

BİLİŞİM TEKNOLOJİLERİNİN UZMAN SİSTEM BOYUTU İLE ANALİZİ*

Mehmet YILDIZ**

ÖZET

Rekabetin, rakiplerle ortaklaşa hareket ederek müşteriler lehine değer yaratacak faaliyetlere odaklanması şeklinde algılandığı günümüz rekabet koşulları altında, stratejik rekabet üstünlüğü sağlamak isteyen işletmeler, geleneksel yönetim anlayışlarından daha farklı stratejiler geliştirmelidir. Bilişim teknolojilerinin üretim süreci tasarımında kullanımı, işletmelere bu stratejileri şekillendirmelerinde önemli katkılar sağlamaktadır. Temel olarak üretim süreci, ürün, müşteri ve rekabet koşullarının şekillendirdiği elektronik sektörde, stratejik rekabet üstünlüğü sağlamada bilişim teknolojilerinin etkisi göz ardı edilemez. Uzman sistemler, işletmelere karşılaştıkları problemleri uygun algoritmalar kullanarak hızlı, etkin ve doğru şekilde çözme konusunda önemli destekler sağlamaktadır.

Anahtar Kelimeler

Bilişim Teknolojileri, Uzman Sistemler, Programlama Dilleri

ABSTRACT

In today's competition conditions where competition is perceived to behave jointly with the competitors, businesses that want to gain strategic competition superiority need to develop diverse strategies compared with their traditional managerial understandings. Using the information technologies on the production process designs provides the businesses with crucial contributions to form these diverse strategies. In the electronics industry where production process is basically determined by product, customer and competition conditions on the gaining the strategic competition superiority information technologies have crucial effects. Expert systems help businesses to solve the problems rapidly, effectively and accurately by using the algorithms.

Keywords

Information Technologies, Expert Systems, Programming Languages

* "Bilişim Teknolojilerinin Stratejik Rekabet Üstünlüğü Sağlanmasında Üretim Süreci Tasarımına Etkisinin Uzman Sistem Yaklaşımı ile Analizi ve Türk Elektronik Sanayi Uygulaması" başlıklı doktora tez çalışmasından alıntıdır.

** Selçuk Üniversitesi, İktisadi ve İdari Bilimler Fakültesi.

1.GİRİŞ

Dünyada yaşanan hızlı değişim ve küreselleşme süreci, iş dünyasında yoğun bir rekabet ortamının doğmasına neden olmuştur. Bu yeni iş dünyası da işletmelerin başarılı olabilmesi diğer işletmelerden farklı olmalarına, müşterileri için değer yaratabilmelerine ve bunu sürekli kılabilmelerine bağlıdır. Günümüz küresel rekabet ortamında faaliyette bulunan işletmelerin dönemsel başarı hedefi ile rekabet etmek yerine gelecek dönemlerdeki başarıyı hedef olarak rekabet etmeleri gerekmektedir. Bir başka ifadeyle kalite, hız, maliyet ve müşteri memnuniyeti avantajı sağlayarak uzun dönemli bir rekabet avantajı sağlamak işletmelerin başarısı ve uzun süreli varlıkları için asgari koşullar haline gelmiştir.

Ölçek ekonomisinin öngörmüş olduğu, birim başına daha düşük maliyet elde edebilmek için yüksek hacimli üretim yerine, her bir müşterinin birbirinden farklı ve bağımsız istek ve ihtiyaçları olabileceği gerçeği ışığında, çeşit ekonomisi ilkeleri doğrultusunda birbirinden farklı çok çeşitli ürün yada hizmeti olabildiğince kısa sürede üretebilme çabası günümüz işletmelerinin faaliyet gösterdikleri sektörde varlıklarını sürdürülebilmelerinin adeta bir ön koşulu haline gelmiştir.

Hızla değişim gösteren pazar koşulları dikkate alındığında, küresel anlamda rekabet edebilir bir pozisyonda olabilmek için, günümüz işletmeleri etkili bir üretim sistemine sahip olmak ve ürettikleri ürün yada hizmetleri son kullanıcı olan müşterilerine hızlı ve istenilen özelliklerde teslim etme zorunluluğu ile karşı karşıyadır. Yeni ekonomi koşullarında üretim yapan işletmeler iş yapma biçimlerini yeniden gözden geçirme durumu ile karşı karşıya kalmışlardır. Rekabet, pazar, müşteri, tedarikçiler, iç ve dış çevresel faktörlerdeki yeni oluşumlara tepki olarak günümüz işletmeleri teknolojidenden ve özellikle bilişim teknolojilerinden daha çok faydalanma stratejisini izlemektedirler.

1.Bilişim Teknolojileri

Bilginin elektronik makineler aracılığıyla düzenli ve akla uygun bir biçimde işlenmesini sağlayan bilişim teknolojilerinden yararlanmak hem insan olarak, hem örgütsel hem de yönetsel olarak yönetime çok sayıda kolaylıklar sağlamaktadır. Bu açıdan yapılan araştırmalarda bilişim teknolojilerinin örgütsel düzeyde kullanılmasında çok önemli etkiler yap-

tiğine dair çok sayıda bulgular bulunmaktadır¹. Genel bir ifadeyle bilginin toplanmasını, işlenmesini, saklanmasını ve gerektiğinde herhangi bir yere iletilmesi ya da herhangi bir yerden bu bilgiye erişilmesini bugün için elektronik, optik vb. tekniklerle otomatik olarak sağlayan sistemler bütünü olarak tanımlanacağı gibi², bilginin toplanması, saklanması, işlenmesi, erişilmesi ve dağıtılmasına hizmet eden teknolojiler, uygulama ve hizmetlerin bütünü olarak da açıklanabilir³.

Çağımızda yaşanan gelişmelerin temelinde bilişim teknolojileri yatmaktadır. Bilişim teknolojileri bireyleri, işletmeleri ve hatta ülkeleri olumlu ya da olumsuz etkilemektedir. Bilişim teknolojilerinin günümüzde temelini oluşturan Internet'in geçmişi çok eski olmasa da büyük bir hızla gelişmiş ve bunun sonucunda yeni bir dünya yapısı ortaya çıkartmıştır. 1990'lı yılların başında küreselleşme tartışılırken bilişim teknolojilerinin ortaya çıkardığı yapı bu tartışmaları ister istemez sona erdirmiştir. Sanal ortamla oluşan bu yapı engel tanımaksızın büyümüş ve ülkeler arası sınırları kaldırmıştır. İletişim, zaman ve doğru bilgi kavramının stratejik unsurlar arasında kabul edildiği günümüzde en gözde araç Internet ve ilintili teknolojileri olmuştur⁴.

Bilişim teknolojilerindeki gelişmeleri incelediğimizde internetin sadece bilgisayarlar aracılığı ile kullanılabilen bir teknoloji olmadığını görürüz. Günümüzde yeni nesil mobil telefonlar daimi internet bağlantısını destekleme yolunda geliştirilmekte, internet televizyonları yoluyla bu gelişime katkı sağlamaktadır. Ayrıca hızla yaygınlaşan dijital televizyon yayıncılığı bilişim teknolojilerinin kullanım yelpazesini geliştirmektedir. Diğer yandan günümüz koşullarında yönetim faaliyetlerinin boyutları genişlemiş, daha etkin kararlar alabilmek için gerekli bilgi gereksinmesi ve seçeneklerin adedi artmıştır. Artan rekabet koşullarında üretilen bilgi, hızla üretilip hızla tüketilen bir yapıya sahiptir. Bu noktada bilgi, üretildiği andan itibaren çabuk tüketilmeli aynı zamanda organizasyon içinde pay

¹ Türkoğlu, Recep, "**Bilişim Teknolojilerinden Bilgi Yönetiminde Yararlanma Düzeyi (Bir Örnek Olay Çalışması)**",

http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=615, Erişim Tarihi: 28.08.2005, s.8.

² Yurdakul Ceyhan, M. Ufuk Çağlayan, "**Bilgi Teknolojileri Türkiye İçin Nasıl Bir Gelecek Hazırlamakta**", Türkiye İş Bankası Kültür Yayınları, Genel Yayın No: 361, Ticaret Basım Sanayi, Ankara, 1997, s.12.

³ Halime İnceler Sarıhan, "**Teknoloji Yönetimi**", Desnet Yayınları, İstanbul, 1998, s.9.

⁴ Sevim ve Öncel, **a.g.m.**, s.3.

laşılabilirliktir. Bu anlamda bilişim teknolojilerini yönetsel amaçlı kullanılması kaçınılmaz olmaktadır⁵.

İşletmelerde kullanılan bilişim teknolojileri; bilgisayarlar, bilgi girdi ve çıktı donanımları ve iletişim araçlarından oluşur. Bilişim teknolojileri, işletmelerin bütün birimlerinde kullanılmaktadır. Pazarlamada satışların takibi, pazar araştırmaları ve bunlardan elde edilen bilgilerin değerlendirilmesinde, satış raporlarının hazırlanması ve sınıflandırılmasında, dağıtımda, satılan ürün miktarları ve satıcıların elindeki ürün miktarının bilinmesinde, elde mevcut hazır ürünlerin miktarının takibinde ve sipariş vermede bilişim teknolojileri etkin bir biçimde kullanılmakta ve işletmelere büyük faydalar sağlamaktadır. Üretimde; üretimin planlaması ve kontrolüne yönelik olarak geliştirilen programlar sayesinde bu tür işlemler daha kolay ve hızlı bir şekilde yapılmaktadır. Üretimde kullanılan bilişim teknolojileri ile teknik ve yöntemler; Bilgisayar Destekli Tasarım, Bilgisayar Destekli Üretim, Malzeme İhtiyaç Planlaması, Otomatik Depolama ve Dağıtım Sistemlerinde kullanılmaktadır. Finansal konularda, planlama, tahminlerin yapılması, muhasebe kayıtlarının tutulması ve takibi, fatura düzenleme vb. birçok faaliyet bilgi teknolojileri sayesinde daha kolay ve hızlı yapılabilir hale gelmiştir⁶.

Bilişim teknolojilerindeki gelişmeler, bu teknolojinin ürettiği bilgi birikimi insanlığın ortak ürünü haline getirmiştir. Ancak buradaki temel sorun bu bilgileri verimli bir şekilde kullanan birey, işletme, ülke ayakta kalabilmekte, kullanamayanlar ise rekabet edememektedir. Bu anlamda bilgi birikimini kullanmak rekabet edebilmede yeter koşul değildir. Örneğin, ülkemizde bilgisayar, cep telefonu gibi ileri teknolojilerinden faydalanıyoruz, ancak kısa bir süre sonra bu ürünler teknolojik ömrünü yitiriyor. Yani bu alanda yatırım ancak takip edebilme adına gerçekleşmektedir. Satın alınan bir donanım esasında yeni çıkacak bir başka donanımın habercisi olmaktadır. Sonuç olarak ikinci yönelim insanlığın bu ortak ürünü olacak bilgi birikimine katkıda bulunmak gerekmektedir. Katkı sağladığınız oranda bilişim teknolojilerinden faydalanabiliriz. Bilişim teknolojilerinin örgütlerde yaygın bir şekilde uygulanması sonucu örgütlerdeki değişimin büyük bir çoğunluğunu bilişim sistemlerini kapsayan değişim çalışmaları içermektedir. Bu anlamda, bilişim sisteminin gel-

⁵ ve Mesut Şerafettin Sevim Öncel, "İşletmelerde Bilişim Teknolojilerinin Kullanım Düzeyinin Belirlenmesine Yönelik Bir Saha Çalışması", <http://inet-tr.org.tr/inetconf8/bildiri/128.doc>, Erişim Tarihi: 23.08.2005, s.1.

⁶ Deniz Seyran, "Bilişim Teknolojilerinin Türk İşletmelerindeki Durumu ve Kullanımında Ortaya Çıkan Olumsuzluklar" İstanbul: Bilişim 98 Fuarı, Bildiriler Kitabı, s.123.

iştirilmesi bir örgütte örgütsel değişimi yerleştirmenin ana unsuru olduğu söylenebilir⁷.

2. Karar Verme Problemlerinde Sayısal ve Sezgisel Yöntemlerin Kullanımı

Sayısal Yöntemler, işletmelerde karşılaşılan karar verme problemlerinin çözümü için, bilimsel yaklaşıma göre geliştirilen matematiksel modellerin kullanılmasıdır⁸. Sayısal Yöntemler çözme teknikleri sayesinde var olan nicel değerler ışığında deterministik modeller olan, doğrusal ve doğrusal olmayan programlama, şebekeler, hedef programlama, tamsayı programlama, deterministik dinamik programlama ve deterministik stok modelleri ile olasılıklı modeller olan, karar teorisi, olasılıklı dinamik programlama, bekleme modelleri, simulasyon ve markov zincileri gibi faaliyetlerde en akılcı ve etkin kararı vermeye yönelik çözümler bulma hedeflenir. Sayısal Yöntemler konusunda çalışan uzman kişiler, en iyi çözümü elde etme amacı ile birçok matematiksel, istatistiksel ve sınıflandırma içerikli model kullanır⁹.

Bazı matematiksel modeller ise mevcut matematiksel algoritmalar ile çözülemeyecek kadar karmaşık bir yapıda olabilir. Bu tür problemlerde en iyi çözümü bulma yerine sadece en çok kabul görececek bir çözümü hedefleyen sezgisel(heuristic) yöntemler kullanılır. Hüristikler, uzmanlara, problem çözme ya da karar verme aşamasında destek olan stratejiler ya da yaklaşık hesaplama ile elde edilen kurallardır. Uzmanlar, hüristikleri uzun süreli çalışmalarından, tecrübe, deneyim, bilgi ve becerilerinden elde etmektedirler¹⁰. Sezgisel yöntemler, normal olarak olsa olsa mantığını kullanarak probleme kabul görececek çözüm arayan yöntemlerdir¹¹. Matematiksel modeller izlenecek stratejileri belirleyen sistemleri temsil

⁷ Mahmut Tekin, Hasan Kürşat Güleş ve Adem Ögüt, **Teknoloji Yönetimi**, Nobel, Ankara, 2003, s.53.

⁸ Mahmut Tekin, **Sayısal Yöntemler (Bilgisayar Çözümlü Alıştırmalar)**, Konya, 2004, s.3.

⁹ http://tr.wikipedia.org/wiki/Say%C4%B1sal_y%C3%B6ntemler, Erişim Tarihi: 12.02.2002, s.1.

¹⁰ Demet Bayraktar, **Kalite Güvence Sistemindeki Bazı Öğelerin Denetim Sürecine İlişkin Bilgi Tabanlı Bir Uzman Sistem Yaklaşımı**, Yayınlanmamış Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 1995, ss. 24-25.

¹¹ Hamdy A. Taha, (Çev. Ş. Alp Baray ve Şakir Esnaf), **Yöneylem Araştırması**, İstanbul, 2000, s.4.

etmek için kullanıldıkları halde, sezgisel modeller temel olarak inceleme dışı bırakılan alternatif stratejileri geliştirmek için kullanılmaktadır¹².

Sayısal Yöntemlere ait modellerin kullanılması, yöneticilerin karar verme ve problemlerin çözümünde işletmenin sahip olduğu temel yetenekleri tam olarak kullanmalarını sağlamaktadır. Sayısal Yöntemler bilim dalı, en uygun çözümü bulmada mevcut veriler arasında anlamlı ilişkiler kurarak mevcut kısıtlayıcı şartları da gerçekleştirecek şekilde çözüm bulmayı hedef alır. Böylesi bir araştırma stratejisinin en önemli faydası ise beyin gücünün bütün imkanları ile kullanılmasına imkan vermesidir. Başka bir ifade ile iyi bir analitik araştırmanın, çok iyi bir beyin ve bilimsel bütünlük gerektirdiği söylenebilir¹³. Yöneylem Araştırması (YA) olarak da adlandırılan sayısal yöntemlerde, temel hedef problemlere bilimsel yolla yaklaşarak modele ait parametreleri, modelin yapısını ve bunların test edilmesini de kapsayan sınırlayıcı koşullar altında karar verme açısından en iyi çözümün bulunmasıdır¹⁴. Neredeyse tüm sayısal yöntemler teknikleri yapısında yineleme (iterasyon) bulunan ve her yineleme sonunda çözümün optimumuna (en iyiye) daha yakın hale getirildiği hesaplama algoritmalarıyla sonuçlandırılır. Algoritmaların bu yapısı oldukça uzun ve zaman alıcı hesaplamalara neden olduğu için, bu algoritmaların bilgisayar yazılım programları ile çözüme ulaştırılması kaçınılmaz olmaktadır¹⁵.

2.1. Uzman Sistemler (Expert Systems)

Uzman sistemler, 1970'lerde yapay zeka alanındaki araştırmacılar tarafından geliştirilen ve ticari olarak 1980lerde uygulanmaya başlayan bir tür bilgisayar programıdır. Bu programlar, belirli bir problem hakkındaki bilgiyi çözümleyen, problemlere çözümler sağlayan, tasarımına bağlı olarak, düzeltmeleri yapmak için bir iş dizisi öneren programlardır. Uzman sistemler, insan tarafından yapılan işlerin bilgisayarlara daha iyi nasıl yaptırılacağına araştırmasını yapan bilim dalı olan yapay zeka programlama tekniklerinin bir dalıdır. Uzman sistem genellikle, konusunda uzmanlaşmış insanların üstlendiği zor bir görevi gerçekleştirmek için oluşturulan, bilgi ve çıkarıma dayanan bir bilgisayar programıdır. Uzman insanlar alanındaki bilgilere dayanarak mantıksal çıkarımda bulunarak

¹² Berna Dikçınar, **Planlama Sürecinde Bilgi Teknolojileri-Planlama Destek Sistemi Modeli**, Yayınlanmamış Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2000, s.16.

¹³ Behçet Altaylı, **Yönetim Kararlarında Kantitatif Yöntemler-Yöneylem Araştırması**, Ankara, 1996, s.3.

¹⁴ Şule Özkan, **Yöneylem Araştırması Nicel Karar Teknikleri**, Ankara, 2005, s.1.

¹⁵ Taha, **a.g.e.**, s.4.

sonuca ulaşır. Uzman sistemlerde, yine sahip oldukları bilgiye dayanarak çıkarımda bulunup sonuca varır¹⁶. Uzman sistemler, uzmanlara şu temel noktalarda yardımcı olur¹⁷.

- Problemi anlama
- Problemi çözmeye
- Çözümü açıklama
- Çözümü değerlendirme
- Bilgileri genişletme
- Yeteneklerin değerlendirilmesi
- Bilgilerin yapılandırılması

Uzman sistemler, insan tecrübe ve deneyimlerine dayanan problem çözmeye ve bazı faaliyetleri gerçekleştirmeye yönelik bilgi tabanlı bilgisayar sistemleridir. Uzman sistemler, analitik modelleri kurulamayan birçok gerçek hayat probleminin sezgisel modellerle ifadesini ve çözümünü mümkün kılmaktadır. Uzman sistemler yardımı ile yöneticiler, şu anda çözümü mümkün olarak düşünülmemiş çok karmaşık problemler ile daha çok bilgiyi işlemeyi mümkün kılan güçlü bilgisayar donanımlı iş istasyonlarına sahip olabilmektedir. Yöneticiler bu iş istasyonlarını kullanarak, verdikleri kararların doğruluk ve etkinliğini test edebilmektedir. Bu sayede, yönetim kademesinin her aşamasındaki yönetici iş akışını yakından izleyebilecek, problemleri sezip operatörlere zekice önerilerde bulunabilecek uzman sistemlerden destek alabilmektedir. Uzman sistem kullanımının işletmelere getirdiği bu avantajlar uzmanlara ve profesyonellere olan ilgiyi de beraberinde getirmektedir¹⁸. Bu nedenle kendi alanında uzmanlaşmış kişilerin işletme organizasyon yapısına dahil edilmesi bu işletmelere önemli rekabet avantajı sağlayacaktır.

2.1.1. Uzman Sistemlerin Genel Yapısı

Geleneksel karar problemleri ele alındığında, kararı etkileyen faktörlerin ağırlıkları belirlenirken, özellikle karar vericinin yeterlilik düzeyi ile

¹⁶ Hasan H. Önder, "Uzaktan Eğitimde Bilgisayar Kullanımı ve Uzman Sistemler", The Turkish Online Journal of Educational Technology - TOJET July 2003 ISSN: 1303-6521 Volume 2, Issue 3, Article 17, s.2.

¹⁷ Ömer Uysal ve Mehmet Kurban, "Elektrik Enerji Sistemlerinin Uzman Sistemler Kullanılarak İşletilmesi", I. Ege Enerji Sempozyumu Ve Sergisi, Pamukkale Üniversitesi Mühendislik Fakültesi Denizli, Mayıs 2003, s.1.

¹⁸ Süleyman Ersöz, "Üretim Pazarlama Entegrasyonunda Uzman Sistemler", Yayınlanmamış Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 1998, ss.53-54.

geçmiş verilerin ve bilgilerin tutarlılıkları önem kazanmaktadır. Karar probleminin, ele alındığı zaman dilimi içerisinde, anılan veri ve bilgilerin hazır olacağı varsayımı ihmal edilemeyecek kadar önemlidir. Bu noktada uzman sistem kullanımının geleneksel yaklaşımlardan temel farklarından birisi ortaya çıkmaktadır. Uzman sistem, sahip olduğu bilgi tabanı ile geçmiş deneyimlerden elde edilen verilerin, uzman insan bilgisinin ve bilimsel bilgilerin probleme aktarılabilmesini ve özellikle kararı etkileyen tüm faktörlerin bir dökümünün oluşturularak modelde eksik kalan her hangi bir nokta olmamasını sağlamaya adaydır. Bunun yanı sıra, karar probleminin çözülmesi esnasında geleneksel yaklaşımlar karar vericinin o anki durumundan ve eğilimlerinden etkilenmeye çok açıktır. Karar vericinin riske göre aldığı tavırdan oluşan riski seven, riskten kaçan olarak tanımlanan profilleri, karar problemini tamamen etkilemeye açıktır.

Bilgi Edinimi: İşletme problemlerin uzman sistem geliştirilerek çözümü hedeflendiğinde bu çözüme katkıda bulunacak bilgi, genellikle işletmelerin yönetim bilişim departmanlarında çalışan bilgi mühendisleri tarafından toplanmaktadır. Elde edilen bu bilgiler uygun yapay zeka dili yazılım programlarından biri kullanılarak kodlanır ve bilgi tabanına kaydedilir. 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, çözüme ulaştırılmak istenen problemle ilgili temel verileri, kuralları ve sezgisel (heuristic) bilgiyi kapsamaktadır. Bir uzman sistemin etkinliği bilgi tabanında var olan bilginin kalitesi ile doğrudan ilgilidir. Bundan dolayı, bilgi tabanındaki bilgi çözüme ulaştırılmak istenen problemi ne kadar iyi temsil ederse, geliştirilecek olan uzman sistemin aynı oranda etkin olacağı söylenebilir. Bir uzman sistemin gücünün önemli bir bölümünü bilgi tabanına yerleştirilen bilgi oluşturmaktadır. Ancak bu bilgi klasik veri tabanlarındaki gibi, bilgilerin toplanıp depolanması şeklinde olmayıp, temsil ettiği problemin yapısını yansıtarak, problemi anlamaya da yönelik olarak ifade edilmelidir. Bu problemin yapısı ve problemi anlama özellikleri, geniş ölçüde bilginin ifade edilme şekline bağlıdır. Bilgiyi ifade etmenin birçok yolu bulunmakla birlikte en çok kullanılan ifade teknikleri, kurallar, çerçeveler ve ifade ağlarıdır. Bunlarda kurala dayalı ifade şekli için çerçeveler ve ifade ağları tercih edilmektedir¹⁹. Literatürde uzman sistemlerin "bilgi tabanlı sistemler" olarak

¹⁹ Aydın Tabur, "Uzman Sistemler ve Yapay Zeka",

http://www.koubm.org/article/article_info.asp?AID=37, Erişim Tarihi: 12.11.2005. s.2.

adlandırılmasının en önemli nedenlerinden biri, uzman sistem geliştirilmesinde kullanılan bilginin öneminden kaynaklanmaktadır. Bir başka ifade ile, uzman sistemlerin merkezini bilgi oluşturmakta ve uzman sistemin etkinliği bu bilgiden doğrudan etkilenmektedir.

Başarılı ve kaliteli bir bilgi tabanı oluşturabilmek için uzmanlar aşağıdaki konularda yardıma ihtiyaç duyarlar²⁰:

- *Mantıktaki boşlukların tanımlanması*: Kuralları listelemek bu boşlukları anlamayı ve yorumlamayı sağlayamaz.
- *Çakışmaların saptanması*: İnsan uzmanların oluşturacakları sistemlerde mantık çakışmaları söz konusu olabilmektedir.
- *Eksik kriterlerin saptanması*: Çalışmalar insan uzmanın oluşturduğu sistemde bazı eksik faktörlerin bulunduğunu ve bunların başka yöntemlerle tanımlanması gerektiğini göstermektedir.
- *Gereksiz kriterlerin ve kuralların saptanması*: Birçok uzmanın gereksizliğe neden olan yanılgılara gitmesi gerçek hayatta sıkça karşılaşılan bir sorundur.

Çıkarım Mekanizması: Bir uzman sisteme ait en önemli elemanın bilgi tabanı olmasına rağmen, bu bilgi tabanını kullanıp sonuç üretebilecek iyi bir çıkarım mekanizması olmaması durumunda, bilgi tabanından etkin bir şekilde fayda sağlanamayacağı söylenebilir. Çözümüne ulaştırılmak istenen problemin uygun çözüme ulaştırıldığı ve temel fonksiyonu çıkarım ve kontrol olan alan çıkarım mekanizmasıdır. Çıkarım mekanizması temel olarak uzman sistem çalışma prosedürlerini belirleyen talimatlardan oluşur. Bilgi tabanındaki bilginin bu talimatlar çerçevesinde kullanılarak, problemin nasıl çözüme ulaştırılacağına yönelik uygun çözüm bulunmaya çalışılır. Çıkarım mekanizması, ileriye ve geriye doğru zincirleme şeklinde iki mekanizmadan oluşur. Problem hedefinin verildiği ve rahatlıkla formüle edilebildiği durumda kullanılan ileriye doğru zincirleme, formüle edilen bir problemin bilgi tabanındaki bilgiler çerçevesinde vereceği uygun çözümü bulmayı hedef alır. Hedef tanımlamanın zor olduğu veya birden fazla hedefin olduğu durumda kullanılan geriye doğru zincirleme ise, ileriye doğru zincirlemeden farklı olarak elde edilen bir sonucun nedenlerini araştırmayı hedef alır. Bilgi tabanında bulunan bilgilerin ifade ediliş şekline göre, değişik çıkarım mekanizmaları mevcuttur. En çok kullanılan bilgi ifade tarzı olan kural yorumlayıcısı olarak ad-

²⁰ Erçetin Öz ve Ömer Faruk Baykoç, "Tedarikçi Seçimi Problemine

Karar Teorisi Destekli Uzman Sistem Yaklaşımı, Gazi Üniv. Müh. Mim. Fak. Der., Cilt 19, No 3, 275-286, 2004, s.3.

landırılıp, bu yorumlayıcıların yapılarında genellikle IF-THEN tipi kurallardan faydalanılır.

Çalışma Belleği: Çalışma belleği, çözüme ulaştırılmak istenen problemle ilgili geçmiş çözümler dahil tüm verilerin kaydedildiği alandır. Başlangıçta boş gibi görünen çalışma belleği, daha sonra kullanıcının etkileşim boyunca sakladığı olguları ve sistem tarafından çıkarılan olguları kapsayacaktır. Çalışma belleği, çözüme gidilirken elde edilen olguları kayıt altına alır ve bu nedenle de dinamik bir yapıya sahiptir. Tipik bir etkileşim, kullanıcının bir arabirim yardımıyla uzman sisteme erişmesiyle gerçekleşir ve kullanıcının sağladığı ilk girdiler çalışma belleğinde depolanır. Daha sonra, uzman sistem bilgi tabanındaki kuralları kullanmak üzere çıkarım mekanizmasını çalıştırır. Ayrıca, çıkarım mekanizmasının ürettiği yeni olgu ve sonuçlar da çalışma belleğinde depolanır. Bu işlem, sistem kullanıcı tarafından sonlandırılana ya da tüm kurallar kullanılına kadar devam eder. Programın çalışması sonunda çalışma belleği, sistemin nihai çözüme erişinceye kadar ulaştığı tüm ara sonuçları ve kullanıcı tarafından sisteme girilen tüm olguları içerir²¹.

Açıklama Düzeni: Tanımlanmış olan problemi çözüme ulaştırmak için kullanılmış olan uzman sistem mantığının kullanıcı tarafından anlaşılmasını sağlayan birim açıklama düzenidir. Bu birim sayesinde, hangi işlemin neden yapıldığını kavrayabilen kullanıcıların etkileşimli olarak sisteme gerektiğinde müdahale ederek mevcut duruma uygun güncellemeleri yapmaları ve kendi bilgi birikimlerini sisteme yansıtmaları mümkün olmaktadır. Kullanıcı, bilgisayarla etkileşim esnasında sisteme müdahale ederek ne yapıldığını ve neden böyle bir akıl yürütme izlendiğini sisteme sorabilir. Açıklama düzeni bir uzman sistem yapısının zorunlu bir birimi olmayıp, sistemi kullanacak olan kullanıcıların sistemin çalışma şeklini kavraması, kolaya kullanım imkanı sağlaması ve kullanıcı müdahalesine imkan tanınması açısından geliştirilen birçok uzman sistem yapısında var olan bir birimdir.

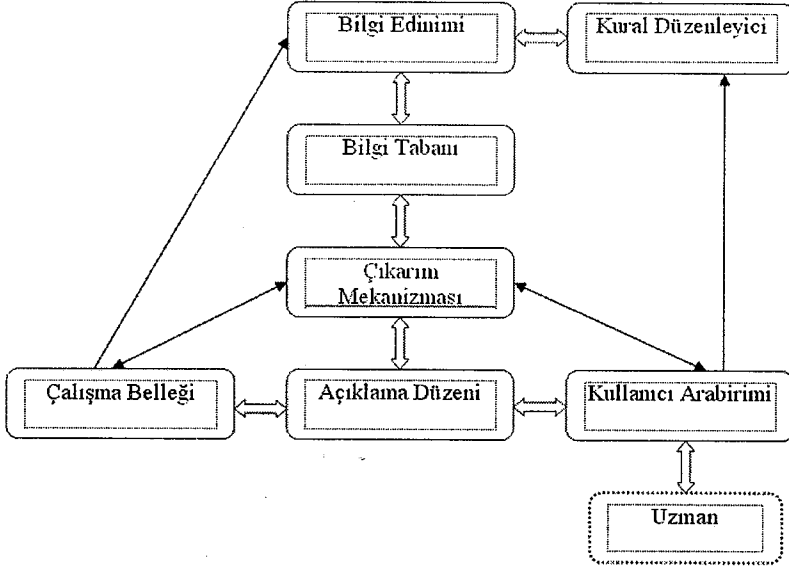
Kullanıcı Arabirimi: Uzman sistemler, kullanıcı ile bilgisayar arasında probleme yönelik iletişimin sağlanması için bir dil işleyici içerir. Bu iletişim, en sağlıklı doğal dil ile yapılır. Kısaca kullanıcı ara birimi kullanıcının, uzman sistemin çalışması esnasında bilgisayar ekranında gördüğü görüntüdür. Kullanıcı arabirimi, kullanıcı ile bilgisayar arasında bir çevirmen rolünü üstlenmiştir²². Kullanıcı arabirim düzenininin, kullanıcıdan

²¹ Ahmet Ufuk Karadeniz, "Uzman Sistemlerden Üretim Yönetiminde Yararlanma ve Tüpraş Örneği", Yayınlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi, Sosyal Bilimler Enstitüsü, 2002, s.57.

²² Özgür İrfan ve diğerleri, "Kalıp Frezeleme Uygulamaları İçin Bir Uzman

bilgiyi kabul etmesi veya sistemden bilgiyi kabul ederek kullanıcın anlayabileceği bir biçime dönüştürmesi gerekir. Özel bir kullanıyı arabiriminin niteliği, uygulama çevresine, uzmanlığa, beklentilere, kullanıcılar arasındaki farklara, yazılım performansına ve mevcut deneyime bağlıdır.

Şekil 1. Tipik Bir Uzman Sistemin Genel Yapısı



Uzman sistem programlar genel olarak "Muhakeme Etme"; yani eldeki verilere göre en uygun durumu belirleme esasına göre uygun durumu belirleme esasına göre çalışırlar. Genellikle Bilgi Tabanındaki tüm kuralların muhakeme edilmesi iki teknikle gerçekleştirilir²³.

2.1.2. Uzman Sistem Geliştirmekte Kullanılan Programlama Dilleri ve Diğer Araçlar

Ölçek ekonomisinden çeşit ekonomisine geçtiğimiz günümüz küresel rekabet ortamında, işletmeler bireyselliğini keşfetmiş ve bu bireysel farklılığını kullandığı ürün ve yararlandığı hizmetlerle yansıtmak isteyen müşterilerin bulunduğu bir ortamda rekabet etmek durumundadır. Kuşkusuz böylesi bir rekabet ortamında başta müşteriler olmak üzere,

Sistem Yazılımı, Mühendis ve Makina - Cilt: 45, Sayı: 537, s.29.

²³ C. Edmund and C. Robert, "Developing Expert Systems", J. Wiley Inc., 1990.

(aktaran Murat Tolga Özkan ve Mehmet Gülesin, "Uzman Sistem Yaklaşımı ile Cıvata ve Dişli Çark Seçimi", Turk J Engin Environ Sci., 25, 2001, s.170.)

sektör, rakipler, tedarikçiler, ürün, teknoloji, pazar ve işletmenin içinde bulunduğu faaliyet alanı çerçevesindeki diğer birimler hakkında tam ve güncel bilgilere sahip olmak işletmelere önemli bir rekabet avantajını da beraberinde getirecektir. Arzu edilen bilgiyi elde etmenin en yaygın yöntemlerinden bir tanesi konusunda uzman olan kişi ve kuruluşlardan faydalanmaktır. Ancak özel ihtisas gerektiren karmaşık problemlerin çözümünde gerekli bilgi birikimine sahip uzman kişileri nicelik ve nitelik olarak bulmak çoğu zaman mümkün olmamaktadır. Üstelik zaman ve maliyet açısından da var ise böylesi bir çözüm yolu bulmak verimli olmayabilmektedir. Bu eksiklikleri gidermek için dünyanın birçok üniversitelerinde, uzman insanların bilgi ve düşüncesini simüle eden yapay zeka uygulamaları geliştirilmektedir²⁴.

İlk geliştirilen uzman sistem bilgisayar yazılım programları, amaca yönelik çözümler bulma konusunda kodlanmış olan verilerin her program için tekrarlanarak doğru sonucu bulmasına yönelik olarak geliştirilmiştir. Bu yazılım programları daha sonra uzman sistem kabuklarının gelişimine öncülük etmiştir. Uzman sistem kabukları, günümüzdeki tüm uzman sistem yazılımlarında yer alan ve bilgi tabanından farklı araçlardır. İlk uzman sistem kabuğu temel tıbbi teşhislerde bulunmak için geliştirilen MYCIN yazılım programından türetilen EMYCIN (Empty MYCIN) dir. Günümüzdeki uzman sistem yazılımları işlevleri açısından EMYCIN'den çok daha komplike bir yapıdadır. Bir uzman sistem yazılımından faydalanmak için kodlanmış olan bilgiyi sisteme girip sistemi çalıştırmak yeterli hale gelmiştir²⁵. Farklı sektörlerdeki birçok iş problemi ilgili uzman sistem yazılım programlarının geliştirilmesi ile çözüme ulaştırılarak işletmelere rekabet avantajı sağlamada önemli katkılar sağlamaktadır.

2.1.2.1. Yapay Zeka Programlama Dilleri

Daha çok deneme niteliğinde geliştirilen ilk uzman sistemler yapay zeka programlama dillerinden birini esas alarak oluşturulmuştur.

Yapay zeka kavramı ilk olarak 1956 yılında IBM'in sponsorluğunda Dartmouth College da çalışan bir grup bilgisayar mühendisinin çalışmaları neticesinde ortaya çıkmıştır. Bu mühendisler o zamanlar, mevcut çalışmalarını daha etkin kılacak yeni bir yazılım programı üzerinde çalışmanın yanı sıra insan beyni benzetiminden yola çıkarak insan beyni gibi neden-sonuç ilişkisi kurabilecek yeni bir bilgisayar geliştirmeyi de düşünüyordu. IBM'in sponsorluğunda başlatılan bu çalışmadan sonra, değişik ülkelerden birçok araştırmacı zeki davranışları programlamaya yönelik

²⁴ Çil, **a.g.e.** s.84.

²⁵ Zahedi, **a.g.e.**, s.68.

farklı çalışmalar sergiledi. Hernekadar tüm bu çalışmalar farklı sektörlerdeki birbirinden farklı problemleri çözmeye odakmış olsa da, bu çalışmalarını ortak bir nokta da buluşturan özellik programlama da kullanılan temel programlama dili idi²⁶.

1960'lı yılların ortalarına kadar bilgisayarlar oldukça pahalı ve sadece özel görevleri yerine getirmek bir başka deyişle belli bir zaman diliminde sadece tek işi yerine getirmek için tasarlanmıştı. Daha sonraki yıllarda hem bilgisayar fiyatları düşmeye başlamış hem de daha fazla işlem çok daha kısa sürelerde yapılabilir hale gelmiştir. Geliştirilen programlama dilleri de doğal olarak bilgisayarın özelliğine göre tasarlanmıştır. 1960'lı yıllarda özel görevler ve tek iş yapmak için geliştirilen programlama dilleri, bilgisayar teknolojisindeki gelişmelerin paralelinde, bilgi paylaşım sistemi ve ana bilgisayarlar milyonlarca işlemi çok kısa sürelerde gerçekleştirebilecek şekilde tasarlanmaya başlanmıştır. Programlama dilleri temel olarak, doğru cevabın özelliklerinin veri olarak bilgisayara girilerek yapılacak işlemlerin belirtilmesi bir başka deyişle bilgisayara işlemlerin nasıl yapılacağı değil ne yapacağını tanımlanması esasına dayanır. İşlemlerin mekanizmaları yerine mantığa odaklanma, verilerin gerçek ve kural olarak belirtilebilmesi ve karmaşık problemlerin bile daha hızlı ve kolay tanımlanabilmesi gibi özelliklere sahip olan programlama dilleri, yapısal, fonksiyonel ve mantıksal programlama dili olarak sınıflandırılabilir²⁷.

2.1.2.1.1. Yapısal Programlama Dilleri

Prosedürel (procedural) ve komutsal (imperative) programlama ile benzer ve ortak özellikler taşıyan yapısal (structured) programlama, kodlama tekrarı yapmamak ve yazılan kodlamaların daha kolay takip edilebilmesi için geliştirilecek olan programları alt fonksiyonlara ya da alt programlara ayırarak program kodlamada yapısal yaklaşımı benimser²⁸. Komutlar dizisi şeklinde yazılan yapısal programlama dilleri bilgisayara ne yapması gerektiğinin adım adım ve sırası ile bildirilmesidir. Yapısal programlama dillerini yazmada farklı teknik ve metodolojiler geliştirilmiştir. Bunlardan en yaygın olanlarından bir tanesi, 1970'li yıllarda IT danışmanı olan Michael A. Jackson tarafından geliştirilmiş olan yapısal programlama olup, bu teknik veri yapıları ile program yapısının uyumlu hale getirilme-

²⁶ Jay Liebowitz, **The Handbook of Applied Expert Systems**, FL: CRC Pres LLC, 1998, s.4-3.

²⁷ <http://www.mersin.edu.tr/pano/upimages/bolumler/108/prologiris.doc>, Erişim Tarihi: 12.08.2005, s.1.

²⁸ <http://javascript.about.com/od/reference/g/gprocedure.htm>, Erişim Tarihi: 12.09.2004, s.1.

sini esas alır. Edsger Wybe Dijkstra tarafından geliştirilen yapısal programlama ise programları alt bölüm ve programlara ayırarak bu alt bölüm ve programların bağımsız olarak ele alınmasını esas alır²⁹. Yapısal programlama prosedürel programlama dilleri ile temelde aynı özellikleri taşımasına rağmen özellikle 1970'li yıllarda yapısal programlamanın bir teknik olarak kabul görmesi ve yaygınlaşması ile birlikte o tarihlerden sonra yazılan birçok prosedürel programlama dili yapısal programlama dili özelliklerini bünyesine katmıştır³⁰. Yaygın olarak bilinen ve kullanılan yapısal programlama dillerine örnek olarak FORTRAN, ALGOL, COBOL, BASIC, PL/1, PASCAL, C, MODULA ve ADA gösterilebilir.

İlk **FORTRAN (Formula Translator / Translation)** programlama dili 1954-57 yılları arasında John W. Backus'ın önderlik ettiği IBM çalışma grubu tarafından IBM 704 modeli için geliştirilmiştir. Programlama dilinin ilk versiyonları büyük harflerle "FORTRAN" şeklinde isimlendirilmiş olup, Fortran 90 versiyonundan itibaren büyük harfle isimlendirmeye son verilmiştir. Başlangıçta prosedürel (procedural) programlama dili olarak geliştirilen Fortran'ın son versiyonları nesne-yönelimli programlamanın bazı özelliklerini de desteklemektedir. Fortran'ın programcılık dünyasına getirdiği önemli yenilikler vardır: Atama deyimi (assignment statement), hesaplanabilir ifade (computable statement), dizi (array), veri işleme (data process), kağıda döküm (print in paper), GOTO, IF, iterasyon (iteration), altprogram (subroutine) ve belki de en önemlisi kaynak programın makinadan bağımsızlığı ilkesi anlamına gelen taşınabilirlik (portability) kavramı. Programlama dünyasına getirdiği bu yeni kavramlarla çok kolay kullanılan ve okunan bir programlama dili haline gelen Fortran, ALGOL, PL1, PASCAL, C, ADA gibi programlama dillerinin geliştirilmesinde zemin teşkil eder³¹. Bugüne kadar geliştirilmiş olan Fortran programlama dili versiyonları ve geliştirilme yılları sırası ile FORTRAN II 1958, FORTRAN IV 1961, FORTRAN 66 1966, FORTRAN 77 1977, Fortran 90 1990, Fortran 95 1995 ve Fortran 2003 2003 tür³².

ALGOL (Algorithmic Language) programlama dili ilk olarak 1950'li yılların sonlarında (ALGOL 58) John Backus tarafından geliştiril-

²⁹ http://www.absoluteastronomy.com/encyclopedia/s/st/structured_programming.htm, Erişim Tarihi: 12.09.2003, s.1.

³⁰ <http://encyclopedia.laborlawtalk.com/structured%20programming>, Erişim Tarihi: 30.04.2004, s.1.

³¹ Timur Karaçay, "Programlamanın Evrimi", <http://mail.baskent.edu.tr/~tkaracay/angora/pe.html>, Erişim Tarihi: 22.08.2005, s.3.

³² <http://www.ibiblio.org/pub/languages/fortran/ch1-1.html>, Erişim Tarihi: 13.09.2004, s.2.

miştir. Daha sonraki versiyonları ise Peter Naur tarafından geliştirilen ALGOL 60 ve Edsger Dijkstra tarafından geliştirilen ALGOL 68 dir. Fortran programlama dilinin getirmiş olduğu bazı olumsuzlukları ortadan kaldırmaya hedef alan ALGOL, PASCAL başta olmak üzere diğer programlama dillerinin geliştirilmesinde öncülük rolü oynamıştır³³.

COBOL (Common Business Oriented Language) ilk olarak 1959 yılında Amerika Birleşik Devletleri Savunma Bakanlığı'nda Charles Phillips'in organize ettiği bir komite (Short Range Committee) tarafından geliştirildi. İsminden de anlaşılacağı gibi COBOL programlama dili temel olarak iş dünyası, finans ile kamu ve özel sektör yönetim birimleri için geliştirilmiştir. Sistem programlarını geliştirmek için yazılmadığından COBOL programlama dili ile herhangi bir işletim sistemi geliştirmek mümkün değildir. Buna rağmen, altı aylık bir çalışmadan sonra geliştirilen COBOL programlama dili kırk yılı aşkın süredir hala kullanılmaktadır. Veri işleme araştırma kuruluşu olan Gartner Group 1997 yılı verilerine göre 300 milyardan fazla bilgisayar kodu olduğunu ve bu kodlarının %80'inin COBOL programlama dili ile oluşturulduğunu hesaplamıştır³⁴.

BASIC (Beginner's All-Purpose Symbolic Instruction Code) programlama dili 1963 yılında Dartmouth College öğretim üyeleri Prof. Dr. John G. Kemeny ve Thomas E. Kurtz tarafından geliştirilmiştir. FORTRAN II ve ALGOL 60 programlama dillerine, zaman paylaşımı, metin işleme ve aritmetik işlemlerin yapılabilmesi gibi özelliklerin eklenmesi ile geliştirilen BASIC programlama dilinin günümüzde en çok kullanılan versiyonları BasicA, GWBasic, Commodore Basic V2, QBASIC, QuickBasic, TurboBasic, Power Basic ve Visual Basic dir³⁵.

1960 yılı başlarında IBM tarafından System/360 projesinin bir parçası olarak geliştirilen **PL/1 (Programming Language One)** programlama dili, bilim, mühendislik ve iş dünyası problemlerini çözmede yardımcı olmak amacıyla geliştirilmiştir. Orijinal adı NPL (New Programming Language) olan bu programlama dili, NPL kısaltması İngiltere'deki "National Physical Laboratory" için kullanıldığından "PL/1" kısaltması kullanılması uygun görülmüştür. PL/1 programlama dilinden önce iş dünyası daha çok COBOL programlama dilini, mühendisler ve bilimsel araştırmacılar ise daha çok Fortran programlama dilini kullanıyordu. Bu farklılığı ortadan kaldırmak amacıyla taşıyan PL/1 programlama dili, hem iş dünya-

³³ <http://www.engin.umd.umich.edu/CIS/course.des/cis400/algol/algol.html>, Erişim Tarihi: 23.04.2005, s.1.

³⁴ <http://www.csis.ul.ie/COBOL/Course/COBOLIntro.htm>, Erişim Tarihi: 07.09.2004, s.1.

³⁵ <http://www.webopedia.com/TERM/B/BASIC.html>, Erişim Tarihi: 21.12.2004, s.1.

sı problemlerini hem de bilimsel araştırmacıların karşılaştıkları problemleri çözmeye hedefi ile geliştirilmiştir³⁶.

Yapısal programlama dillerinden bir diğeri 1970 yılında Niklaus Wirth tarafından geliştirilen **PASCAL** (Blaise Pascal) programlama dilidir. ALGOL programlama dilini esas alarak geliştirilen bu programlama dilinin ismi Fransız matematikçi ve felsefeci Blaise Pascal'ın onuruna PASCAL olarak verilmiştir. Başlangıçta öğrencilere yapısal programlamayı öğretme amacı ile geliştirilen PASCAL programlama dili günümüzde ağırlıklı olarak eğitim ve yazılım sektörlerinde kullanılmakta olup, Macintosh işletim sistemi temel olarak PASCAL programlama dili kullanılarak geliştirilmiştir. IBM bilgisayarlarda ise ağırlıklı olarak Turbo Pascal programlama dili kullanılmaktadır³⁷.

C programlama dili 1970'li yılların başında Ken Thompson ve Dennis Ritchie tarafından UNIX işletim sisteminde kullanılmak için geliştirilmiştir. Daha sonraları birçok işletim sistemi tarafından kullanılmaya başlanan C programlama dili günümüzde yaygın olarak kullanılan programlama dillerinden biridir. Etkinliği ve sistem program ve uygulamalarını yazmada ki üstünlüğü ile ön plana çıkan C programlama dili ağırlıklı olarak bilgisayar bilim dünyasında kullanılmaktadır. En önemli programlama dili fonksiyonlarından bir tanesi de bellek yönetimi ile ilgili olanaklardır. C programlama dili, statik, otomatik ve dinamik bellek atama gibi üç farklı özelliği ile bu alanda ön plana çıkmaktadır. C programlama dili verimli, basit fakat güçlü yapısından ve bu dilde geliştirilen uygulamaların taşınabilir olmasından dolayı programcılar, mühendisler ve sistem programcıları tarafından tercih edilir ve yaygın olarak kullanılır³⁸. C programlama dili günümüzde işletim sistemi yazımında, bilimsel çalışma için yazılan programlarda, mühendislik çalışmalarında, veri tabanı uygulamalarında ve hemen hemen her türlü uygulama alanında yer almıştır³⁹.

MODULA(MODULAR LANGUAGE) programlama dili Niklaus Wirth tarafından 1970 yılı ortalarında geliştirilmiştir. Pascal programlama dilinin de Niklaus Wirth tarafından geliştirilmesinin bir yansıması olarak, bu iki programlama dili ortak özelliklere sahiptir. Daha çok Modula-2 ismi ile bilinen bu programlama dili genel amaçlı yapısal bir programlama dili olmasına rağmen, esnek yapısı ile farklı sistem ihtiyaçlarına da cevap verebil-

³⁶ <http://www.engin.umd.umich.edu/CIS/course.des/cis400/pl1/pl1.html>, Erişim Tarihi: 13.03.2005, s.1.

³⁷ Bill Catambay, "**The Pascal Programming Language**", <http://www.pascal-central.com/ppl/>, Erişim Tarihi: 12.07.2004, s.4.

³⁸ İsmet Kocaman, **C Programlama Dili**, Sistem Yayıncılık, 1998, s.1.

³⁹ Rifat Çölkesen, **İşte C**, Beta Basım Yayım Dağıtım A.Ş., 4.Baskı, 1998, s.1.

mektedir. Bu programlama dili her biri iki kısma ayrılmış modüllerin birleşiminden oluşmaktadır: alt sistem parçalarını barındıran arabirim olarak tanım modülü ve çalışma kodlarını barındıran uygulama modülü. Bu programlama dilinin en çok bilinen ve kullanılan uzantısı Modula-2 olmasına rağmen, bu programlama dilinin günümüzdeki diğer uzantıları Modula-2+, Modula-2*, Modula-3, Oberon ve Oberon-2 dir. Bu uzantılar Modula-2'nin yerine geçen daha yeni versiyonlar olmayıp, farklı amaçlar için geliştirilmiş diğer programlama dilleridir⁴⁰.

ADA(Ada Lovelace) programlama dili ilk olarak 1977-1983 yılları arasında Jean Ichbiah önderliğindeki bir çalışma grubu tarafından tasarlanmıştır.C programlama dili ile birçok ortak özelliğe sahip olan ADA programlama dilinin yaygın kullanım alanları, ülke savunması, bankacılık sistemi, ticari havacılık, iletişim sistemi, bilgisayar-destekli tasarım ve üretim sektörüdür. Örneğin, Boenig 777 uçuş sistemleri yazılımları Ada programlama dili kullanılarak geliştirilmiştir. 1983 yılında ANSI standardı, 1987 yılında ise ISO standardı haline gelen bu programlama dilinin günümüzde yaygın olarak kullanılan versiyonları Ada95 ve Ada2005'tir⁴¹.

2.1.2.1.2.Fonksiyonel Programlama Dilleri

Amerika Birleşik Devletleri'nde yapay zeka programlama dili olarak **LISP** (**L**IST **P**rocessing) programlama dili tercih ediliyordu.1958 yılında temel hesaplamaları yapmak için John McCarthy tarafından geliştirilen ve 1970-1980 yılları arasında yapay zeka programlama dili olarak yeniden tasarlanan ve kabul gören LISP programlama dilinin günümüzde yaygın olarak kullanılan lehçeleri Common LISP (CL) ve Scheme dir⁴². 1984 yılında NASA Jonhson Space Center (Jonhson Uzay Merkezi) tarafından geliştirilen CLIPS (**C** Language **I**ntegrated **P**roduction **S**ystem) uzman sistem kabuğu, LISP program dilinden kaynaklanan, kişisel bilgisayarlarda yaygın olarak kullanılamama, program geliştirilmesinde yüksek maliyetli donanım ve araçlara ihtiyaç göstermesi ve diğer program dilleri ile kolay entegre edilememe gibi problemleri ortadan kaldırmak amacı ile geliştirilmiştir.

Fonksiyonel programlama dillerinden olan **LOGO**'nun ilk versiyonu, 1967 yılında MIT (Massachusetts Institute of Technology) Yapay Zeka Laboratuvarı'nda Wally Feurzeig ve Seymour Papert'in öncülük ettiği

⁴⁰ http://encyclopedia.laborlawtalk.com/Modula-2_programming_language, Erişim Tarihi: 21.09.2005, s.1.

⁴¹ <http://www.engin.umd.umich.edu/CIS/course.des/cis400/ada/ada.html>, Erişim Tarihi: 09.01.2005, s.1.

⁴² http://en.wikipedia.org/wiki/Lisp_programming_language, Erişim Tarihi: 20.06.2004.

çalışma grubu tarafından geliştirilmiştir. LISP'in bir lehçesi olan Logo, öğrenme aracı olarak tasarlanmıştır. Modülerite, gelişmelere açık, entereaktif ve esneklik gibi temel özellikleri olan Logo temel olarak, matematik, dil, müzik, telekomünikasyon ve bilimsel faaliyetlerin programlanmasında kullanılmaktadır. Günümüzde Logo programlama dili esas alınarak geliştirilen diğer programlara örnek olarak MicroWorlds, LEGO, StarLogo, UCBLLogo ve MSWLogo gösterilebilir⁴³.

1950 yılında Harvard Üniversitesi öğretim üyelerinden Kenneth E. Iverson tarafından geliştirilen notasyonları temel alarak işlem yapana **APL** (Array Processing Language), seri işlem yapabilen programlama dillerindendir. Iverson tarafından geliştirilen notasyonları esas alan APL, diğer programlama dilleri gibi sürekli bir gelişme göstermiş olup günümüzdeki APL programlama dilleri yapısal ve modüler programlamayı da desteklemektedir⁴⁴.

SNOBOL (String Oriented Symbolic Language) programlama dili 1962 ve 1967 yıllarında AT&T Bell Laboratuvarlarında David J. Farber, Ralph E. Griswold ve Ivan P. Polonsky tarafından geliştirilmiştir. Dizgi işlemede kolaylık sağlamak amacıyla geliştirilen ve günümüzdeki son versiyonu SNOBOL4 olan bu programlama dili ticari amaçlardan çok araştırma aracı (edebiyat ve müzik veritabanı analizi) olarak kullanılmaktadır⁴⁵.

2.1.2.1.3. Komutsal Programlama Dilleri

İngiltere ve Japonya'daki araştırmacılar ise zeki programları geliştirmekte **PROLOG** (PROgramming in LOGic) programlama dilini kullanıyordu. 1972 yılında Alain Colmerauer ve Robert Kowalski tarafından geliştirilen ve lojik programlama dillerinden biri olan PROLOG çözümün nasıl bulunacağını değil problemin ne olduğunun tanımlanması esasına dayanır. Prolog kuralları yukarıdan aşağıya yazılış sırasıyla göz önüne alır ve bu kuralların elemanlarını da sağdan sola sıra ile kontrol eder⁴⁶. Tanımlayıcı (declarative-descriptive) programlama dillerinden olan PROLOG, bir problemin çözümüne giderken belirlenen hedeflere nasıl ulaşılacağını ortaya koymaktansa kullanıcının ortaya koyduğu durum (facts and rules) ve hedef (query) ile ilgili verileri yorumlayarak çözümü

⁴³ <http://el.media.mit.edu/logo-foundation/logo/index.html>, Erişim Tarihi: 16.08.2005, s.1.

⁴⁴ http://en.wikipedia.org/wiki/APL_programming_language, Erişim Tarihi: 07.07.2004, s.1.

⁴⁵ <http://www.engin.umd.umich.edu/CIS/course.des/cis400/snobol/snobol.html>, Erişim Tarihi: 09.12.2004, s.1.

⁴⁶ Semra Doğandağ, "Mantıksal Programlama", http://bm-dergi.emo.org.tr/index.php?option=com_content&task=view&id=37&Itemid=74, Erişim Tarihi: 06.05.2005, s.3.

bulmaya çalışır⁴⁷. Kural tabanlı (rule-based) ilk üretim sistemi programlama dillerinden olan **OPS** (**O**fficial **P**roduction **S**ystem), 1975 yılında Carnegie Mellon Üniversitesi (CMU) öğretim üyesi Charles Forgy tarafından geliştirildi ve 1980 yılından sonra uzman sistem geliştirilmekte kullanılmaya başlanan program dillerinden biri haline geldi. OPS5, ileriye doğru zincirleme sonuç çıkarma aracı olarak çalışan bellek birimlerini taramak sureti ile üretim sistem belleğinde kayıtlı bulunan en uygun karşılığı bularak işlem yapar. İlk OPS5 uygulamaları LISP programlama dilini kullanarak geliştirilmiş fakat daha sonraları daha hızlı işlem yapmaya imkan tanıyan BLISS (Basic Language for Implementation of System Software) programlama dili esas alınmıştır⁴⁸.

Nesne-yönelimli (object-oriented) programlama dilleri programcılara gerçek dünya paradigmalarını kullanma imkanı sağlar⁴⁹. Bu tarz programlama dillerinden biri olan **SMALLTALK**, Xerox PARC da Alan Kay, Dan Ingalls, Ted Kaehler, Adele Goldberg ve diğer araştırmacılar tarafından 1970'li yıllarda tasarlanmıştır⁵⁰. Orijinal olarak Smalltalk-71 olarak geliştirilen bu programlama dili sürekli gelişme ve yeniliklere açık olduğundan yaygın kullanım alanı bulmuş ve Actor model, Objective-C, C#, Java ve Ruby gibi birçok bilgisayar programlama dilinin geliştirilmesine önemli katkıda bulunmuştur⁵¹.

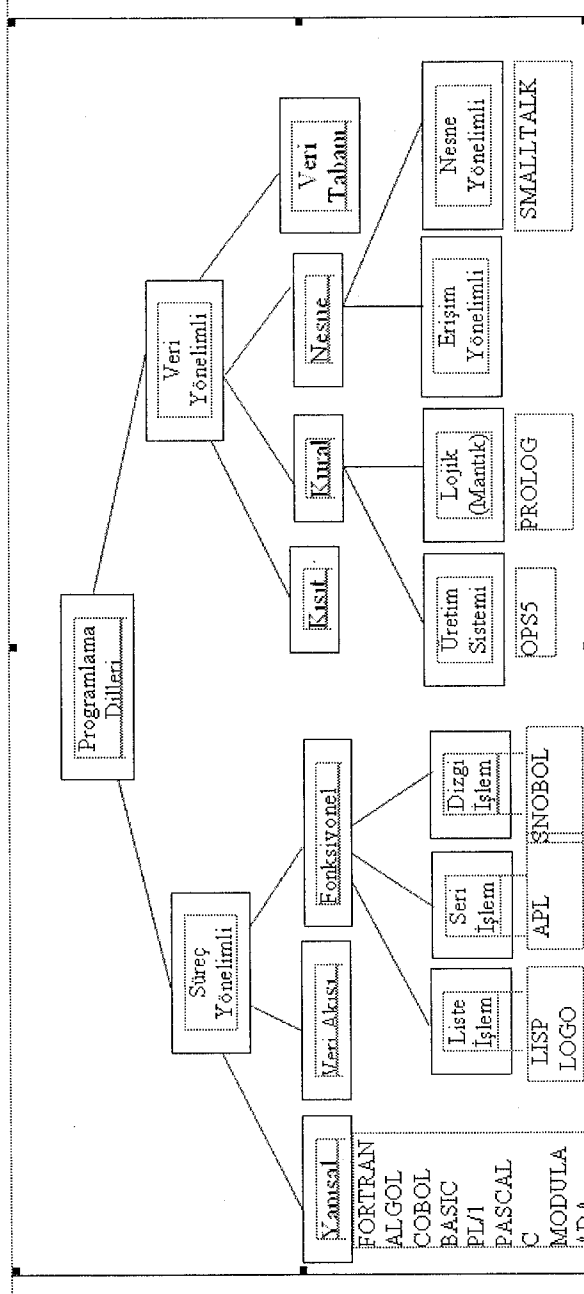
⁴⁷ Ulle Endriss, **An Introduction to Prolog Programming**, London: Ulle Endriss King's College, 2000, s. 5.

⁴⁸ <http://en.wikipedia.org/wiki/OPS5>, Erişim Tarihi: 22.06.2004, s.1.

⁴⁹ Rajah Y. Chacko, **"A Gentle Introduction To Object-Oriented Programming"**, <http://www.stc.org/confproceed/1994/PDFs/P254256.PDF>, Erişim Tarihi: 22.06.2004, s.1.

⁵⁰ <http://en.wikipedia.org/wiki/Smalltalk>, Erişim Tarihi: 01.03.2005.

⁵¹ Peter William Lount, **"What is Smalltalk?"**, <http://www.smalltalk.org/smalltalk/whatis-smalltalk.html>, Erişim Tarihi: 01.03.2005, s.3.



Kaynak: <http://cisnet.baruch.cuny.edu/friedman/cplusplus/paradigms.doc>, Eğitim Tarihi: 22.02.2005, s.3.

Şekil 2. Programlama Dillerine Genel Bakış

KAYNAKÇA

- Altaylı, Behçet; **Yönetim Kararlarında Kantitatif Yöntemler-Yöneylem Araştırması**, Ankara, 1996.
- Bayraktar, Demet; **Kalite Güvence Sistemindeki Bazı Öğelerin Denetim Sürecine İlişkin Bilgi Tabanlı Bir Uzman Sistem Yaklaşımı**, Yayınlanmamış Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 1995.
- Catambay, Bill; **"The Pascal Programming Language"**, <http://www.pascal-central.com/ppl/>, Erişim Tarihi: 12.07.2004.
- Ceyhun, Yurdakul, M. Ufuk Çağlayan; **"Bilgi Teknolojileri Türkiye İçin Nasıl Bir Gelecek Hazırlamakta"**, Türkiye İş Bankası Kültür Yayınları, Genel Yayın No: 361, Ticaret Basım Sanayi, Ankara, 1997.
- Çölkesen, Rifat; **İşte C**, Beta Basım Yayım Dağıtım A.Ş., 4.Baskı, 1998.
http://encyclopedia.laborlawtalk.com/Modula-2_programming_language,
Erişim Tarihi: 21.09.2005.
- Dikçınar, Berna; **Planlama Sürecinde Bilgi Teknolojileri-Planlama Destek Sistemi Modeli**, Yayınlanmamış Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2000.
- Doğandağ, Semra; **"Mantıksal Programlama"** , http://bm-dergi.emo.org.tr/index.php?option=com_content&task=view&id=37&Itemid=74, Erişim Tarihi: 06.05.2005.
- Edmund, C. and C. Robert; **"Developing Expert Systems"**, J. Wiley Inc.,1990. (aktaran Murat Tolga Özkan ve Mehmet Gülesin, **"Uzman Sistem Yaklaşımı ile Civata ve Dişli Çark Seçimi"**, Turk J Engin Environ Sci., 25, 2001.)
- Endriss, Ulle; **An Introduction to Prolog Programming** , London: Ulle Endriss King's College, 2000.
- Ersöz, Süleyman; **Üretim Pazarlama Entegrasyonunda Uzman Sistemler**, Yayınlanmamış Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 1998.

İrfan, Özgür ve diğerleri; **"Kalıp Frezeleme Uygulamaları İçin Bir Uzman**

Sistem Yazılımı, Mühendis ve Makina - Cilt: 45, Sayı: 537.

Karaçay, Timur; **"Programlamanın Evrimi"**,
<http://mail.baskent.edu.tr/~tkaracay/angora/pe.html>, Erişim Tarihi: 22.08.2005.

Karadeniz, Ahmet Ufuk; **"Uzman Sistemlerden Üretim Yönetiminde Yararlanma ve Tüpraş Örneği"**, Yayınlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi, Sosyal Bilimler Enstitüsü, 2002.

Kocaman, İsmet; **C Programlama Dili**, Sistem Yayıncılık, 1998.

Liebowitz, Jay; **The Handbook of Applied Expert Systems"**, FL: CRC Pres LLC, 1998.

<http://www.mersin.edu.tr/pano/upimages/bolumler/108/prologiris.doc>,
Erişim Tarihi: 12.08.2005.

Önder, Hasan H.; **"Uzaktan Eğitimde Bilgisayar Kullanımı ve Uzman Sistemler"**, The Turkish Online Journal of Educational Technology - TOJET July 2003 ISSN: 1303-6521 Volume 2, Issue 3, Article 17.

Öz, Erçetin ve Ömer Faruk Baykoç; **"Tedarikçi Seçimi Problemine**

Karar Teorisi Destekli Uzman Sistem Yaklaşımı, Gazi Üniv. Müh. Mim. Fak. Der., Cilt 19, No 3, 275-286, 2004.

Özkan, Şule; **Yöneylem Araştırması Nicel Karar Teknikleri**, Ankara, 2005.

Sarıhan Halime İnceler; **Teknoloji Yönetimi**, Desnet Yayınları, İstanbul, 1998.

Sevim, Şerafettin ve Mesut Öncel; **"İşletmelerde Bilişim Teknolojilerinin Kullanım Düzeyinin Belirlenmesine Yönelik Bir Saha Çalışması"**, <http://inet-tr.org.tr/inetconf8/bildiri/128.doc>, Erişim Tarihi: 23.08.2005.

Seyran, Deniz; **"Bilgi Teknolojilerinin Türk İşletmelerindeki Durumu ve Kullanımında Ortaya Çıkan Olumsuzluklar"** İstanbul: Bilişim 98 Fuarı, Bildiriler Kitabı.

Tabur, Aydın; **"Uzman Sistemler ve Yapay Zeka"**, http://www.koubm.org/article/article_info.asp?AID=37, Erişim Tarihi: 12.11.2005.

Taha, Hamdy A., (Çev. Ş. Alp Baray ve Şakir Esnaf); **Yöneylem Araştırması**, İstanbul, 2000.

Tekin, Mahmut, Hasan Kürşat Güleş ve Adem Öğüt; **Teknoloji Yönetimi**, Nobel, Ankara, 2003.

Tekin, Mahmut; **Sayısal Yöntemler (Bilgisayar Çözümlü Alıştırmalar)**, Konya, 2004.

Türkoğlu, Recep; **"Bilişim Teknolojilerinden Bilgi Yönetiminde Yararlanma Düzeyi (Bir Örnek Olay Çalışması)"**, http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=615, Erişim Tarihi: 28.08.2005.

Uysal, Ömer ve Mehmet Kurban; **"Elektrik Enerji Sistemlerinin Uzman Sistemler Kullanılarak İşletilmesi**, I. Ege Enerji Sempozyumu Ve Sergisi, Pamukkale Üniversitesi Mühendislik Fakültesi, Denizli, Mayıs 2003.

http://tr.wikipedia.org/wiki/Say%C4%B1sal_y%C3%B6ntemler, Erişim Tarihi: 12.02.2002

<http://javascript.about.com/od/reference/g/gprocedure.htm>, Erişim Tarihi: 12.09.2004.

http://www.absoluteastronomy.com/encyclopedia/s/st/structured_programming.htm, Erişim Tarihi: 12.09.2003.

<http://encyclopedia.laborlawtalk.com/structured%20programming>, Erişim Tarihi: 30.04.2004.

<http://www.ibiblio.org/pub/languages/fortran/ch1-1.html>, Erişim Tarihi: 13.09.2004.

<http://www.engin.umd.umich.edu/CIS/course.des/cis400/algol/algol.html>, Erişim Tarihi: 23.04.2005.

<http://www.csis.ul.ie/COBOL/Course/COBOLIntro.htm>, Erişim Tarihi: 07.09.2004.

<http://www.webopedia.com/TERM/B/BASIC.html>, Erişim Tarihi: 21.12.2004.

<http://www.engin.umd.umich.edu/CIS/course.des/cis400/pl1/pl1.html>, Erişim Tarihi: 13.03.2005.

<http://www.engin.umd.umich.edu/CIS/course.des/cis400/ada/ada.html>, Erişim Tarihi: 09.01.2005.

http://en.wikipedia.org/wiki/Lisp_programming_language, Erişim Tarihi: 20.06.2004.

<http://el.media.mit.edu/logo-foundation/logo/index.html>, Erişim Tarihi: 16.08.2005.

http://en.wikipedia.org/wiki/APL_programming_language, Erişim Tarihi: 07.07.2004.

<http://www.engin.umd.umich.edu/CIS/course.des/cis400/snobol/snobol.html>, Erişim Tarihi: 09.12.2004.

<http://cisnet.baruch.cuny.edu/friedman/cplusplus/paradigms.doc>, Erişim Tarihi: 22.02.2005.