimi ve veri alanı ayırabilme imkanı sağlayan bir dildir. Bu özelliği sayesinde boşlukları ayarlama işlemini çok verimli olarak gerçekleştirir ve programcuyu karmaşık ve esnek programlar

oluşturmada özgür bırakır. LISP çok az sayıda veri yada veri tipi tanımını gerektirir ve hatasız bir LISP programı matematiksel olarak çok kolay sağlanır.

Bir LISP programı fonksiyon tanımları kümesini içerir ve bir fonksiyondan başka bir fonksiyonun çağırılması çok kolaydır. Bir LISP programcısı kolayca ve sık sık değişiklikler yapabilir ve her seferinde tüm programı derlemek zorunda kalmadan bunları deneyebilir. LISP'te yeni fonksiyonları oluşturulması çok kolay olması uzman sistem dizaynında çok önemlidir. LISP programları kendi kendinin değiştirebilme yeteneğine sahiptir.

Bir çok orijinal uzman sistem kabuğu LISP ile yazılır. Çünkü :

- Temel veri yapısı listedir.
- Programlar ayrıca liste yapıları ile temsil edilebilir.
- Temel işlemleri listeler üzerindeki işlemlerdir.

Bir LISP programı, basitçe sembolik ifadelerin sıralanması olarak ifade edilebilir.

Örnek:

```
>> (* 1/4 (+ 1/2 1/4))  
3/16
```

Burada $\frac{1}{4}$ sembolik ifade, $(+ \frac{1}{2} \frac{1}{4})$ ise listedir.

c) PYTHON: Uzman sistem ve yapay zeka programlamada son zamanlarda en fazla tercih edilen dil olarak Python karşımıza çıkmaktadır. Kolay kodlama yapılabilmesi ve hatasız olması bu popülerliğin başta gelen sebepleridir. Kolay kodlamadan kasıt, Python'un söz dizimlerinin farklı yapay zeka algoritmalarında uygulanabilir olması, bunun yanında diğer dillere kıyasla daha kısa sürede yazılması ile öne çıkmasına neden olmaktadır.

Ayrıca Python'un kullanıcılarına sunduğu geniş kütüphanesi de tercih sebeplerinden bir tanesidir. Mesela temel paketlerden biri olan *NumPy* matematiksel hesaplamalarda yardımcı olurken, başka bir paket olan *Scipy* ileri derece kodlamada kullanılır. Hatta yapay zeka uygulamaları için *Pybrain* adında bir paketi de bulunmaktadır.

Bunların yanında Python'ın öğrenim rahatlığı da tercih sebeplerinden biridir. İnternette fazlasıyla kaynak olması ve çoğu programcının kullanmasından dolayı karşılaşılan sorunların çözümlerinin internette olması öğrenimini kolaylaştırmaktadır.

Sonuç olarak, uzman sistem ve yapay zeka programlama için kullanılacak dil genel olarak hangi alt alan üzerine yoğunlaşılacağına ve hangi özelliklere ihtiyaç duyulduğuna göre değişmektedir. Mesela hızın önemli olduğu bir yazılımda, C++ uygun olabilirken, mantıksal problemlerle baş etmenin gerekli olduğu bir yazılımda PROLOG daha uygun bir seçim olacaktır. Her dilin farklı yönlerden cazip olmasının yanı sıra, günümüzde kullanım rahatlığı ve geniş kütüphanesiyle PYTHON bir adım öne çıkan programlama dilidir.

5.3. Bölüm Kaynakları

1. E. Öztemel, *Yapay Sinir Ağları*, Papatya Yayıncılık, 2012.
2. URL: https://web.itu.edu.tr/~sonmez/lisans/es/uzman_sistemler_giris.pdf, Mart 2019.
3. URL: http://kerqun.baun.edu.tr/20172018Guz/YZ_Sunumlar/Uzman_Sistemler_Tugce_Ozturk.pdf, Mart 2019.
4. URL: <http://www.elektrik.gen.tr/2015/08/uzman-sistemler-genel-bakis/414>, Mart 2019.