

142885

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

142885

**YAPAY ZEKADA ZEKİ ETMENLER VE
UYGULAMALARI**

YÜKSEK LİSANS TEZİ
Müh. Gülnur BEKLERİÇ
(511981029)

Tezin Enstitüye Verildiği Tarih : 05 Mayıs 2003
Tezin Savunulduğu Tarih : 29 Mayıs 2003

Tez Danışmanı : Doç. Dr. Şakir KOCABAŞ (İ.T.Ü.) *S. Kocabaş*
Diğer Jüri Üyeleri : Doç. Dr. İbrahim ÖZKOL (İ.T.Ü.) *İ. Özkol*
Yrd. Doç. Dr. Selman NAS (İ.T.Ü.) *Selman Nas*

MAYIS 2003

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

ÖNSÖZ

Tez çalışmalarım boyunca bana yol gösteren tez danışmanım Doç. Dr. Şakir KOCABAŞ'a teşekkür ederim.

Çalışmalarım boyunca yardımcı olan ve sabır göstererek destek olan annem Zaide BEKLERİÇ ve kardeşim Soner BEKLERİÇ'e teşekkürlerimi sunarım.

Yoğun iş günlerimizde bile zaman yönünden toleranslı davranan Aksigorta Sistem Destek ve Operasyon Bölüm Müdürü sayın Sadık CANBOL'a, diğer yakın çalışma arkadaşlarımdan Tahsin ERHAN ve Fikret SARINERGİZ'e manevi destek ve sabırlarından dolayı teşekkür ederim.

Mayıs 2003

Müh. Gülnur BEKLERİÇ

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
2. YAPAY ZEKA	3
2.1. Giriş	3
2.2. Kısa Tarihçe	3
2.3. Temel Özellikleri	3
2.4. Alanları	5
2.5. Konuları	6
3. ETMENLER	8
3.1. Giriş	8
3.2. Etmen Nedir?	8
3.3. Etmenlerin Tarihi	9
3.4. Etmen Çevresinin Özellikleri	10
3.5. Etmen Türleri	11
3.6. Etmenler ve Nesneler	12
3.7. Etmenler ve Uzman Sistemler	13
4. ZEKİ ETMENLER	14
4.1. Giriş	14
4.2. Zeki Etmenler İçin Soyut Mimariler	14
4.2.1. Algılama	16
4.2.2. Durumlu Etmen	18
4.3. Zeki Etmenler İçin Somut Mimariler	19
4.3.1. Mantık Tabanlı Mimariler	19
4.3.2. Tepkili Mimariler	22
4.3.3. İnanç-İstek-Amaç Mimarileri	26
4.3.4. Katmanlı Mimariler	31
4.4. Zeki Etmenlerde Gerçek Zamanlılık	33
4.5. Zeki Etmenlerde Gerçek Zamanlı Kontrol İçin Gerekli Şartlar	34
4.5.1. Genel Çerçeve	34
4.5.2. Zeki Etmenin Çevresi	34
4.5.3. Etmen Özellikleri	35
4.6. Etmen Mimarisi	36
4.6.1. Üst Düzey Yapı	36
4.6.2. Algılama Bileşeni	37
4.6.3. Eylem Bileşeni	37
4.6.4. Düşünme Bileşeni	38

5. ZEKİ ETMEN UYGULAMA ALANLARI	40
5.1. Endüstriyel Uygulamalar	40
5.1.1. Süreç Kontrolü	40
5.1.2. Üretim	41
5.1.3. Hava Trafik Kontrolü	41
5.2. Ticari Uygulamalar	42
5.2.1. Bilgi Yönetimi	42
5.2.2. Elektronik Ticaret	43
5.2.3. İş Süreci Yönetimi	44
5.3. Tıbbi Uygulamalar	45
5.3.1. Hasta Takibi	45
5.3.2. Sağlık Bakımı	45
5.4. Eğlence	46
5.4.1. Oyunlar	46
5.4.2. Etkileşimli Tiyatro ve Sinema	47
6. UYGULAMA	48
6.1. Giriş	48
6.2. Zeki Etmen Çalışma Çerçevesi	48
6.2.1. Tasarım Amaçları	48
6.2.2. Geliştirilen Zeki Etmen Mimarisi	49
6.2.3. Çerçeveler	50
6.3. Kişisel Etmen Yöneticisi	51
6.3.1. SchedulerAgent	51
6.3.2. UserNotificationAgent	51
6.3.3. KitapEtmen	51
6.3.4. Uygulamanın Çalıştırılması	53
6.3.5. HTML Sorunu	55
7. SONUÇLAR VE TARTIŞMA	57
KAYNAKLAR	58
ÖZGEÇMİŞ	59

KISALTMALAR

BDI	:Belief – Desire – Intention
CIAgent	:Constructing Intelligent Agent
DTD	:Document Type Definition
HTML	:HyperText Markup Language
KQML	:Knowledge Query and Manipulation Language
PAManager	:Personal Agent Manager
XML	:eXtensible Markup Language



TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. Mantık Tabanlı Mimaride Eylem Kodu	21
Tablo 4.2. Tepkili Etmen Mimarisinde Eylem Kodu	24
Tablo 4.3. BDI Mimarisinde Eylem Kodu.....	30



ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 4.1 : Algılama ve Eylem Altsistemleri.....	16
Şekil 4.2 : Durumunu Koruyan Etmenler.....	18
Şekil 4.3 : İnanç-İstek-Amaç Mimarisinin Genel Şematik Çizimi.....	29
Şekil 4.4 : Katmanlı Etmen Mimarisinin Üç Türündeki Bilgi ve Kontrol Akışları	32
Şekil 4.5 : Amaçlanan Etmen Mimarisi.....	38
Şekil 6.1 : Kitap-Net Sitesi Ana Sayfası.....	52
Şekil 6.2 : Kitap-Net Sorgulama Sonuç Sayfası.....	53
Şekil 6.3 : Etmenin İşlediği HTML Kod Örneği.....	53
Şekil 6.4 : UserNotificationAgent Çalıştırılması.....	54
Şekil 6.5 : KitapEtmen ve SchedulerAgent Çalıştırılması.....	54
Şekil 6.6 : Bilgi Diyalogları.....	55

SEMBOL LİSTESİ

S	:Çevre durumları kümesi
A	:Etmen eylemleri kümesi
S	:Çevre durumu
A	:Etmenin eylemi
S^*	:Çevre durumları serisi
h	:Etmenin geçmiş serisi
P	:Algılar kümesi
I	:Tüm iç durumların kümesi
L	:Mantık cümlelerinin kümesi
D	:L veritabanlarının kümesi
Δ	:D'nin elemanı
ρ	:Çıkarım kuralları kümesi
R	:Tüm kuralların kümesi
Beh	:Davranış kuralları kümesi
$\prec$	:Kısıtlama bağıntısı
b_1, b_2	:Farklı davranış kümeleri
γ	:Dünyanın değişim oranı
Bel	:İnançlar kümesi
Des	:İstekler kümesi
Int	:Amaçlar kümesi

YAPAY ZEKADA ZEKİ ETMENLER VE UYGULAMALARI

ÖZET

İş yerimizde bilgisayar başında çalışırken, evde iş yaparken ve başka bir şeylerle uğraşırken bir yardımcımızın olmasını bazen hepimiz istemiştizdir. Mesela bir evin temizliğini yaparken en azından süpürme işini bir robot bizim yerimize yapsa, bilgisayara yeni bir donanım taktığımız zaman işletim sisteminin kendisi tanıtma işlemini halletse, İnternet’de bir şey ararken birçok arama makinesine girip çıkmadan bilgilerin hepsi önümüze gelse, bir program çıktısını diğer bir programın girdisi olarak kullanmak için çok fazla uğraşmamıza gerek kalmadan bir aracı yazılım yardımcı olsa ne kadarda zamandan kazandırırdı.

Bu tezde önemli veya önemsiz olabilecek tekrar tekrar yapılan, zaman alan işleri yapan veya yardımcı olan etmen türleri, özellikle de zeki etmenler anlatılıyor. Tezin konusu şunlardır: Birincisi, etmen ve zeki etmenlerin tanımını vererek artık birçok yerde karşımıza çıkmaya başlayan bu kavramlara açıklık getirmektir. İkincisi, zeki etmen uygulamalarını oluşturmak için etmenin kullanım amacına göre mimari çeşitlerini teorik olarak tanıtmaktır. Ayrıca zeki etmenlerde gerçek zamanlılık ve buna uygun mimarinin oluşturulması hakkında bilgi vermektir. Üçüncüsü, zeki etmen uygulamalarını kullanıldıkları alanlara göre ayırarak bu uygulamalardan ne kadar çok farklı şekillerde yararlanıldığını açıklamaktır. Son olarak, Java’da yazılmış ve İnternet ortamında çalışan bir örnek uygulamayla bu tür zeki etmenlerin nasıl geliştirileceğini ortaya koymaktır.

Tezdeki örnek uygulamada geliştirilen zeki etmen bir bilgi toplama etmenidir. Kullanıcının İnternet’den ucuz kitap bulmasına yardımcı olur. Kullanıcının yerine İnternet’den online satış yapan iki kitap sitesine girebilmekte ve kullanıcının istediği kitapları bu sitelerde araştırabilmektedir. Etmen sitelerdeki kitapları, kullanıcı tarafından girilen, kitap konularına ve yazar adlarına göre sorgulayabilmektedir. Sitelerde bulduğu kitap fiyatları ve indirimleriyle ilgili olarak kullanıcıyı bilgilendirebilmektedir. Kullanıcının belirlediği indirim kriterlerine göre çıkarımlarda bulunabilmektedir.

Bu tür bir örnek uygulamanın seçilmesinin sebebi İnternet’in bize sağladığı bilgiye ulaşmayı kolaylaştırmasıdır. Arama motorlarına bulunmasını istediğimiz bilgiyi yazdığımızda her zaman istediğimiz bilgi bize sonuç olarak sunulmaz. Çünkü arama motorları, ilgilendiğimiz sayfaların bağlantısı ile birlikte pek çok fazla şey bulur. Sonra her sayfaya tek tek girip kendimiz mukayese etmek zorunda kalırız. Bu örnek uygulamada kullanılan etmen ise bizim yerimize bu işlemleri yapar. Belli yerlerden istediğimiz bilgiyi alıp bizim için bilgi tabanında toplar. Sonra belirlenen kriterlere göre mukayeselerde bulunup kullanıcıyı bilgilendirir.

INTELLIGENT AGENTS AND THEIR APPLICATIONS IN ARTIFICIAL INTELLIGENCE

SUMMARY

Many of us would like to have an assistant when we are engaged in using computers in our office, or in our homes when we are involved in house work. For example, when we are busy with other things at home, we would like to have a robot to do cleaning, or when we plug-in a new hardware to our computer a software agent to do necessary procedures itself. Similarly, we would like to have all the necessary information collected and presented to us without us having to enter many search engines when we want to obtain information from the Internet. Also, we would like to have a software agent help us in using the output of a computer program as input to another program, without us having to deal with the data transformation ourselves.

In this thesis, the types of agents (particularly intelligent agents) are explained, that can perform time-consuming tasks important or unimportant but that need to be done repeatedly. The thesis starts with the definitions agents and intelligent agents, the concepts which are likely to meet in many areas of life. Secondly, we describe the types of agent architectures for different applications depending on their uses, and provide information on suitable architectures for real-time performance of agents. Third, we provide a summary of the different applications of intelligent agents drawing attention to the variety of their uses. Lastly, we describe how such intelligent agents can be developed, through an implementations working in the Internet medium that we developed in Java.

The intelligent agent that we describe in this thesis is an information gathering agent, capable of helping the user to collect information to purchase cheap books through the Internet. The agent is capable of entering two online book sales websites on behalf of the user, and search for the books that the user wants. The agent can do the search through the subjects of the books and the author's names given by the user, and informs the user on the books that it finds in the websites together with their special discounts, making inferences in accordance with the discount criteria determined by the user.

Our reason for selecting an application in this area is to see how to make it easy to reach information provided by the Internet. If we use the conventional search engines alone for this purpose, we do not usually get the exact information that we want. This is because the search engines return many more information together with that form the web pages that we actually needed, and we end up having to enter each web page ourselves to see the contents, and make the comparisons. In our example application, the agent performs all this work for us. It copies the information in its knowledge for us, and then informs us its search results, after making the comparisons based on the criteria that we provided.

1. GİRİŞ

1950'lerden bu yana etmenler bir çok alanda geliştirilmiş ve çeşitleri de gün geçtikçe artmıştır. Günümüzde etmenler, yapay zeka, bilgisayar bilimi, sosyoloji, ekonomi, yöntem bilim ve psikoloji gibi bir çok farklı disiplinden sonuçlar, kavramlar ve fikirleri üzerine çeken, yerleşmiş ve gelecek vaat eden araştırma ve uygulama alanlarında kullanılmaktadır. Bu geniş uygulama alanı ve çok disiplinlilik doğal olarak etmenleri birkaç kelimedede tanıtmayı zorlaştırmaktadır.

Bir etmen, bir yazılım programı veya robot gibi varlıktır. Algılayarak görebilir ve kısmen kendi deneyimiyle oluşturduğu davranışla, bulunduğu çevresi içinde otonom olarak hareket edebilir. Zeki bir varlık olarak etmen, kendisine verilen algısal ve yeterli donanımla çevresel şartların değişikliğine göre esnek ve mantıklı olarak çalışır. Etkileşimli bir varlık olarak etmen, diğer etmenleri veya insanları kendi eylemleriyle etkileyebilir.

Bu tez çalışmasında, etmen çeşitleri tanıtılmıştır. Özellikle zeki etmenler üzerinde durulmuştur. İstenen özellikte uygulamaların nasıl geliştirileceğine dair teorik olarak soyut ve somut etmen mimarileri verilmiştir. Bu çalışmadaki temel etmen oluşturma hakkındaki bilgiler, herhangi bir alandaki etmen çalışması için cesaretlendirici ve yardımcı olacağı düşünülerek verilmiştir.

Etmenler hızlı değişen, öngörü yapılamayan açık çevrelerde gurbüzlükle çalışabilmek zorundadır. Bu çevrelerde davranışların hatalı olabileceğine dair önemli olasılıklar vardır. Bu tür çevrelerde bulunan etmenler zeki etmenler olarak bilinir. Zeki etmenler için şu iki çalışma alanı örnek gösterilebilir:

Dünyadan dış gezegenlere uzun uçuş için bir uzay araştırması yaparken genellikle, yer mürettebatı devamlı olarak süreçleri izler ve beklenmedik olasılıklarla nasıl uğraşacaklarına karar verirler. Bu iş çok masraflıdır ve eğer kararlara çok acil ihtiyaç duyulursa basit bir şekilde uygulama yapılamaz. Bu nedenlerden, NASA gibi organizasyonlar ciddi olarak, daha zengin karar verme kapasitesi ve güvenilirlik veren, daha otonom çalışan araştırmalara yatırım yapmaktadır.

İnternet'de belirli bir sorunun cevabını aramak uzun ve sıkıcı bir işlem haline gelebilir. Öyleyse, neden bir bilgisayar programının - bir etmenin - bizim için aramalar yapmasını istemeyelim. Etmen, verilecek bir soruyu çeşitli farklı İnternet

bilgi kaynaklarından bilgi paracıklarını alarak sentezlemeye ihtiya duyacaktır. Belirli bazı kaynaklar yoksa hata oluřabilecekdir (belki ađ hataları yznden), veya uzak sunucudaki sonular ulařılamaz olacaktır.

Bu tezdeki uygulamada alıřma alanı olarak İnternet seilmiřtir. Ulařılmak istenen bilgi kitap olarak belirlenmiřtir. Uygulama dili olarak Java kullanılmıřtır. Sistemde  ayrı etmen kullanılmıřtır. Etmenlerin zamanlayıcı, haberci, bilgi toplayıcı ve analizci grevleri vardır. Etmenlerin ilk ikisi basit yapıdadır. ncs ise zeki bir etmendir. Zeki etmende yapay zekadaki ıkarım kuralları uygulanmıřtır. Bilgi tabanındaki verilerle ileriye dođru zincirleme ıkarım kuralı metodunu kullanılarak sonuca ulařılır. Oluřturulan etmenlerin birbirleriyle ve Java bileřenleriyle alıřabilmesi iin grafik oluřturma aracı olarak Borland JBuilder rn kullanılmıřtır. Etmenlerin dıř dnyadaki veya diđer etmenlerdeki olayları kontrol edebilmesi JavaBean bileřen modeliyle sađlanmıřtır.

Blm 2’de yapay zeka hakkında genel bilgiler verilmiřtir. Blm 3’de etmenler, trleri ve evre zellikleri tanıtılmıřtır. Blm 4’de zeki etmenler tanıtılmıř, tasarımları iin soyut ve deđiřik somut mimariler zerinde durulmuřtur. Zeki etmenlerde gerek zamanlılık ve gerek zamanlı kontrol iin gerekli řartların aıklaması yapılmıřtır. Blm 5’de zeki etmen uygulamaları alanlara ayrılarak tanıtılmıřtır. Blm 6’da uygulama anlatılmıřtır. Blm 7’de ise sonular verilmiřtir.

2. YAPAY ZEKA

2.1. Giriş

Yapay zeka, bir bilgisayarın yada bilgisayar denetimli bir makinenin, genellikle insana özgü nitelikler olduğu varsayılan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak kabaca tanımlanabilir [1].

2.2. Kısa Tarihçe

Yapay zeka Amerika Birleşik Devletleri Dartmouth'ta düzenlenen ve bir grup yüksek düzeyli bilim adamını bir araya getiren bir konferansın tarihi olan 1956'da doğmuştur. Bu konferansta J. Mc. Carthy, M. Minsky, C. Shannon, A. Newell ve H. Simon gibi bazı araştırmacıların çabaları sayesinde süregelmıştır.

İlk yapay zeka programları (bir mantık teoremleri ispatlayıcısı ve bir satranç oyunu programı; her ikisi de Newell ve Simon'ındır) ve LISP bu dönemde ortaya çıkmıştır.

Yapay zekanın doğuşuna yol açan katalizör 1943'de bilgisayarların ortaya çıkmasıdır. Bu dönemden itibaren bazı öncüler, bu makineleri biraz zeka ile donatma meselesini ele aldılar. Böylelikle 1950'de matematikçi A. Turing bir makinenin zeki olup olmadığına karar verme olanağı tanıyan bir test ortaya koydu. Temel kuralı tartışmalı olmakla birlikte bu test, bu dönemden itibaren bilgisayarın 'zeka'sına verilen önemi gayet iyi göstermektedir [1].

2.3. Temel Özellikleri

Yapay zekanın ulaştığı alanlar sınırsızdır. Bununla birlikte, bütün yapay zeka sistemleri arasında temel özelliklerini daha iyi sınıflandırma olanağı sağlayan bir ortak noktalar bütünü vardır.

Bir yapay zeka programının özelliği her şeyden önce, 'klasik' enformatikte de olduğu gibi, sayılardan çok sembolik bilgileri kullanmasıdır. Bu bilgiler, kavramları, nesneleri düşünürken bir insanın göz önünde bulundurduğu olguların aynılarını temsil eder.

Bu, sayısal işlem yöntemlerinin kullanılmasını dışlamaz, ancak sonuçların işlenmesi genellikle sembole dayalı olarak yapılır. Buna göre tıbbi teşhise yarayan bir sistem 38,2 °C sayısal değerinden hastanın hafif ateşli olduğu sonucunu çıkaracaktır.

Yapay zekada bulunan ikinci kavram da klasik algoritma metotlarının tersine “tecrübeye dayalı bilgi” metotları kavramıdır. Program yapmak için belli bir programlama dilinde kodlanmış olan algoritma, verilmiş bir problemi çözmeye yönelik işlemler dizisinin eksiksiz betimlemesinden oluşur. Tecrübeye dayalı bilginin determinist olmayan yollarını izleyen ve başarının garantili olmadığı ancak “çalıştığı” zaman genellikle işlem süresinde büyük tasarruf sağlayan bir çözüm yolunu denemesi gerekir.

Formel integral durumundaki bir problemin insan tarafından çözümlenmesi genellikle şöyle bir yol ortaya koyar: Integralin tipine göre bir değişken değişikliği denenir, sonra gerekirse bir başkası veya ilk deneme, problemin sadeleşmesini sağlamadıysa bölümler halinde bir başka integrasyon denenir. Bir problemin çözümünün tecrübeye dayalı bilgi araştırması sadece en çok umut vaat edenleri göz önünde bulundurmak üzere çözüm yollarının tamamının “kısaltılmasından” ibarettir.

Yapay zekanın bir başka özelliği de işlenen verilerin ve bilgilerin eksik, yanlış ve hatta çelişkili olduğu durumlarla yetinmesidir. Bir insan için örneğin tıpta, iş idaresinde, bankacılık sektöründe vb.de böyle gündelik durumların söz konusu olduğunu belirtmek gerekir. Yapay zeka çok çetin bir problemi tamamen çözmese de etkili metotlar sağlar.

Önemli bir yapay zeka kavramı da bilgidir. Başlangıçta araştırmacıların problemlerin çözümlenmesi için genel metotlar bulmaya çalışmalarına karşın, günümüzün yapay zekası az sayıda ve iyice sınırlanmış alanlarda, bu alanlara özgü çok miktardaki bilgiye dayanarak problemleri çözebilen sistemleri anlamakla yetiniyor. Bu bilgiler bir programlama dilinde prosedürler biçiminde kodlandıkları zaman, doğal biçimlerine daha yakın olarak, yapay zeka sistemine eklenebilirler.

Son olarak, yapay zeka, özü bakımından çok-disiplinlidir. Bir yapay zeka sisteminin gerçekleştirilmesi için geliştirilmiş enformatik tekniklerine geniş ölçüde başvurmak gerekir. Ama yapay zeka köklerini başka disiplinlerinden de alır: Mantık ve bilinç psikolojisi (bilgilerin ve kanıtlamaların gösterilme temelleri için), dilbilim (konuşma ve yazı dilinin işlenmesi için), ergonomi, felsefe ve bir gün mutlaka sinirbilim (nöroloji) ve biyoloji [1].

2.4. Alanları

Yukarıda bahsettiğimiz yapay zeka'ya özgü özellikler bir uygulama alanları kümesinde bir araya gelirler [1]:

Otomatik Teorem Kanıtlaması: Bu etkinlik yapay zeka'nın başlangıcından beri vardır. Özellikle sembolik mantıkla ispatı karışık teoremlerin daha basit ispatlarının bulunmasında kayda değer beyanlar elde edilmiştir. Bu çalışmalar ortaya koydukları genel işlem ve yorumlama metotları açısından önemlidirler.

Yazılı Anadilin İşlenmesi: Bir makineye anadilinde başvurabilmek özellikle enformatik uzmanı olmayan kişilere önemli bir yarar sağlayacaktır. Anadilin işlenmesi alanı aslında; insan-makine diyalogu bağlamında cümlelerin açıklanması ve üretilmesi, veri tabanlarına ya da servislerine giriş ve otomatik çeviri vb. etkinlikleri kapsar.

Sözün Otomatik Olarak İşlenmesi: Bu alan iki bölümden oluşur: Tanıma (makineye konuşmak ve anlamasını sağlamak) ve sentez (makineyi konuşturmak). Yazılı anadilin işlenmesi ile benzerlikler sunar. Burada yine, makine tarafından bir cümlelerin anlaşılması, birçok bilgiyi devreye sokan bir süreçtir. Fakat bazen, gürültüler ve akustik değişkenlik, benzeşen sinyalin işlenmesini zorlaştırır. Deneysel insan-makine diyalogu sistemleri geliştirilmiştir. Bununla birlikte, bu sistemler, problemin olağanüstü zorluğu yüzünden özel kelimelerle sınırlı bir sözcük dağarcığını bilmekle yetinirler.

Görüntülerin Yorumlanması ve Bilgisayar Tarafından Görme: Görme bir makinenin çevresini fark etmeye yönelik başka bir özelliktedir. Aynen işitsel algılama durumunda olduğu gibi görme probleminin basitleştirilmesi, basit görsel formların algoritmik form tanıma metotları yardımıyla tanımlanmasından ibarettir. Örneğin bir metin baskı karakteri, bir sahnenin yerleri, iyi saptanmış eşyaları, mikroskop altında bir resmin üzerindeki hücreler ya da kromozomlar, bir uydu resmi üzerindeki özel bölgeler vb. Bununla birlikte, bir sahnenin yada bir görüntünün gerçekten anlaşılması, bağlamsal bir işlemi yürütmek için düşünülen alana özgü bir bilgi bütününe dayanarak bu metotların ötesine gitmeyi gerektirir: Tıbbi teşhis amacı ile radyolojik görüntülerin açıklanması, basılı ya da elyazısı bir metnin anlaşılması, üretim zinciri üzerindeki bir nesnenin kontrolü, özerk bir aracın kılavuzluğu vb. olduğu gibi. Bu son etkinlikler özellikle robotbilime aittir. Görüntülerin işlenmesi yapay zekanın endüstriyel alanlardaki ilk uygulamalarından biridir.

Robotbilim: Robotbilimin geniş bir alanı ve özellikle de sanayide iş otomasyonu etkinliklerini kapsar (fabrikasyon, yönetim, tamir). Varolan robotların büyük bir

bölümü, işleri sırayla durmadan tekrarlar ve zeki değıllerdir. Buna karşın yeni kuşak robotlar, giderek çevrelerini algılamaya ve hareketlerini planlamaya yönelik zeka yeteneğı ile donatılmıştır. Böylece zeka, yarının sanayisinin önemli bir bileşeni olan robotbilimde sürekli artan bir rol oynayacaktır.

Oyunlar: Oyunlar (satranç, dama, tavla), araştırmacılar için yapay zekanın ta başından beri tercih edilen bir alan olmuştur. Bu çalışmaların amacı oyunla ilgili olmaktan öteye gitmiştir, çünkü tecrübeye dayalı bilgi stratejileri kullanarak genel çözüm arama kavramlarına dönüşmektedir. Bu alana yönelik çalışmalardaki gelişmeler, yapay zekanın evrimini yeterince açıklıyor. İlk sistemler kısıtlı bir zamanda çok sayıdaki çözüm yolunu göz önünde bulundurma becerisi üzerine kurulmuştur. Bu arada amaca ulaşmak için geliştirilmiş hareketler planı kavramı devreye donatılmaktadır. Bu son gelişme ile, incelenen oyuna özgü bilgilerin kullanılması söz konusu olmaktadır.

Uzman Sistemler: İnsanların tecrübelerini bilgisayara aktaran yapay zeka programlarına, “Uzman Sistemler” (Expert Systems) denir. Yapay zeka tekniklerinin en belirgin olanı ve belki de en yaygın olarak kullanılanları hiç şüphesiz uzman sistemlerdir. Bir uzman sistem, sınırlı bir sübjektif saha için uzman bilgisini depolayabilir, mantıksal sonuçları takip etmek suretiyle problemleri çözebilir. Veri işleminden bilgi işlemine bir geçiş olarak ifade edilebilir ve aynı zamanda algoritmalar ile sonuç çıkarma mekanizmaları da yer değıştirmiştir.

2.5. Konuları

Günümüzde dünyada ve ülkemizde yapay zeka denilince ilk önce uzman sistemler gelmektedir. Son 10 yıl içerisinde bu biraz değışmiş ve yapay sinir ağıları, genetik algoritmalar ve bulanık mantık da artık oldukça yoğun bir şekilde konuşulur olmuştur. Özellikle ülkemizde yapay zeka çalışmalarının çoğunluğu bu teknolojiler üzerinde yürütölmektedir. Halbuki zekanın değışik boyutlarını inceleyen daha birçok farklı yaklaşım vardır. Mesela insanların problemleri çözmede sağduyularını kullandıkları bilinen bir şeydir. Yukarıda belirtilen tekniklerin hiçbirisi direkt olarak sağduyu bilgisini işlemeye yönelik bir yaklaşım getirmemektedir. Bunun gibi daha birçok yapay zeka yaklaşımından ve araştırma konusundan bahsetmek mümkündür.

Bunlardan bazıları şöyle sıralanabilir:

Uzman sistemler,

Yapay Sinir ağıları,

Genetik algoritmalar,

Endüktif öğrenme,

Açıklama tabanlı öğrenme,

Benzerliğe dayanan öğrenme,

Kalitatif muhakeme (çıkarım) veya sağduyu bilgi işleme,

Veri tabanlı muhakeme,

Model tabanlı muhakeme,

Non monotonik muhakeme veya doğruyu koruma mekanizması,

Geometrik muhakeme,

Dağıtılmış yapay zeka,

Paralel yapay zeka sistemleri,

Zeki etmenler,

Doğal dil işleme,

Nesne tabanlı zeki sistemler,

Zeki veritabanları,

Bilimsel buluşların modellenmesi,

Kavramsal grafikler,

Bilimsel keşifler,

Zeki multimedya birimleri,

Kaos,

Mantık programlama,

Zeki öğretim sistemleri vb...

Yukarıda söz edilen birçok konu halen araştırmaya oldukça açık konulardır [1].

3. ETMENLER

3.1. Giriş

Bilgisayarlar ne yapacaklarını bilmede çok iyi değildir: Bir bilgisayarın yapacağı her hareket programcı tarafından önceden belirlenir, planlanır ve kodlanır. Eğer bir bilgisayar programı, tasarımcısının önceden göremediği bir durumla karşılaşrsa, sonuç genellikle hoş olmaz - en iyisinde sistem çöker, en kötüsünde de birçok kayıplarla karşı karşıya kalınır. Bu olağan durum bilgisayarlarla aramızdaki ilişkinin merkezinde meydana gelir. Bu durum kendini o kadar çok belirtir ki bilgisayar programlamada çok az üzerinde durulur. Ama uygulamaların giderek büyüyen boyutlarında, sistemlerin tasarım hedeflerini yerine getirmeleri için, sistemin neye ihtiyacı olduğuna kendisinin karar verebilmesine ihtiyaç duyarız. Bu tür bilgisayar sistemleri etmen (agent) olarak bilinir [2].

3.2. Etmen Nedir?

Günümüzde etmenler, bilgisayar biliminin ve yapay zekanın birçok bölümlerinin ilgi odağıdır. Etmenler email filtreleri gibi nispeten küçük sistemlerden, hava trafiği kontrolü gibi kapsamlı, açık, karmaşık, önemli görevi olan sistemlere kadar uzanan, çeşit sayısı gün geçtikçe artan uygulamalarda kullanılmaktadır.

Etmen tabanlı sistemlerin gelişme ayrıntılarına girmeden önce, etmen teriminin bize ne anlatmak istediğini açıklamamız gerekir. Bunda doğrudan zorluklarla karşılaşırız. Çünkü etmen sistemleri ve mimarileri üzerine çalışan çeşitli üniversiteler ve kuruluşlar, geliştirdikleri uygulamalar ve sistemler için çeşitli tanımlar sunmuşlardır. Tanımların farklı olması, tasarlanan etmen yapılarının, tasarım amaçlarının ve yaşam modellerinin farklı olmasından kaynaklanmaktadır. Bu yüzden kapsamlı olarak kabul edilmiş etmen teriminin tanımı bulunmaz. Esasen, etmen kavramı için merkez kelime otonom (kendi başına) için genel bir fikir birliği varken, bunun ötesindekilerde çok az kabul vardır. Farklı alanlar için farklı önemliliğe sahip etmenlerle birlikte belirtilmiş çeşitli özellikler yine zorluklar arasında yer alır. Bu nedenle, bazı uygulamalar için, etmenin deneyimlerinden öğrenebilme yeteneği en

yüksek değere sahipken, diğer uygulamalarda, öğrenme sadece önemsiz değil istenmeyen bir özellikte olabilmektedir. Fakat yinede aşağıdaki tanımın yaygın olarak birçok araştırmacı tarafından kabul edildiği bilinmektedir.

Etmen, içinde bulunduğu çevrede tasarım amaçlarına erişmek için kendi başına eyleme geçebilen ortama yerleştirilmiş bilgisayar sistemidir.

'Yerleştirilmiş' deyiimi etmenin çevresinden çeşitli uyarılar aldığını ve çevresini eylemleriyle değiştirme yeteneğinin olduğu anlamına gelmektedir. Kendi başına davranış etmenin insan yardımı olmadan hareket edebilme yeteneğidir. Ortama yerleştirilmiş kendi başlarına çalışan bilgisayar sistemleri yeni bir buluş değildir. Etmenleri diğer otonom sistemlerden ayıran en önemli özellik esneklikleridir. 'Esneklik' deyiimiyle diğer etmenler ve insanlarla iletişime girme yetenekleri ve çevrelerine karşı olan sorumlu ve önetkin (proaktif) tutumları anlatılmaktadır [3].

Etmenlerin önetkin ve sorumlu davranışları Petrie tarafından yazıcı örneğiyle açıklanır. Normal bir bilgisayar yazıcısı kullanıcı tarafından verilen bir yazma isteğini yerine getiremezse hata mesajı verir ve böylece kullanıcı ve yazıcı arasındaki etkileşim sona erer. Ama etmen tarafından yönlendiren yazıcı, geri çevrilen yazma isteğini ve geri çevrilme nedenini aklında tutar. Yazmayı engelleyen sorun çözüldüğünde, kullanıcıya haber vererek yazmayı hala isteyip istemediğini sorar [4].

3.3. Etmenlerin Tarihi

Tarihin başlangıcından beri insan olmayan yardımcılara karşı ilgi duyulmuştur. Androidler, robotlar, cyborglar ve diğer bilim kurgu yaratıkları kültürün kalıcı bir parçası haline gelmiştir. "Robot" Çekçe "hizmetçi" anlamına gelmektedir. 1921'de Karel Capek'in "Rossumun Evrensel Robotları" eseriyle robot terimi tanınmıştır.

Çeşitli otomatlar yüzyıllardır kullanılmasına rağmen, otonom etmen kavramı II. Dünya Savaşında bilgisayarların gelişmesiyle ortaya çıkmıştır. Norman günümüzdeki zeki etmen kavramının atalarının otomatik mekanik aletler olduğunu belirtmektedir, fabrikalarda kullanılan dokuma tezgahları ve uçaklardaki otomatik pilot gibi [5]. Otomatik mekanik aletler yerine otomatik dijital yazılımlar kullanma fikri Nicholas Negroponte ve Alan Kay'in öngörülleri ile taraf topladı. Alan Kay yazılım etmenleri fikrinin gelişimini şöyle açıklıyor:

"Etmen fikri 1950'lerde John McCarthy ve Oliver G. Selfridge tarafından ortaya atıldı. Amaçları belli bir hedef verilince hedefe ulaşmak için gerekli bilgisayar işlemlerini yapabilecek bir sistem geliştirmekti. Sistem çıkmaza girene kadar kendi başına

çalışacak, takıldığında ne yapması gerektiğini insanlara soracaktı. Yapacakları sistem bilgisayar dünyasında iş yapan sanal bir robot olacaktı [6].

3.4. Etmen Çevresinin Özellikleri

Bir etmenin karşılaştığı ana problem kendi tasarım amacına en uygun olan eylemleri yapmasını sağlayacak kararı verebilmesidir. Aslında etmen mimarileri, bir çevre içine yerleştirilmiş karar verme sistemleri için yazılım yapılarıdır. Karar verme işleminin karmaşık olması birçok farklı çevresel özelliklerden etkilenebilir. Russal ve Norvig çevre özelliklerinin aşağıdaki gibi sınıflandırılmasını tavsiye etmişlerdir.

- Erişilebilen / Erişilemeyen

Erişilebilen bir çevrede etmen, çevre durumu hakkında tam, doğru, ve güncel bilgiler alır. Karmaşık çevreler (örneğin gerçek dünya ve İnternet) erişilemeyendir.

- Determinist / Determinist Olmayan

Determinist bir çevredeki herhangi bir eylemin bir tek garantili etkisi vardır - bir eylemin yapılmasıyla sonuçlanacak durum hakkında belirsizlik yoktur. Determinist olmayan çevredeki etmenin bir eyleminin etkisi (gerçek dünyada olduğu gibi) her zaman aynı sonucu doğurmayabilir.

- Ayrık / Bağımlı Olaylar İçerme

Çevre ayrık olaylardan oluşuyorsa etmenin performansı bu olaylara bağlıdır. Bu olaylar arasında bir bağlantı yoktur. Ayrık olaylı çevrede etmen sadece o anki durumu değerlendirerek ne yapacağına karar verir. Şimdiki ve gelecekteki olaylar arasında bağlantı kurmasına ihtiyaç yoktur.

- Statik / Dinamik

Statik çevre, etmenin yaptığı eylemler dışında, değişmeden kalabilir. Dinamik çevrenin üzerinde diğer işlemler de çalışır ve etmenin kontrolü ötesindeki yollarla da değişme meydana gelir. Gerçek dünya en yüksek dinamik özelliğe sahip olan çevredir.

- Ayrık / Devamlı

Belirli sabit sayıda eylem ve algılamalar bulunuyorsa çevre ayrıktır. Örneğin satranç oyununda çevre ayrıktır ve taksi şoföründe çevre devamlıdır.

Genellikle en karmaşık ve etmen geliştirmenin zor olduğu çevre sınıfı erişilemez, determinist olmayan, bağımlı olaylar içeren, dinamik ve devamlı olanlardır [2].

3.5. Etmen Türleri

Etmen arařtırmalarını üç kola ayırabiliriz:

- İnanç - İstek - Amaç (Belief, Desire, Intention) Modeli
- Dağıtık Yapay Zeka (DAI), Çoklu Etmen Sistemleri (MAS)
- Daha Geniş “Etmen” Kavramı

İnanç - istek - amaç modeli (BDI) 1977’lerde başlamıştır. Yapay zeka etmen tasarımında istemli yapıyı kullanır. İstemli yapı etmeni tasarımında etmen, içinde bulunduğu dünyanın sembolik bir anlatımını tutar. Kararlarını mantıksal neden-sonuç ilişkilerine veya örüntü tanıma gibi çeşitli bilimsel yöntemlere dayanarak alır. Bu tasarım şeklinin avantajı mantıklı olduğu bilimsel olarak ispatlanmış kararlar almasıdır. Dezavantajı ise gerçek zamanlı veya zamanın kısıtlı olduğu alanlarda uygun bir karar vermek için geç kalmasıdır [7].

Dağıtık yapay zeka 1980’lerde geliştirilmiş ve hızlı bir şekilde çeşitlendirilmiştir. Birden fazla etmenli olan sistemlerle ilgilenir. Araştırmalar iki değişik sınıfa ayrılmıştır: Dağıtık problem çözme ve çoklu etmen sistemleri. Dağıtık problem çözme bir problemin birbiriyle yardımlaşan değişik modüller tarafından bilgi paylaşımı ve bilgi bölüşümüyle çözülmesiyle ilgilenir. Çoklu etmen sisteminde her etmenin kendi stratejisi vardır. Etmen, sistemin genel çıkarlarından çok kendi çıkarlarını düşünerek hareket eder. Etmenlere dışarıdan stratejiler empoze edilemez. Etmenler amaçlarına uygun olarak birbirleriyle çatışabilir veya ortak çalışabilirler. Zamanla çoklu etmenli sistem terimi dağıtık problem çözme kapsayarak geniş bir şekilde kullanılmaya başlanmıştır. Sistem içindeki etmenler otonom ve heterojen yapıya sahip olabilir. Çoklu etmenli sistemin avantajları çabukluk, esneklik ve değişen ortam koşullarına hızlı tepki yeteneği olarak gösterilebilir [3].

1990’lardan günümüze kadar “etmen” kavramı genişlemiştir. Aşağıdaki etmen türleri ortaya çıkmıştır:

- Arayüz Etmenleri: Kullanıcı ile aynı ortamda işbirliği yapan kişisel yardımcılardır. Kullanıcının davranışlarını gözlemler, raporlar ve öğrenme ile yeni yetenekler kazanır. Kullanıcısına gerekirse daha iyi yollar önerebilir [8].
- Tepkisel (Reaktif) Etmenler: Etmenlerin zeki olması şart değildir, az zekalı hatta aptal olabilirler, önemli olan eylemlerinin zekice olmasıdır. Çalışmalarının amacı

istemekten yapmaya, neden sonuç ilişkilerini ortaya çıkarmaktansa kontrollü hareketlere kaymıştır [9],[10].

- **Hareketli Etmenler:** Bu tür etmenler geniş alanlı ağlarda bir makineden diğerine gezebilir. Başka ortamlarda işlemlerini gerçekleştirdikten sonra kendi ortamlarına geri dönerek elde ettikleri bilgiyi sunarlar [8].
- **Bilgi Etmenleri:** İnternet gibi bir bilgi ağında kaynaklara etkin şekilde erişip kullanıcı isteklerine göre bilgi toplayan etmenlerdir. Birçok dağıtılmış kaynaktaki bilgiyi idare eder veya karşılaştırma yaparak okurlar. Bu etmenler sabit veya hareketli olabilir [8],[11].
- **Heterojen Etmenler:** En az bir etmen farklı türdedir. Tek başına çalışan uygulamaların iç işlemleri için istek üzerine çalışır. Uygulamalarla üç yolla birleşir. Uygulamayı tekrar yazar. Uygulamanın giriş ve çıkışını yakalar, içlerine kod sokarak onları paketler [11].

3.6. Etmenler ve Nesneler

Etmenler ve nesneler arasında açık benzerlikler bulunurken, önemli farklılıklar da vardır. İlki etmenlerin ve nesnelerin hallendirilme derecelerindedir.

Nesneler, çeşitli durumlarda olabilen, durumlara ilişkin metotları olan ve mesaj aktarma ile haberleşen bilgi işlemsel birimlerdir. Etmenleri nesnelerden ayıran en önemli özellikler:

- Etmenlerin ve nesnelerin özerklik dereceleri farklıdır: Nesnelerin metotları “public”, “protected” veya “private” olarak tanımlanabilmesine karşın diğerleri tarafından çağrılabilir. Kararı, çağırın nesne verir. Etmenlerde ise istekte bulunan etmen isterse isteği reddedebilir. Kararı, hizmete sahip olan etmen verir.
- Nesneler, tepki gösterme, sosyal olma gibi özelliklere sahip değildir.
- Nesneler, tek başına denetime sahip değildir ve içinde bulundukları sistemin denetimi altındadır. Nesneler arası mesajların anlamı nesneden nesneye değişmesine rağmen etmenler arasında değişmez [2].

3.7. Etmenler ve Uzman Sistemler

Uzman sistemler 1980'lerde yapay zekanın en önemli teknolojilerindendi. Uzman sistem problem çözme yeteneği olan veya bazı bilgi çokluğu olan alanlarda tavsiyelerde bulunur. MYCIN uzman sistemlerin klasik bir örneğidir. MYCIN insanlardaki kan enfeksiyonu tedavisinde yardımcı olmayı hedeflemişti. MYCIN bir takım olgularla (sembolik gösterimli) sistemi göstermek için bir kullanıcıyla etkileşen süreç yoluyla çalışıyordu, sistem sonra bazı sonuçlar çıkartmak için olguları kullanıyordu. MYCIN daha çok bir danışman gibi hareket etmekteydi; doğrudan insanlarla veya herhangi bir çevrede çalışmıyordu. Bu yüzden belki de etmenler ve MYCIN gibi uzman sistemler arasındaki en önemli farklılık budur, yani uzman sistemlerin herhangi bir çevreyle doğrudan etkileşimi bulunmaz: Bilgilerini algılayıcılar yoluyla toplamaz, fakat bir kullanıcı sayesinde bilgiyi alıp yarım insan gibi hareket edebilir. Aynı zamanda herhangi bir çevrede hareket etmezler fakat daha çok üçüncü bir kısma geri besleme veya tavsiye verir. Ek olarak, uzman sistemlerin genellikle diğer etmenlerle birlikte çalışma yeteneğinin olmasına ihtiyaç duyulmaz. Bu farklılıklara rağmen, bazı uzman sistemler, (özellikle gerçek-zamanlı kontrol görevlerini yapanlar), etmenlere çok fazla benzer. ARCHON sistemi buna güzel bir örnektir. ARCHON, çoklu-etmen sistemlerinin oluşturulması için bir yazılım platformu sağlar [2].

4. ZEKİ ETMENLER

4.1. Giriş

Zeki etmenler, dinamik olarak değişen bir çevreyle gerçek zaman kısıtlaması altında etkileşim gerektiren eşzamanlı görevleri yapabilen sistemlerdir [12]. Zeki etmen gerektiren görevlerden bazıları şunlardır: Enerji santrali kontrolü, deney kontrolü, eğitim, uçak pilotu yardımcılığı ve yoğun bakım hastası takibi.

Bu tür görevleri yapabilmesi için bir etmenin şu özelliklere sahip olması gereklidir:

- 1) Algılama - yani çevresinden bilgi toplayabilmesi için gerekli duyu verilerini alma ve yorumlama özelliği,
- 2) Düşünme - yani bilgi kullanarak durum değerlendirme, problem çözme ve davranış kararları alma özelliği,
- 3) Eylem - karar verilen davranışları yerine getirip çevreyi etkileyebilecek organları çalıştırabilme özelliği [12].

4.2. Zeki Etmenler İçin Soyut Mimariler

Şimdiye kadar anlatılmış olan etmenlerin soyut görünümünü çok basit olarak formüle edebiliriz [2]. İlk önce, karakterize edilebilen etmen çevre durumunu çevre durumları kümesi $S = \{s_1, s_2, \dots\}$ olarak kabul edelim. Verilen bir örnekte çevre bu durumlardan bir tanesi içinde farz edilsin. Bir etmenin çevresini etkileme yeteneği eylemler kümesi $A = \{a_1, a_2, \dots\}$ olarak gösterilsin. O halde soyut açıdan, etmene bir fonksiyon olarak şu şekilde bakılabilir:

$$eylem : S^* \rightarrow A$$

Çevre durumları serileri eylemlere bağlanır. Burada bir etmenin, standart bir etmen olarak fonksiyonla modellenmesinden bahsedilir. Bir etmenin geçmişine, geçmiş tecrübelerine dayanarak hangi davranışı yapacağına karar vermesine sezgi denir. Tecrübeler çevre durumlarının bir dizisi olarak gösterilir.

(Kararsız) çevrenin davranış fonksiyonu:

$$cev : S \times A \rightarrow \wp(S)$$

Çevrenin geçerli durumu $s \in S$ ve etmenin yaptığı eylem $a \in A$ çevre durumlarının kümesine $cev(s,a)$ bağlanırlar, s durumunda a eyleminin yapılmasıyla bu sonuç ortaya çıkabilir. Eğer cev 'in dizisindeki tüm kümeler tamamıyla tekse (yani herhangi bir durumda herhangi bir hareketin yapılmasının sonucunda kümede tek bir üye varsa), çevre kararlıdır ve çevrenin davranışı doğru olarak önceden bilinebilir.

Geçmiş, etmen ve çevre etkileşimi olarak gösterebiliriz. h geçmiş bir dizidir.

$$h : s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots \xrightarrow{a_{u-1}} s_u \xrightarrow{a_u} \dots$$

Burada s_0 çevrenin başlangıç durumudur (etmenin çalışmaya başladığı çevre durumu olabilir), a_u etmenin yapmak için seçtiği u . eylemidir ve s_u u . çevre durumudur (s_{u-1} durumundaki a_{u-1} eylemi olası çalışma sonuçlarından birisidir). Eğer $eylem : S^* \rightarrow A$ bir etmense $cev : S \times A \rightarrow \wp(S)$ de çevredir ve s_0 çevrenin başlangıç koşuludur. Aşağıdaki dizi

$$h : s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots \xrightarrow{a_{u-1}} s_u \xrightarrow{a_u} \dots$$

Aşağıdaki iki koşul geçerli olursa çevredeki mümkün olabilecek etmen geçmişi olarak gösterilebilir:

$$\forall u \in \mathbb{N}, a_u = eylem((s_0, s_1, \dots, s_u))$$

ve

$$\forall u \in \mathbb{N}, u > 0, s_u = cev(s_{u-1}, a_{u-1})$$

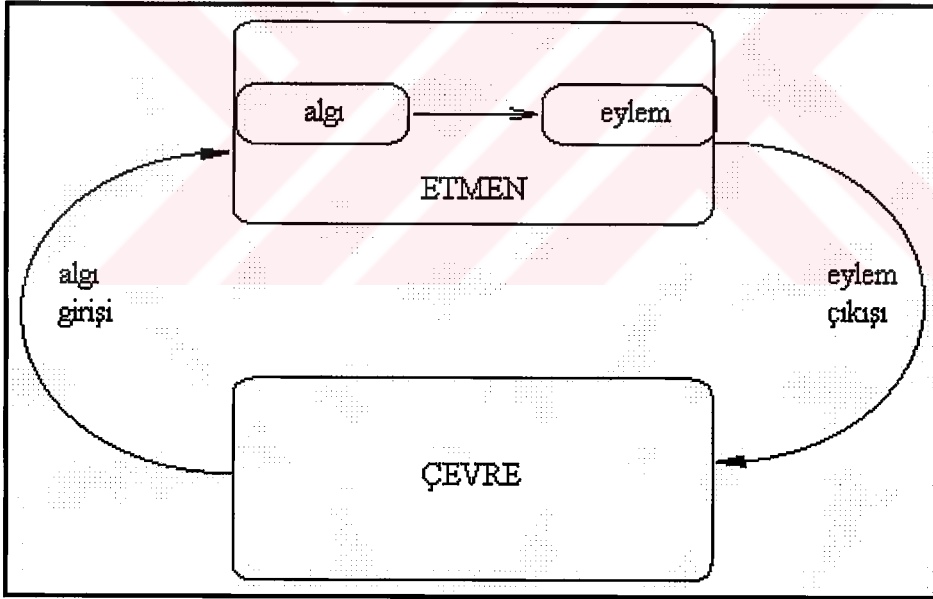
$cev : S \times A \rightarrow \wp(S)$ çevresindeki $eylem : S^* \rightarrow A$ etmeninin karakteristik davranışı bu özellikleri sağlayan tüm geçmişlerin kümesidir. Eğer bazı ϕ gibi özellikler tüm geçmişlerde sağlanıyorsa, bu özellik bu çevredeki etmenin değişmez özelliği sayılır. Örneğin etmenimiz nükleer reaktörü kontrol ediyorsa, (nükleer reaktör çevredir) kontrolcü / reaktörün tüm olası geçmişlerinde "reaktör infilak etmez", istenilen özelliği değişmez özellik olarak sayılabilir. Çevredeki tüm etmen geçmişlerinin kümesini $h(etmen, çevre)$ olarak gösterelim. Eğer $h(etm_1, cev) = h(etm_2, cev)$ ise, etm_1 ve etm_2 etmenlerinin cev çevresiyle uygun davranış eşitliğinden söz edilebilir ve eğer etmenler tüm çevrelerde uygun davranış olarak eşitse sadece davranış olarak eşitlikten söz edilir.

Genellikle, çevresiyle etkileşimini sonlandırmayan etmenlerle ilgilenilir, bu etmenler sonlanmazlar. Bu tür durumlarda, geçmişler sonsuz olarak düşünülür [2].

4.2.1. Algılama

Soyut düzeyde etmenlere bakmak sadece tahmin etme analizidir. Fakat etmenleri oluşturmak için bize yardımcı olmaz. Çünkü eyleme karar veren fonksiyonu nasıl tasarlayacağımız hakkında hiç ipucu vermez. Bu sebepten standart yazılım mühendisliğinde yapıldığı gibi etmenlerin soyut modeli alt sistemlere bölünerek geliştirilmelidir.

Etmenlerin görünümünü geliştirirken, çoğunlukla alt sistemlerle ilgili, etmenin bilgi ve kontrol yapılarına göre tasarım seçimleri yapılır. Bir etmenin mimarisi temelde etmen doğasının bir taslağıdır. İşlemler bilgi yapılarının üzerinde çalışabilir ve bilgi yapılarının arasında kontrol akışı bulunur. Burada yüksek seviye tasarım kararları incelenecektir. Bunların ilkinde etmenin karar fonksiyonu Şekil 4.1'deki gibi algılama ve eylem alt sistemlerine bölünür.



Şekil 4. 1 Algılama ve Eylem Altsistemleri

algı fonksiyonundaki ana fikir, etmenin çevresini gözlemesi için bir yeteneğe sahip olmasıdır, eylem fonksiyonu etmenin karar verme işlemidir. *algı* fonksiyonu gerçek dünyada bulunan bir etmen olayında donanım içinde oluşturulabilir. Örneğin, hareketli bir robot üzerinde kızıl ötesi algılayıcı veya video kamera olabilir. Yazılım etmeni için algılayıcı, yazılım çevresiyle ilgili bilgileri alan sistem komutları olabilir, ls, finger gibi. *algı* fonksiyonunun çıktısı algıdır - algı girişi P' ye algılar kümesi (boş olmayan) diyelim.

O zaman *algı* fonksiyonu şu şekildedir:

$$algı: S \rightarrow P$$

Çevre durumları algılara bağlanır ve eylem fonksiyonu da şu şekilde olur:

$$eylem: P^* \rightarrow A$$

Algılar serisi eylemlere bağlanır.

Bu basit tanımlar etmenlerle ilgili bazı özellikleri ve algılamayı keşfetmemizi sağlar. Farz edelim iki tane çevre durumumuz var, $s_1 \in S$ ve $s_2 \in S$, $s_1 \neq s_2$, fakat $algı(s_1) = algı(s_2)$ olsun. Bu iki farklı çevre durumu aynı algıya bağlanıyor. Bu yüzden etmen farklı çevre durumlarında aynı algı bilgisini alacaktır. O zaman etmenle ilgili olan s_1 ve s_2 ayırt edilemezdir. Bu örneği somutlaştırmak için termostat örneği verelim. x şu durumu gösterson

“Oda sıcaklığı NORMAL”

ve y de şu durumu gösterson

“John Major Başbakanıdır.”

Eğer çevremiz hakkında iki olgumuz varsa, çevre durumlarının kümesi S sadece dört eleman içerir:

$$S = \{\underbrace{\{-x, \neg y\}}_{s_1}, \underbrace{\{\neg x, y\}}_{s_2}, \underbrace{\{x, \neg y\}}_{s_3}, \underbrace{\{x, y\}}_{s_4}\}$$

Durum s_1 de oda sıcaklığı NORMAL değil ve John Major Başbakan değil; s_2 de oda sıcaklığı NORMAL değil ve John Major Başbakanıdır. Bizim termostatımız sadece odanın sıcaklığına duyarlıdır. Oda sıcaklığı da sonuç olarak John Major’un Başbakan olup olmamasıyla ilgili değildir. Böylece buradaki John Major Başbakanıdır veya değildir durumları tam olarak termostat için ayırt edilemezdir. Biçimsel olarak, termostat için *algı* fonksiyonu kendi alanı içinde iki algıya sahip olacaktır, p_1 ve p_2 , biri çok soğuğu diğeri NORMAL belirtecektir. Termostat için *algı* fonksiyonu aşağıdaki gibi davranacaktır:

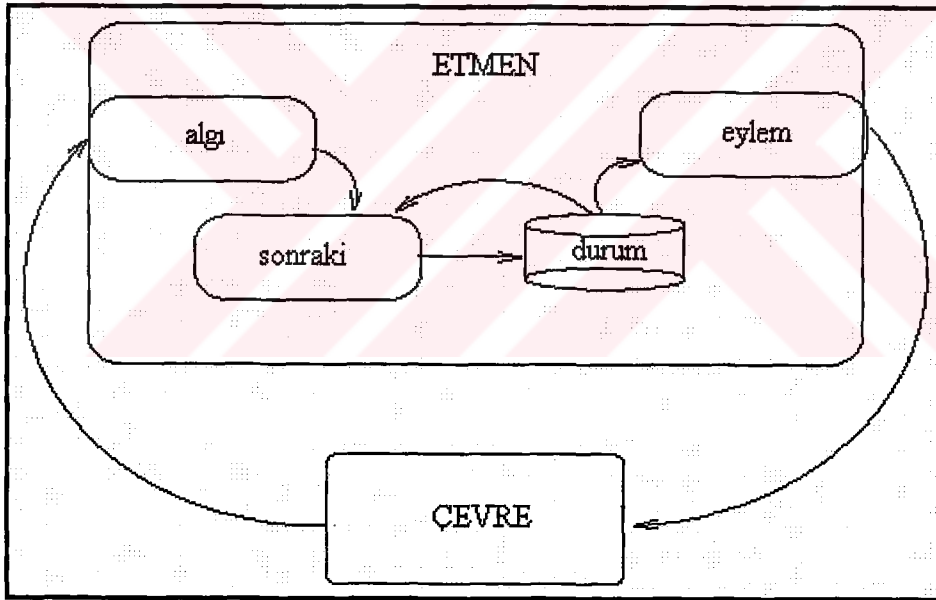
$$algı(s) = \begin{cases} p_1 & s = s_1 \text{ veya } s = s_2 \text{ ise} \\ p_2 & s = s_3 \text{ veya } s = s_4 \text{ ise} \end{cases}$$

Verilen $s \in S$ ve $s' \in S$ iki çevresel durumda, eğer $algı(s) = algı(s')$ ise, $s \equiv s'$ yazılır. Çevresel durumlar üzerindeki eşit ilişkiyi “ $\equiv$ ”sembolü gösterir. Karşılıklı

olarak ayırt edilemez durum kümesinin içinde S 'nin parçalarıdır. Sezgisel kaba eşitlik sınıfları etmenin algılamasını az etkiler. Eğer $|| = |S|$ ise, (yani, farklı algıların sayısı farklı çevre durumlarının sayısına eşit ise), etmen her durumu fark edebilir. Çevredeki etmen en mükemmel algılamaya sahiptir; o her şeyi bilir. Bunun tersinde eğer $|| = 1$ ise, etmen algılama yeteneğine sahip değildir, farklı durumlar arasındaki değişikliği ayırt edemez. Bu durumda etmen tüm çevresel koşulları aynı kabul eder.

4.2.2. Durumlu Etmen

Şimdiye kadarki gösterimlerde sezgiyle ilgili bir şey bulunmamaktadır. Durumlu etmen şemasında ise aşağıda anlatılan ek bir eşitlikten bahsedilmektedir. Buradaki fikir etmen durumlarının korunmasının düşünülmesidir. Bu tip etmenlerde bir çeşit iç bilgi yapısı bulunur, genellikle çevre durumu ve geçmiş hakkındaki bilgileri kayıt eder. (Şekil 4.2'ye bakın).



Şekil 4. 2 Durumunu Koruyan Etmenler

I 'ya etmenin iç tüm durumlarının kümesi denilsin. Etmenin karar alma mekanizması bu parçanın üzerindeki bilgiye dayanır. Durum tabanlı etmenlerde *algı* algılama fonksiyonu değişmez.

$$algı : S \rightarrow P$$

Hareket seçme fonksiyonu eylem şu şekilde tanımlanacaktır:

$$eylem : I \rightarrow A$$

Ek fonksiyon *sonraki* bir iç durumu ve algıyı bir iç duruma bağlar.

$$sonraki : I \times P \rightarrow I$$

Durum tabanlı etmenin davranışı şu şekilde özetlenebilir. Etmen başlangıç iç durum i_0 ile başlar. Sonra s çevre durumunu gözler ve $algı(s)$ algısını üretir. Sonrada iç durumunu $sonraki(i_0, algı(s))$ olabilecek şekilde *sonraki* fonksiyonuyla günceller. Etmen tarafından $eylem(sonraki(i_0, algı(s)))$ eylemi seçilir. Bu davranış etmen tarafından yapılır ve etmen diğer döngüye girer. *algı* yoluyla dünyayı algılar, *sonraki* ile durumlarını günceller ve *eylem* ile bir eylem gerçekleştirir.

4.3. Zeki Etmenler İçin Somut Mimariler

Bu kısımda eylem fonksiyonunun nasıl oluşturulacağı açıklanır. Bu işlem için dört sınıf etmen dikkate alınacaktır:

- Mantık tabanlı etmenler - mantıksal çıkarımlarla karar alma gerçekleştirilir.
- Tepkili etmenler - koşullardan eyleme doğrudan bağlanma biçimleriyle karar alma oluşturulur.
- İnanç - istek - amaç etmenleri - karar verme etmenin inanç, istek ve amaçlarının gösterildiği bilgi yapılarının değiştirilmesine dayandırılarak oluşturulur.
- Katmanlı mimariler - karar verme çeşitli yazılım katmanları yoluyla gerçekleştirilir, katmanların her biri farklı soyutlama düzeyinde bulunan çevre hakkında çok veya az belirgin olarak mantıksaldır [11].

4.3.1. Mantık Tabanlı Mimariler

Yapay zeki sistemlerin oluşturulmasıyla ilgili geleneksel yaklaşım (sembolik yapay zeka), verilen sistem çevresi, istenilen davranışın sembolik gösterimiyle ve bu gösterimin sözdizimsel değişimiyle sistemde zeki davranışın üretilmesini önerir. Sembolik gösterimler, mantıksal formül ve sözdizimsel değişiklik, mantıksal çıkarım veya teorem ispatlamaya karşılık gelir.

Teorem ispatlama olarak etmen fikri çekicidir. Farz edelim ki, zeki bir etmenin nasıl davranacağını açıklayan bir teori var. Bu teori, örneğin, bir etmenin, tasarım amaçlarını sağlayacak aynı amaçları nasıl üreteceğini, bu amaçları gerçekleştirmesi için amaç doğrultulu ve tepkili davranışı nasıl içine ekleyeceğini ve bunun gibi şeyleri açıklayabilsin. Sonra bu ϕ teorisi, etmenin nasıl davranacağını açıklayan bir tarif olarak düşünölsün. Geleneksel yaklaşım, sonuç olacak bir uygulamaya ulaşana

kadar kademeli olarak daha somut aşamaların bir serisi sayesinde tarifi düzeltilmesini gerektiren, yerine getirilecek bir sistem uygular. Fakat teorem ispatlayıcı olarak etmenlerin görüntüsünde bu türde bir düzeltme bulunmaz. Bunun yerine ϕ çalışabilir tarif olarak görülür; etmenin davranışını üretebilmesi için doğrudan çalışır.

Bu fikrin nasıl çalıştığını görmek için, mantık tabanlı etmenlerin basit bir modeli geliştirilir ve tasarlanmış etmenler olarak adlandırılır. Bu tür etmenlerde, iç durum olarak klasik ilk-sıra yüklem mantığı (first-order predicate logic) formül veritabanı olduğunu farz edelim. Örneğin, etmenin veritabanında şu tür formüller olacaktır:

Açık(vana221)

Sıcaklık(reaktör4726, 321)

Basınç(tank776, 28)

Bazı çevre özelliklerini göstermek için formüllerin ne kadar kullanışlı olduğunu görmek çok zor değildir. Veritabanı etmenin çevresiyle ilgili bilgileridir. Etmenin veritabanı biraz insanlardaki inançla aynı rolü oynar. Bir kişi vana 221'in açık olduğu inancına sahip olabilir - etmen veritabanında Açık(vana221) yüklemi olabilir. Doğal olarak, insanlar gibi etmenler de hata yapabilir. Vana 221 gerçekte kapalıyken, ben vana 221'in açık olduğuna inanabilirim; etmenin veritabanında Açık(vana221) olması vana 221'in (veya aslında herhangi bir vananın) açık olması anlamına gelmeyebilir. Etmenin algılayıcıları bozuk olabilir, bu bozulmanın sonucu olarak, bilginin tarihi geçmiş olabilir veya etmen tasarımcısının istediği Açık(vana221) formülünün yorumu tamamıyla farklı bir şey olabilir.

L klasik ilk-sıra mantık cümlelerinin kümesi olsun ve $D = \wp(L)$, L veritabanlarının kümesi olsun, yani, L formülü kümelerinin kümesi. O zaman etmenin iç durumu D 'nin bir elemanı olur. D 'nin elemanları $\Delta, \Delta_1, \dots$ olarak gösterilsin. Etmenin karar alma işlemi çıkarım kuralları kümesi ρ sayesinde modellenir. Bunlar basit mantıksal çıkarım kurallarıdır. Eğer çıkarım kuralı ρ kullanılarak veritabanı Δ 'dan formül ϕ sağlanabiliyorsa $\Delta \vdash_{\rho} \phi$ yazarız. Etmenin algılama fonksiyonu değişmeden kalır:

$$alg_1 : S \rightarrow P$$

Aynı şekilde *sonraki* fonksiyonunun biçimi şudur:

$$sonraki : D \times P \rightarrow D$$

Bunun için bir veritabanı ve bir algı yeni bir veritabanına bağlanır, fakat etmenin eylem seçim fonksiyonu tekdir.

$$eylem : D \rightarrow A$$

eylem çıkarım kuralları terimleriyle tanımlanır. Bu fonksiyonun kod tanımı olarak aşağıdakiler yazılabilir.

Tablo 4. 1 Mantık Tabanlı Mimaride Eylem Kodu

1.	function <i>eylem</i> ($\Delta : D$) : <i>A</i>
2.	begin
3.	for each $a \in A$ do
4.	if $\Delta \vdash_{\rho} Do(a)$ then
5.	return a
6.	enf-if
7.	end-for
8.	for each $a \in A$ do
9.	if $\Delta \forall_{\rho} \neg Do(a)$ then
10.	return a
11.	enf-if
12.	end-for
13.	return <i>null</i>
14.	end function <i>eylem</i>

Etmen tasarımcısının düşüncesi şudur; eğer $Do(a)$ formülü sağlanabilirse, her türlü çıkarım kuralları ρ ve Δ veritabanını etmen tasarımcısı kodlayacaktır, buradaki a terimi eylemi gösterir ve a eylemi gerçekleştirmek için en iyi eylemdir. Böylece fonksiyonun ilk kısmında (1. - 7. Satırlar), etmen döngü içinde mümkün olabilecek her a eylemini yapar ve ρ çıkarım kurallarını kullanarak veritabanından (bir parametre olarak fonksiyona geçirilir) formül $Do(a)$ yı türetmeyi dener. Etmen $Do(a)$ 'nın türetilmesinde başarılı olursa, eylemin yapılması için a 'yı döndürür.

Tüm $a \in A$ eylemleri için etmen $Do(a)$ 'nın sağlanmasında başarısız olursa ne olur? Bu durumda, kurallarla ve veritabanı ile tutarlı, açıkça yasaklanmamış bir eylemi bulmak için etmen deneme yapar. (8.) - (12.) satırlarda bu yüzden etmen, çıkarım kurallarını kullanarak veritabanından türetemeyeceği $\neg Do(a)$ türündeki $a \in A$ eylemini bulmaya kalkışır. Bu türde bir eylem bulursa, gerçekleştirilecek eylem olarak bunu döndürür. Fakat en az tutarlılıkta bile bir eylem bulmada başarısız olursa, herhangi bir eylemin seçilmediğini belirten özel eylem *null* (boş)'u döndürür.

Bu yolla, etmenin davranışı, etmenin çıkarım kuralları programıyla ve çevresiyle ilgili bilgileri gösteren, geçerli veritabanı tarafından belirlenir.

Etmenimizin çalışmasını düşünelim. t_1 zamanında etmen Δ_1 veritabanını üretir ve hangi eylemi yapacağını bulmak için ρ kurallarını uygulamaya başlar. Bir süre sonra t_2 zamanında $a \in A$ için $\Delta_1 \vdash_\rho Do(a)$ 'yı gerçekleştirmeye çalışır ve böylece a , t_1 zamanında etmenin yapacağı en iyi eylem olur. Fakat, t_1 ve t_2 arasında çevre değişirse, a 'nın en iyi eylem olarak kalacağını bir garantisi yoktur. Özellikle bu arada geçen zaman uzunsa eylem optimalden uzak olacaktır. Eğer $t_2 - t_1$ sonsuz küçük ise - şu demektir, karar alma etkin olarak ani olursa- problemi güvenli bir şekilde ihmal edebiliriz.

Özetlersek, mantıksal tabanlı yaklaşımlarla etmenleri oluşturmak için karar almaya sonuç çıkarma olarak bakılabilir. Bir etmenin programı -yani karar alma stratejisi- mantıksal teoriyle kodlanır ve problemin kanıtını azaltacak bir eylem seçim işlemi kodlanır. Mantıksal tabanlı yaklaşımlar çok iyidir fakat çok fazla avantajlı değildir. Özellikle, teorem ispatlamanın kalıtım hesaplama karmaşıklığı etmenlerin zaman sınırlı çevrelerde teorem ispatlamanın etkili biçimde çalışıp çalışmadığının sorgulanmasını gerektirir. Bu türdeki etmenlerde karar alma uygun mantık sanıları üzerine yüklenilmiştir - etmen ne yapacağına karar verirken, dünyanın herhangi bir yolla önemle değişmediği farz edilir ve karar alma başladığında mantıklı olabilecek bir sonuç doğurduğunda eylem mantıklı kabul edilir. Ayrıca karmaşık, dinamik ve fiziksel çevrelerle ilgili gösterim ve sonuç çıkarımla birlikte birleştirilmiş problemler çözülememiştir.

Az veya çok sayıda "sade" mantıksal yaklaşımlarla etmen uygulamaları geliştirilmiştir. Bunlardan en çok bilinenleri CONGOLOG sistemi ve METATEM'dir. CONGOLOG sisteminin temeli durum hesabına dayanır. METATEM, zamanı göz önünde bulundurur ve yerine getirilecek mantıklı etmen davranışlarının ayrıntılarını vererek etmenleri programlar [2].

4.3.2. Tepkili Mimariler

Sembolik / mantıksal yaklaşımlarla etmen oluşturmak için çıkan zorlu problemler bazı araştırmacıların kafasında soru işaretleri bırakmıştır ve en sonunda temeli bu tür yaklaşımlara dayanan sanıları değiştirmeyi düşünmüşlerdir. Bu araştırmacılar, mantıksal gösterim dilinin zayıflığı gibi, sembolik yaklaşımla ilgili küçük değişiklikleri tartıştılar, ama zaman-sınırlı çevrelerde çalışan etmenleri oluşturmak için küçük değişikliklerin yeterli olmayacağına karar verdiler; küçük olmayacak tam, yeni bir yaklaşıma ihtiyaçları vardı. Farklı yaklaşımları açıkça tanımlamak oldukça zordu. Çünkü araştırmacılar yeni yaklaşımı tanımlarken sembolik yapay zekayı

kullanmama kararında birleşmişti. Bununla birlikte bazı temalar bu yaklaşımda yeniden gözden geçirilecekti:

- Sembolik gösterimlerin ve bu tür gösterimsel söz dizim değişikliğine dayanan karar almayı kabul etmemek
- Bir etmenin bulunduğu çevresiyle doğal bağlantısı zeki, mantıklı davranıştır, fikri kabul edilir. Zeki davranış etmenin çevresiyle sürdürdüğü etkileşimin bir ürünüdür.
- Zeki davranış çeşitli basit davranışların etkileşiminden ortaya çıkar görüşü vardır.

Alternatif yaklaşımlar, etmen için davranışsal, yerleşmiş ve tepkiseldir diye bahseder. Etmen davranışsaldır, çünkü buradaki temel konu özel davranışların geliştirilmesi ve birleştirilmesidir. Etmen yerleşmiştir, yani etmenler çevrelerinin içine yerleşmişlerdir. Son olarak etmen tepkiseldir, yani bazı sistemler çevreden neden sonuç almadan, çevreye basitçe tepki verir ve sürekli olarak algılama yapar. Bu konu subsumption mimarisinde incelenir, Bu tartışmasız en iyi bilinen tepkili etmen mimarisidir. Rodney Brooks tarafından geliştirilmiştir [2].

Subsumption mimarisinin belirtilmiş iki tanımı bulunur. İlki, etmenin karar alması, görev alan davranışların bir kümesi sayesinde gerçekleştirilir; her davranış özel eylem fonksiyonu olarak düşünülebilir, sürekli olarak algı girişi alır ve bu girişi gerçekleştirmek için bir eyleme bağlar. Her davranış modülü bazı özel görevleri almak için istekte bulunur. Brooks'un uygulamasında, davranış modülleri sonlu durum makineleridir (finite state machines). Burada dikkat edilmesi önemli nokta görev alan modüllerin karmaşık sembolik gösterimlerinin bulunmadığı ve tamamıyla sembolik nedenler çıkarmadığı farz edilir. Birçok uygulamada bu davranışlar kural biçimli uygulanır.

durum → eylem

Basitçe algı girişi eyleme doğrudan bağlanır.

Subsumption yapısının ikinci tanımına göre, birçok davranış eşzamanlı tetiklenebilir. Çoklu eylemler tarafından ayrılmış, farklı eylemler arasından seçim yapmak için açık bir mekanizmanın olması gerekir. Brooks, katmanlar içinde davranışlarla birlikte sıralanan, subsumption hiyerarşisi içinde modüllerin sıralanmasını tasarlamıştı. Hiyerarşideki en düşük katmanlar en yüksek katmanları kısıtlayabilir: daha düşük tabakanın daha yükseğe göre önceliği vardır. Buradaki düşünce daha yüksek tabakaların daha soyut davranışlar göstermesidir. Örneğin, “engellerden kaçma”

davranışı için hareketli bir robottaki bu davranışın amacı bir tane olabilir. Engellerden kaçmaya yüksek bir öncelik vermek anlaşılır olacaktır - bundan dolayı bu davranış doğal olarak yüksek önceliğe sahip, düşük düzey katman içinde kodlanacaktır. Daha detaylı olarak subsumption yapısını açıklamak için basit bir biçim gösterelim.

Etmenin algılama yeteneği olarak gösterilen *algı* fonksiyonunun değişmeden kaldığı farz edilir. Fakat subsumption mimarisi uygulanmış sistemlerde, algı ve eylem arasında tam sıkı bir bağlantı olduğu farz edilir.

Karar fonksiyonu eylem, davranışlar arasında tutulan kısıtlama bağıntısıyla beraber, davranışların kümesiyle oluşturulur. Bir davranış (c, a) çiftidir, buradaki $c \subseteq P$ koşul olarak adlandırılan algıların kümesidir ve $a \in A$ bir eylemdir. Çevre $s \in S$ eğer $algı(s) \in c$ durumundaysa davranış (c, a) tetiklenecektir.

$Beh = \{(c, a) \mid c \subseteq P \text{ ve } a \in A\}$ tüm kuralların kümesidir.

Etmenin $R \subseteq Beh$ davranış kuralları kümesiyle birleştirilmiş davranışların kümesi üzerindeki ikili kısıtlama bağıntısı şudur: $\prec \subseteq R \times R$. Bu R üzerinde sıralanmış toplam olarak farz edilir (geçişlidir, dönüşsüzdür ve asimetriktir). Eğer $(b_1, b_2) \in \prec$ ise $b_1 \prec b_2$ yazılır ve b_1, b_2 'yi kısıtlar diye okunur. Yani, hiyerarşide b_1, b_2 'den daha düşük ve bu yüzden önceliği b_2 'nin üzerindedir. Eylem fonksiyonu da aşağıdaki gibi tanımlanır:

Tablo 4. 2 Tepkili Etmen Mimarisinde Eylem Kodu

```

1. function eylem( $p : P$ ) :  $A$ 
2. var fired :  $\wp(R)$ 
3. var selected :  $A$ 
4. begin
5.   fired :=  $\{(c, a) \mid (c, a) \in R \text{ ve } p \in c\}$ 
6.   for each  $(c, a) \in \textit{fired}$  do
7.     if  $\neg(\exists(c', a') \in \textit{fired} \text{ such that } (c', a') \prec (c, a))$  then
8.       return  $a$ 
9.     end-if
10.  end for
11.  return null
12. end function action

```

Eylem seçimi, önce (5.) satırdaki *fire* ile tüm tetiklenmiş davranışların kümesinin hesaplanmasıyla başlar. Sonra, tetiklenen her bir (c, a) davranışı kontrol edilir, daha yüksek öncelikli diğer tetiklenecek davranışın olup olmadığına karar vermek için kontrol işlemi yapılır. Eğer tetiklenme yoksa, davranışın eylem parçası olan a ,

seçilmiş eylem olarak döndürülür (8.). Eğer hiçbir davranış tetiklenmezse, hiçbir eylemin seçilmediğini belirten ayırt edilebilen eylem *null* döndürülecektir.

Subsumption eylem fonksiyonunun tam zaman güçlüğüne hata olasılığı çok kötü değildir, böylece yukarıdaki algoritmaya bile karar alma işlenebilir. Uygulamada bundan daha iyisini yapabiliriz: verilen sabit karar zamanıyla donanım içine karar alma mantığı kodlanabilir. Modern donanım için bu şu anlama gelir, nano saniye içinde etmenin bir eylemi seçmesi garanti altına alınabilir. Bu hesaplama basitliği subsumption mimarinin gücüdür.

Özetlersek, Brooks'un subsumption mimarisi türündeki tepkili yaklaşımların açık avantajları vardır; basitlik, ekonomik, hesaplanabilir işlenebilirlik, hatalara karşı sağlamlık ve tüm bu seçkinliği bu türdeki yapıları cazip yapar. Fakat bazı temel, çözülememiş problemler, sadece subsumption mimaride değil, diğer sadece tepkili olan mimarilerde de bulunur:

- Etmenler çevrelerinin modelini kullanmazsa, uygun eylem kararı almak için etmenler yerel çevredeki mevcut yeterli bilgiye sahip olmak zorundadır.
- Kararları, genel bilgiye dayanan sadece tepkili olan etmenler, (yani etmenlerin geçerli durumu hakkındaki bilgi), etmen yerel olmayan bilgiyi işlemede nasıl bir karar alacağını görmede zorlanır.
- Sadece tepkili olan etmenlerde deneyimlerden öğrenme ve zamanla performans artırmanın nasıl tasarlanacağını görmek zordur.
- Tepkili sistemlerin en çok beğenilen özelliği, etmen çevresine yerleştirildiğinde, tamamlayıcı davranışların etkileşiminden tüm davranışın ortaya çıkmasıdır. Fakat, özel davranışların, çevrenin ve tüm davranışın arasındaki ilişkiyi akla getiren "ortaya çıkma" terimi anlaşılır değildir. Sonuç olarak bu tür etmenleri oluşturmak için prensipli bir yöntem bilim bulunmaz; bir etmeni yapmak için yorucu deney, deneme ve hata işleminden yararlanılmak zorundadır.
- Verimli etmenler az sayıda davranışlarla oluşturulabilirken (on katmandan daha az), çok fazla katman içeren etmenleri oluşturmak oldukça zordur. Farklı davranışlar arasındaki etkileşimlerin dinamiğini anlamak çok karmaşıktır.

Bu problemler için çeşitli çözümler önerilir. En popüler olanlardan birisi belirli görevleri yapan etmenlerin geliştirilmesi fikridir [2].

4.3.3. İnanç-İstek-Amaç Mimarileri

Bu kısımda inanç - istek - amaç (BDI) yapıları açıklanacaktır. Bu yapıların uygulamalı tümevarımın felsefi geleneğinde kökleri bulunur - amaçlarımızı ilerletmede yapılacak eyleme, an ve an, karar alma süreci.

Uygulamalı tümevarım iki tane önemli süreç içerir; gerçekleştirmek istediğimiz amaçların ne olduğuna ve bu amaçların nasıl gerçekleştirileceğine karar verme. İlk süreç, üzerinde düşünme, ikincisi sonuçları edinme yolları tümevarımı olarak bilinir. BDI modelinin anlaşılmasını sağlamak için, uygulamalı tümevarımın basit bir örneğinin açıklanması yeterlidir. Üniversiteyi iyi bir dereceyle bitirdiyseniz, bir seçeneğiniz akademisyen olmaktır (ama dereceniz kötüyse bu seçenek sizin için mevcut değildir). Diğer seçeneğiniz ise bir işe girmektir. Alternatiflerinizin bir kümesini oluşturduktan sonra bunlar arasında bir seçim yapmanız ve söz vermeniz gerekir. Tercih edilen bu seçenekler etmenin eylemine karar veren amaçlar olacaktır. Amaçlar tümevarım içindeki geri beslemedir. Örneğin akademisyen olmaya karar verdiğim, bu amacımı devam ettirmeliyim ve bunun için zaman ayırmalı ve gayret etmeliyim. Amaçlar uygulamalı tümevarımda çok önemli rol oynar.

- Amaçlar sonuçları edinme yolları tümevarımını çalıştırır.

Eğer akademisyen olma amacını oluşturduysam, amacımı gerçekleştirmeye kalkışacağım, örneğin bir yüksek lisans programına başvurarak, nasıl gerçekleştireceğime karar vereceğim. Üstelik eğer amacı gerçekleştirirken eylemin belirli bir kurunda başarısız olursam, doğal olarak diğerlerini deneyeceğim. Başka bir üniversitenin programına başvuracağım.

- Amaçlar gelecek zaman üzerinde düşünmeyi kısıtlar

Akademisyen olmayı amaçladıysam, bu amaca ters düşen şeylerle vakit kaybetmeyeceğim. Örneğin mantıklı bir etmen, akademisyen olmayı amaçlarken aynı zamanda zengin olmakla ilgilenmeyecektir. (İki amaç tam olarak ayrıcalıklı değildir, ama her ikisinin de aynı zamanda başarıma olasılığı sonsuz küçüktür.)

- Amaçlar sürekli dir.

Genellikle iyi bir sebep ortada yoksa amacımdan vazgeçmem - sürekli dirler, başarılı bir şekilde gerçekleştireceğime veya tam tersine inanıncaya kadar veya başka bir şey olmazsa, amacın amacı ortada kalmayıncaya kadar devam ederler.

- Amaçlar gelecekle ilgili, temeli uygulamalı tümevarıma dayanan davranışlardan etkilenir.

Eğer akademisyen olma amacını kabul ettiysem, gelecekle ilgili planlarımı akademisyen olacağım sanısı üzerine yapabilirim.

Uygulama tümevarım etmenlerinin tasarımındaki ana problem bu farklı bağlantılar arasında iyi bir denge kurulmasıdır. Özellikle etmenin zamanla bazı amaçlarından vazgeçmesinde bu açıkça görülür (çünkü etmen amaçları başarabileceğine, tam tersine veya artık sahip olduğu amacın nedeninin kalmadığı inancına ulaşmıştır). Bu durum zaman zaman etmenin amaçlarının nedenlerini tekrar düşünmesi için durmasında önemlidir. Fakat tekrar düşünme pahalı bir işlemdir - hem zaman hem de işleyen kaynaklar için. Ama bu bize bir ikilemi gösterir:

- Yeterli sıklıkta tekrar düşünmek için durmayan etmen, amaçlarını açıkça gerçekleştiremeyecekse veya gerçekleştirmesi için artık bir sebep ortada kalmamışsa bile amaçlarını başarmaya çalışabilir.
- İlgiğini sık sık düşünen etmen amaçlarını gerçekleştirmek için yetersiz çalışma zamanı geçirebilir ve bu yüzden amaçlarını hiç başaramama riski taşır.

Aslında bu ikilem önetkin ve tepkili davranışın dengelenmesi problemidir.

David Kinny ve Michael Georgeff, çeşitli farklı çevrelerde çalışan cesur etmenleri (tekrar düşünmek için durmazlar) ve dikkatli etmenleri (sık sık tekrar düşünmek için dururlar) araştırmışlardır. Yaptıkları deneylerdeki en önemli parametre γ dünyanın değişim oranıdır. Çıkardıkları anahtar sonuçlar şunlardır:

- Eğer γ düşükse (yani çevre hızlı değişmiyorsa), cesurlar dikkatli etmenlerle karşılaştırıldığında daha iyidir, çünkü, cesurlar işleriyle uğraşırken ve amaçlarını gerçekleştirirken, dikkatli olanlar tekrar düşünmeyle zaman kaybederler.
- Eğer γ yüksekse (yani çevre sık sık değişiyorsa), dikkatli etmenler cesurlardan üstün olur, çünkü, amaçlara farklı hüküm verildiği zaman farkına varabilirler ve üstelik rastlantısal durumlarda ve yeni fırsatlarda avantaj sağlarlar.

Farklı türdeki çevreler farklı türde karar alma stratejilerine gerek duyar. Statik, değişmeyen çevrede sadece önetkin olan, amaca yönelik davranış yeterlidir. Fakat daha dinamik çevrelerde, amaçlarda düzenleme yaparak değişikliklere tepki yaratabilme yeteneğinin olması daha önemlidir [2].

BDI etmenlerdeki uygulamalı tümevarımın işleyişi Şekil 4.3 'te gösterilmiştir. Bu şekilde olduğu gibi, BDI etmenleri için yedi ana bileşen vardır:

- Geçerli inanç kümesi, etmenin çevresi hakkındaki geçerli bilgiyi gösterir.
- İnanç düzeltme fonksiyonu, (*brf*), algı girişini ve etmenin geçerli girişlerini alır ve bunlara dayanarak inançların yeni kümesine karar alır.
- Seçenek üretme fonksiyonu, *seçenekler*, etmenin çevresiyle ilgili geçerli inanç ve amaçlarına dayanarak, etmene mevcut seçimler yaptırır.
- Geçerli seçenekler kümesi, etmen için mevcut olabilecek eylemlerin adımlarını gösterir.
- Filtre fonksiyonu, (*filtre*), etmenin üzerinde düşünme sürecini gösterir ve etmenin geçerli inançları, istekleri ve amaçlarına dayanarak etmenin amaçlarına karar verir.
- Geçerli amaçlar kümesi, etmenin geçerli odak noktasını gösterir - olayların durumlarıyla ilgili uğraşacağına dair söz verir.
- Eylem seçme fonksiyonu (*uygula*), geçerli amaca göre hangi eylemin yapılacağına karar verir.

Bu bileşenleri biçimsel tanımlamak kolaydır. Önce, olabilecek tüm inançların kümesi *Bel*, olabilecek tüm istekler kümesi *Des*, olabilecek tüm amaçlar kümesi *Int* olsun. Bu konuda kümelerin içeriği önemli değildir. (Çoğunlukla bunlar mantıksal formülle ilk-sıra mantıkla gösterilirler). Herhangi bir anda verilen BDI etmeninin durumu (B, D, I) üçlüsüdür, burada $B \subseteq Bel$, $D \subseteq Des$ ve $I \subseteq Int$.

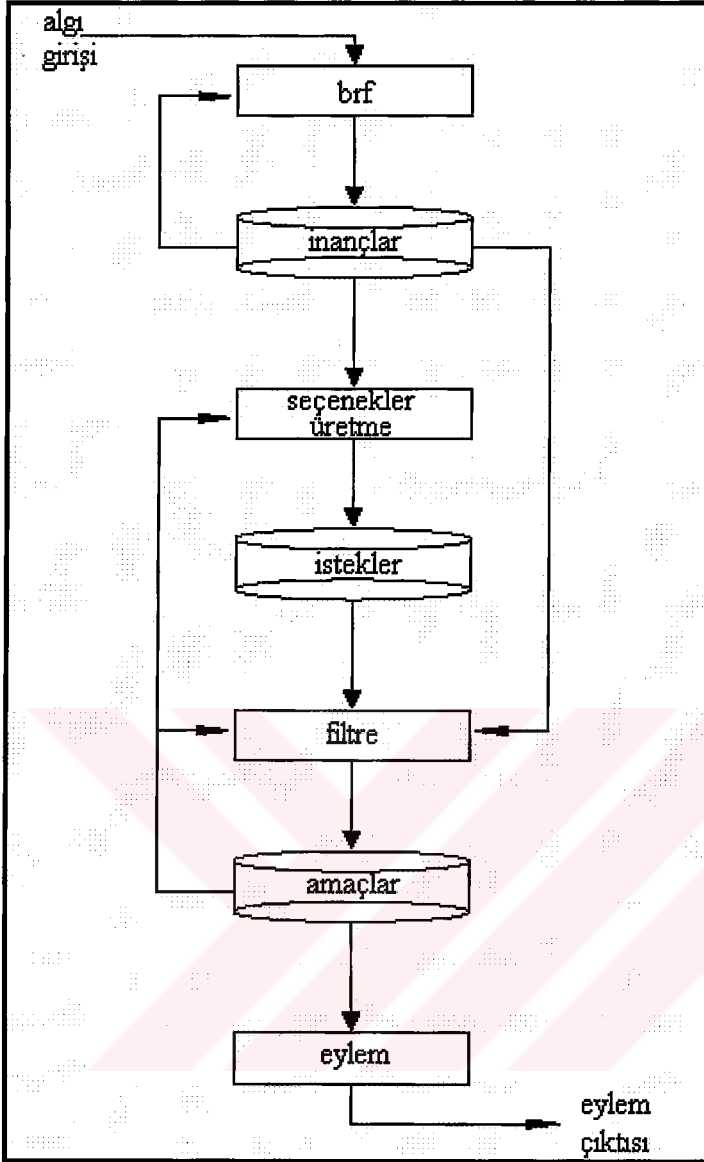
Etmenin inanç düzeltme fonksiyonu şuna bağlanır:

$$brf : \wp(Bel) \times P \rightarrow \wp(Bel)$$

Geçerli algı ve geçerli inançlara dayanarak yeni inançların kümesine karar verilir.

Seçenek üretme fonksiyonu, *seçenekler*, inançların ve amaçların kümesini isteklerin kümesine bağlar.

$$seçenekler : \wp(Bel) \times \wp(Int) \rightarrow \wp(Des)$$



Şekil 4. 3 İnanç-İstek-Amaç Mimarisinin Genel Şematik Çizimi

Bu fonksiyon birbirinden farklı roller oynar. Önce, etmenin sonuçlar edinme yolları tümevarımından (amaçların nasıl gerçekleşeceğine karar verilmesinden sorumlu süreç) olmak zorundadır. Etmen önce x 'e amaç oluşturur, sonra x 'i gerçekleştirmek için seçenekler düşünmek zorundadır. Bu seçenekler x 'den daha somut - daha az soyut - olacaktır. Bu seçeneklerden bazıları kendilerini amaç yaparak, seçenek üretmede geri besleme olacaklardır, sonuçta daha somut seçenekler üretilecektir. Hızlı çalışan eylemlere uyan amaçlara etmen ulaşınca kadar, BDI etmenlerinin seçenek üretim yöntemi hiyerarşik bir plan yapısını tekrarlı olarak ayrıntılara inme, düşünme ve kademeli olarak daha özel amaçlar için söz verme olarak düşünebiliriz.

Seçenek fonksiyonunun yukarıda belirtilen ana amacından başka, ek olarak, birçok farklı diğer sınırlamaları da sağlar. İlk önce tutarlı olmak zorundadır, çevresel

koşullar avantajlı şekilde değiştiğinde fark edebilmeli, amaçlarını gerçekleştirmek için etmene yeni yollar önerebilir veya amaçlarının gerçekleştirme olasılığı tam tersi başarısızsa farklı yeni yollar önerir.

BDI etmeninin üzerinde düşünme süreci (ne yapacağına karar verme) filtre fonksiyonu içinde gösterilir,

$$filtre : \wp(Bel) \times \wp(Des) \times \wp(Int) \rightarrow \wp(Int)$$

Önceden alınmış amaçlar, geçerli inançlar ve isteklerin temeli üzerine dayanan etmenlerin amaçlarını yeniler. Bu fonksiyon iki rolü yerine getirmek zorundadır. İlki, artık gerçekleştirilmesine ihtiyaç duyulmayan veya başarılı bir şekilde gerçekleştirilenlere benzetildiğinde beklenen kazanım süresini aşan, gerçekleştirilmesi pahalı umulan amaçları çıkarmak zorundadır. İkinci olarak, henüz gerçekleştirilmemiş eylemleri tutabilir ve amaçların pozitif yararının bulunacağı zamanı bekleyebilir. Sonuç olarak, ya varolan amaçları gerçekleştirmek yada yeni çıkan fırsatları kullanmak için yeni amaçlar kabul eder.

Bu fonksiyonun hiçbir yerde amaçlarını açıklanması beklenmez. *filtre* fonksiyonu aşağıdaki sınırı sağlayabilir:

$$\forall B \in \wp(Bel), \forall D \in \wp(Des), \forall I \in \wp(Int), filtre(B, D, I) \subseteq I \cup D$$

Diğer bir deyişle, geçerli amaçlar önceden alınmış amaçlar veya kabul edilmiş seçeneklerdir.

uygula fonksiyonu basit olarak yerine getirilebilen amaçlar farz edilir - doğrudan yerine getirilebilen eyleme benzer:

$$uygula : \wp(Int) \rightarrow A$$

Etmenin karar alma fonksiyonu, BDI etmeninin eylem fonksiyonudur.

$$eylem : P \rightarrow A$$

Aşağıdaki kodla tanımlanır:

Tablo 4.3 BDI Mimarisinde Eylem Kodu

1. function <i>eylem</i> (<i>p</i> : <i>P</i>) : <i>A</i>
2. begin
3. <i>B</i> := <i>brf</i> (<i>B</i> , <i>p</i>)
4. <i>D</i> := <i>seçenekler</i> (<i>D</i> , <i>I</i>)
5. <i>I</i> := <i>filtre</i> (<i>B</i> , <i>D</i> , <i>I</i>)
6. return <i>uygula</i> (<i>I</i>)
7. end function <i>action</i>

Etmen amaçlarını bir küme olarak göstermek uygulamada kolaylık sağlar. Her bir amacın ilgili önemliliğini belirten bir öncelikle birleştirmek basit bir alternatiftir. Diğer bir fikirde amaçları yığın şeklinde göstermektir. Bir amaç bırakıldığında amaç yığının üzerine itilir ve gerçekleştirildiğinde veya gerçekleştirilmediğinde amaç çıkarılır. En soyut amaçlar yığının en altındayken, daha somutlar tepeye doğru yerleşir.

BDI yapıdaki en önemli sorun amaçlarını yapma ve yapamama arasındaki denge problemidir; etmenin düşünme süreci etmenin çevresine uygun bir şekilde ayarlanmak zorundadır. Çok dinamik ve iyi öngörü yapılamayan çevreler de göz önüne alınarak düşünme mekanizması tasarımı yapılmalıdır. Çünkü bu çevrelerde etmen, diğerlerine göre amaçlarını daha sık tekrarlayarak düşünülmelidir. Çok statik çevrelerde ise, etmenin tekrar düşünmesinin az olması sorun oluşturmaz.

BDI modeli birçok farklı sebeplerden dolayı caziptir. Öncelikle sezgiseldir – etmenin ne yapacağına ve nasıl yapacağına karar alan süreçlerini tamamıyla fark ederiz. İkincisi, bir etmeni oluşturmak için ihtiyaç duyulacak alt sistemlerin ne türde olduğunu belirten açık fonksiyonel analiz sağlar. Fakat ana zorluğu, bu fonksiyonları nasıl etkin uygulayabileceğimizi bilmemektir.

Uzay gemisinin tepkili kontrol sistemi için hata düzeltme, Avustralya Sydney Hava Alanında hava trafiğini yöneten sistemlerdeki gibi bazı zor etmen uygulamaları bu mimarilerde oluşturulmuştur [2].

4.3.4. Katmanlı Mimariler

Tepkili ve önetkin davranış yeteneği olan etmende verilen koşul, farklı tür davranışlarla iş yapmak için ayrı alt sistemlerin oluşturulmasını kapsayan açık bir ayrışmadır. Bu fikir etkileşimli katmanların hiyerarşisinde sıralanmış, çeşitli alt sistemlerdeki yapıların bir sınıfına yol açar.

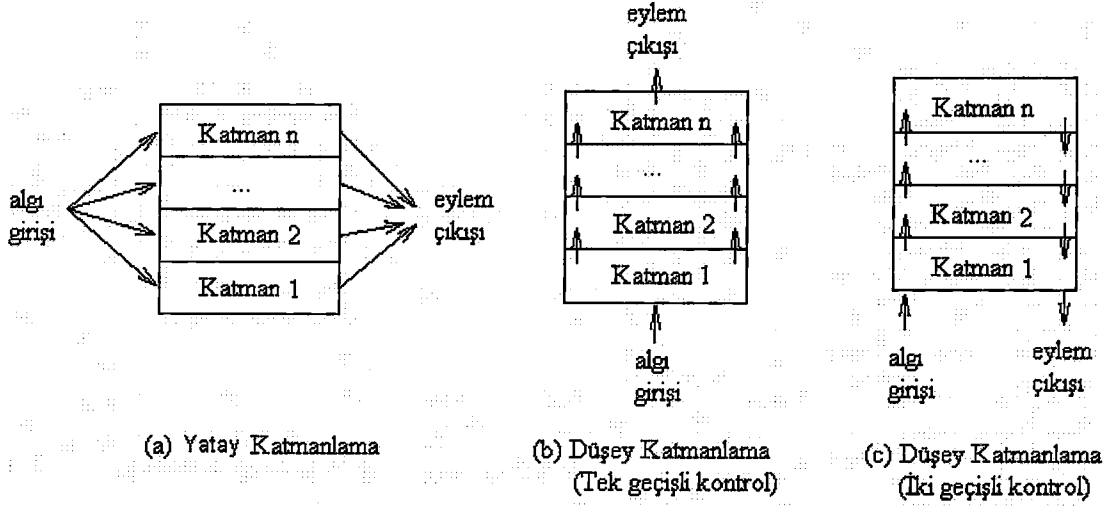
Doğal olarak hem tepkin hem de önetkin davranışlarla çalışmak için en az iki katman olmalıdır. Prensipde çok fazla katman olmasında sorun yoktur. Fakat çok fazla katmanın bulunması, bu tür mimariler için kullanışlı topoloji, katmanların sınırları içindeki bilgi ve kontrol akışıyla olur. Genel konuşursak, katmanlı mimarilerde iki tür kontrol akışı tanımlayabiliriz (Şekil 4.4 'e bakın):

▪ Yatay katmanlama

Yatay olarak katmanlanmış yapıda (Şekil 4.4(a)), yazılım katmanlarının her biri doğrudan algı girişine ve eylem çıkışına bağlanır. Sonuçta, her katman hangi eylemi yapacağına dair öneriler üreten etmene benzer davranır.

- Düşey katmanlama

Düşey olarak katmanlanmış yapıda (Şekil 4.4(b) ve 4.4(c)) algı girişi ve eylem çıkışının her biri en fazla bir katmanla çalışır.



Şekil 4. 4 Katmanlı Etmen Mimarisinin Üç Türündeki Bilgi ve Kontrol Akışları

Yatay olarak katmanlanmış yapıların en büyük avantajı kavramsal kolaylıklarıdır; bir etmenin n tane farklı türde davranış göstermesine ihtiyaç duyarsak, n tane farklı katman oluştururuz. Fakat etmenin tüm davranışıyla uyumlu olmama tehlikesi vardır. Çünkü katmanların her birinde eylem önerisi üretirken bir başkasıyla çekişme sorunu bulunur. Bu yapıların tutarlılığından emin olmak için, aracı bir fonksiyon içine konulur, bu fonksiyon her hangi bir zamanda etmenin kontrolüne sahip olacak katmanla ilgili kararlar alır. Bu tür merkezi kontrole ihtiyaç duymak sorun yaratır; yani tasarımcı katmanlar arasındaki tüm olabilecek etkileşimleri potansiyel olarak düşünmek zorundadır. Yapıda n tane katman varsa ve her katmanın önerilen m tane olabilecek eylem yeteneği varsa, bu demektir ki, düşünülecek m^n tane türde etkileşim vardır.

Bu problemler kısmen düşey olarak katmanlanmış yapıda hafifletilir. Bu yapıları tek geçişli (Şekil 4.4(b)) ve çift geçişli (Şekil 4.4(c)) yapılar içinde alt bölümlere ayırabiliriz. Tek geçişlide, son katman eylem çıktısı üretene kadar, sırayla her bir katmana doğru kontrol akışı vardır. Çift geçişli yapılarda, bilgi yapının yukarısına doğru akar (ilk geçiş) ve sonra kontrol geri olarak aşağıya doğru akar. Çift geçişli yapı düşüncesinde, organizasyonun en yüksek düzeyinden, yukarıya doğru akan bilgi ve sonra komutların aşağıya doğru akmasıyla çalışan organizasyonların yollarıyla arasında bazı ilginç benzerlikler bulunur. Hem tek hem de çift geçişli yapılarda katmanlar arasındaki etkileşimin karmaşıklığı azalır: n tane katman

arasında $n-1$ tane arayüzey bulunur, eğer her bir katman için önerilen m tane eylem yeteneği varsa, katmanlar arasında düşünülmesi gereken en fazla $m^2(n-1)$ tane etkileşim vardır. Bu yatay katmanlama olayından belirgin olarak daha basittir. Fakat bu kolaylık bazı esneklik problemi getirir.

Düşey katmanlanmış yapıda bir karar vermek için, kontrol her farklı katman arasından geçmek zorundadır. Bu hata toleransı değildir; her hangi bir katmandaki hata etmenin performansını etkileyecek olası ciddi sorunlara sebep olur [2].

4.4. Zeki Etmenlerde Gerçek Zamanlılık

Karmaşık bir çevrede bir etmenin algılama, düşünme ve eylem faaliyetleri çoğu zaman onun bilgi işlem kapasitesini zorlar [12]. Bu etmen hemen her zaman algılama, düşünme ve eylem faaliyetlerinin yükü altında bulunduğu için, yapabileceği bütün işlemleri zamanında ifa edemez. Bu yüzden zeki etmenlerin, gerekli bilgi ve işlerlik isteyen görevleri yapmada yeterli bir davranış içinde olması esas kabul edilir.

Diğer özellikleri eşit olmak şartıyla, bir etmenin başarabileceği görevler ve uyum sağlayabileceği şartlar ne kadar fazlaysa, etmen o kadar zeki kabul edilir. Bir etmen, her fırsat için bütün mümkün işlerden hangisini yapacağına karar vermek için, amaçlarıyla alakalı bilgileri, kısıtları, kaynakları ve çevresini kullanabilmelidir.

Zeki etmen dinamik bir çevre içinde olduğundan ve sürekli akış içinde değişen olaylarla karşı karşıya bulunduğu anda, yaptığı eylemlerin gereklerini karşılayacak şekilde sürekli kontrol kararları alabilmelidir. Genel olarak bir etmen, gerçek zaman ve öteki kaynak kısıtları altında alabileceği en iyi sonuçları almak için zeki kontrol uygulamak zorundadır.

Öteki bilgisayar sistemlerinden farklı olarak, zeki etmenler için gerçek zamanlı performans, etmenin birkaç amacından biri olarak görülmelidir. Karşılaştığı çevreyle etkileşimlerinde, elindeki kaynaklar ve verdiği kararlar çevresinde gerçek zamanlılığı az veya çok sağlayabilir. Birçok durumda etmen bir görev için zamanlı bir sonucu ancak öteki görevlerindeki sonuçların kalitesinden veya zamanlılığından fedakarlık yaparak alabilir. Zeki etmenin bir konudaki ihtisası arttıkça, bu tür fedakarlıklara olan ihtiyacı da artar [12].

4.5. Zeki Etmenlerde Gerçek Zamanlı Kontrol İçin Gerekli Şartlar

Zeki etmeni dinamik bir sistem olarak düşünebiliriz. Bütün sistem, zaman içinde, zaman birimlerinin ilgili değişkenlerin değerlerinin temsil edildiği durum uzayına tekabül ettirildiği durum serileri olarak modellendirilir. Bir durum değişkeni değerindeki bir değişim bir e olayıdır. Sistemin davranışı, durum değerlerinin fonksiyonları olarak tarif edilen ölçümlerle tasvir edilir [12].

4.5.1. Genel Çerçeve

Bütün sistemi zeki etmen, Z ve çevresi, Ç olarak iki bileşene ayırıyoruz. Buradaki iki bileşen de dinamik bir durum içindedir, ki bu durum bileşenlerin kendi içindeki ve bileşenler arasındaki bilgi akışına bağlı olarak değişir. Zeki etmen Z'yi de ayrıca üç ana bileşene ayırıyoruz: Algılama A, düşünme D ve eylem E bileşenleri. Bunlarında kendi içlerinde ve birbirleri arasında olaylar meydana gelir. Bileşenler arasındaki etkileşimler tetikleme (trigger) ve cevap (response) olaylarıdır.

4.5.2. Zeki Etmenin Çevresi

Zeki bir etmen belli bir çevreyle etkileşim içinde davranışlarını sürdürür. Zeki etmenin çevresiyle ilgili özelliklerini şöyle sıralayabiliriz:

Veri Yoğunluğu: Bir etmenin çevresinde kendisini ilgilendiren bütün olayları değerlendirmesi uygun olmaz, çünkü çevredeki olayların sayısı veya hızı zeki etmen içinde çevre etkisiyle meydana gelen olaylardan çok daha fazladır.

Veri Dağılımı: Önemli çevre şartları, etmen içindeki olaylar dizisine farklı durum değişkenleri ve zaman aralıkları içinde tekabül eder. (Yani çevrede meydana gelen olaylarla, bunların etmen içinde meydana getirdikleri etkiler farklı zaman aralıklarında yer alır.)

Olayların Çeşitliliği: Çevre şartları önemlilik açısından değişim gösterir; bu da etmen tarafından çevredeki olaylara farklı "önemlilik değeri" yüklenerek ifade edilir.

Gerçek Zaman Sınırlaması: Olayların değeri, kısmen vuku buldukları zaman değişir.

Şartların Çokluğu: Etmenin, karşılaşacağı ve onu ilgilendirecek bütün şartları sıralaması ve değerlendirmeye çalışması uygun değildir.

Öngörülebilirlik: Çevrenin, gelecek olaylar hakkında ihtimali öngörüler yapmaya imkan verecek kadar düzenli olması gerekir.

Mümkün Etkileşimler: Bütüncül (global) olarak planlanan eylemler bazen yerel olarak tespit edilen eylem dizilerinden daha üstündür.

Çevre Modeli: Çevre hakkında bilginin olması gerekir.

Değişik İstekler: Çevreyle etkileşim için çoklu etkileşim istekleri şunları gerektirir: Yorumlama, teşhis (tanıma), öngörü, tepki, planlama ve açıklama.

Değişken Gerilim: Çevrenin etmen üzerindeki gerilimi zaman içinde değişir.

4.5.3. Etmen Özellikleri

Çok genel manada bir etmenin ana gayesini şöyle tarif edebiliriz:

Kabul edilebilir bir zaman süresi içinde kendi davranışının değerini (ölçüsünü) korumak.

Belli bir çevredeki bir etmen için bu özelliği bir formülle ifade edebiliriz. Bunu, Ç çevresi içinde Z'ye bağlı olayların ve gerekiyorsa Z içindeki olayların faydalılığı cinsinden ifade edebiliriz.

Bu faydalılık ölçüsü etmenin bir çevre içindeki davranışını değerlendirmek için kullanılsa da, bunun bize etkin etmenler tasarımı için pek az faydası olur.

Etmen tasarımında yazılım mimarileri için daha temel bir dizi özelliğe ihtiyaç vardır. Aşağıda daha önce özelliklerini verdiğimiz türden çevrelerde esas gayesine ulaşması için gerekli özelliklerden bazılarını veriyoruz. Fakat önce zeki etmenin üç ana bileşenini hatırlatmamız gerekiyor: Algılama, Düşünme, Eylem. Şimdi zeki etmen özelliklerini sıralamaya geçebiliriz.

İletişim: Etmenin çevresiyle etkileşimi için etmen bileşenlerinin katıldığı uygun iletişimlere ihtiyaç vardır. Burada enformasyon en azından çevreden Algılamaya, Algılamadan Düşünmeye ve Düşünmeden Eylem bileşenine, oradan da çevreye geçme bilmelidir.

Serbest Zamanlılık: Veri yoğunluğu ve gerçek zaman kısıtları altında etmen çevreye göre serbest zamanlı olarak bağımsız olmalıdır. Ayrıca A, D ve E bileşenlerindeki olayların hızı da birbirinden bağımsız olmalıdır.

Seçicilik: Veri yoğunluğu ve çevredeki olayların çeşitliliği altında etmen çevredeki olayları nasıl algılayacağına, onlar üzerinde nasıl düşüneceğine ve nasıl tepki göstereceğine kendisi karar vermelidir.

Yakınlık: Bir etmenin algılama verileri kaybolucudur, bir olay üzerinde düşünmesi zamanla etkinliğini kaybeder. Ayrıca eylemlerin etkinliği de geçici dış olaylarla uyumluluğuna bağlıdır. Bu yüzden yakınlık önemli bir seçicilik kıstasıdır (ölçütüdür).

Bütünlük: Etmen gerektiğinde yerel olarak karar verilen eylemler dizisi yerine bütüncül olarak planlanan bir eylem ortaya koyabilmelidir.

Esneklik: Aynı şekilde, etmen hareketli bir çevre içinde beklenmedik önemli olaylara da tepki verebilmelidir.

Cevap Verme: Diğer hususlar aynı kalmak üzere, bir durum ne kadar acilse, etmen bununla ilgili verileri o kadar çabuk algılayabilmeli, gerekli sonuçları çıkarabilmeli ve uygun eylemleri yapabilmelidir.

Zamanlılık: Hareketli bir çevrede kendi davranışının faydalı olabilmesi için etmen, gerçek zaman kısıtlarını karşılayabilmelidir.

Gürbüzlük: Etmen, kaynakları zorlayan durumlara, davranışının faydalılığını kaybetmeden, azaltarak uyum sağlayabilmelidir.

Ölçü Artırma: Etmenin yukarıda sıraladığımız özellikleri problemin büyüklüğüne bağlı olmadan kalabilmelidir.

Gelişme: Etmen, kendi davranışının faydalılığını artırmak için yeni bilgileri alabilmeli ve kullanabilmelidir. Etmendeki bilgi miktarı arttıkça onun yukarıda saydığımız özelliklerinden bazılarını geliştirmesi beklenir. Böylece bütüncül davranışının faydalılığı artar.

4.6. Etmen Mimarisi

Bu kısımda yukarıda belirtilen özelliklere sahip bir etmen mimarisi anlatılmaktadır [12].

4.6.1. Üst Düzey Yapı

Daha önce belirttiğimiz gibi bir Z etmeni Algılama, Düşünme ve Eylem (A, D, E) bileşenlerinden meydana gelir. Şekil 4.5 'te verilen yapı her üç bileşenin daha alt bileşenlere ayrıldığını göstermektedir. Bu şekildeki kıvrık köşeli kutular veri yapılarını, dikdörtgen kutular da paralel işlem operatörlerini göstermektedir. Oklar ise veri yapıları ve operatörler arasındaki veri akışını göstermektedir. Buradaki yapı hiyerarşiktir.

Üç ana bileşen A, D ve E arasındaki İletişim Arabirimi bunlar arasında veri akışını yönlendirir. Çevreden gelen sinyaller Algılama bileşenine gelir. Bu, seçici olarak Düşünme bileşeni tarafından tayin edilen dikkat parametreleri altında, gelen

sinyalleri süzer ve yorumlar. Algı sonuçlarını da çıkış kapısına gönderir. İletişim arabirimi bu algıları ya Eylem bileşeninin giriş kapısına iletir, ki bunlar burada doğrudan eylem oluşturur, veya bunları Düşünme bileşeninin giriş kapısına gönderir.

Düşünme bileşeni, giriş kapısındaki algıları alır, bilgi tabanına ekler, gerekli bilgi işlemlerini yapar. Aldığı eylem kararlarını ve diğer parametreleri kendi çıkış kapısına gönderir. İletişim Arabirimi bu kararları uygun Algılama/Eylem bileşenlerine gönderir. Eylem bileşeni, eylem tasvirlerini giriş kapısından alır ve Düşünme bileşeninden alınan parametrelere (veya ölçülere) uygun olarak etkileyicilerini çalıştırır. Böylece uygulanan eylemler, çevredeki varlıkları etkiler.

Bu yapıdaki birimler paralel çalışırlar. Birbirleriyle doğrudan iletişim halinde değildirler. Birbirlerini ancak dolaylı olarak - kendi çıkış kapılarına mesaj bırakarak - etkilerler ki bu mesajlarda İletişim Arabirimi tarafından öteki birimlerin giriş kapılarına iletilir.

Şekilde gösterildiği gibi, bu yapı Algılama ve Eylem Üniteleri arasında doğrudan mesaj akışı ile hızlı refleks davranışları da destekler. Daha yavaş ve düşünme gerektiren davranışlarda ise her üç bileşende rol alır [12].

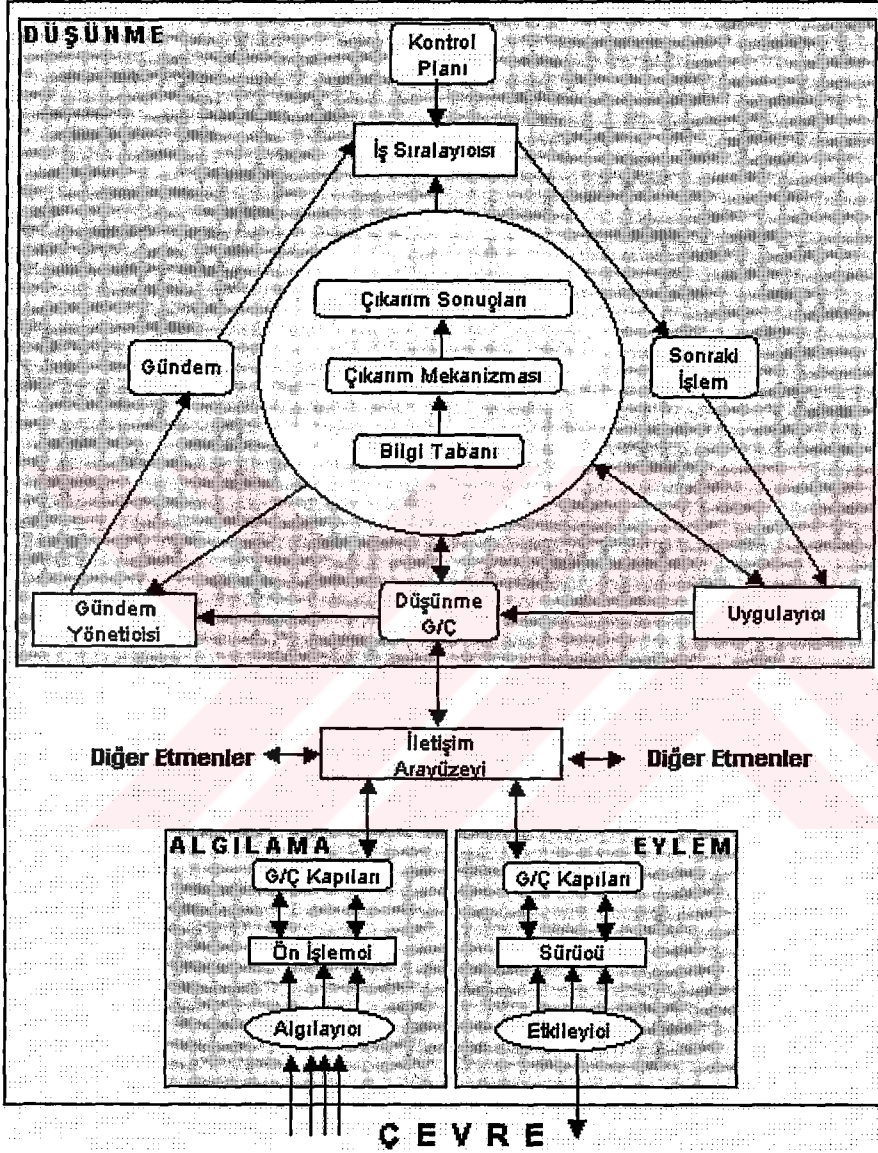
4.6.2. Algılama Bileşeni

Algılama bileşeni ve altsistemleri algılayıcılar ve işlemcilerden oluşur ki bunlar, hareketli çevre hakkında düşünme ve eylem için gerekli bilgileri alır. Her algılayıcı özel bir tür sinyali alır ve sonucu, ilgili ön işlemci tarafından alınmak üzere sınırlı kapasiteli bir bölmede tutar. Her ön işlemci algılanan bilgiyi soyutlar, işaretler ve süzer ve sonucu kendi çıkış kapısına İletişim Arabirimi tarafından alınıp Düşünme veya Eylem bileşenine taşınmak üzere bırakır. Soyutlama, algılanan verilerin soyutlama biçimlerine bağlı olarak yorumlanması ve çoğu zaman da sıkıştırılması şeklinde olur. Süzme ise, süzme kıstaslarına göre soyutlama sonuçlarının iletilmesini kısıtlamaktır.

4.6.3. Eylem Bileşeni

Eylem bileşeni ve alt sistemleri sürücüler ve etkileyicilerden oluşur ve bunlar eylem tasvirlerini giriş kapılarından alırlar. Sonra etkileyiciler üzerinde eylemi kontrol ederler ve geri beslemeyi Düşünme bileşenine gönderirler. Her sürücü, giriş kapısını gözler, istenen eylem şemalarını alır ve onları etkileyici komutları şeklinde uygulanabilir programlara çevirir. Sonra bu programların uygulanmasını uygun zamanda uygun etkileyiciye birbirini izleyen komutlar göndererek denetler. Her sürücü, aynı zamanda davranış üzerindeki önemlilik, öncelik ve diğer kriterleri

hesaba katar. Sürücü, aynı zamanda eylem uygulamasının başarılı olup olmadığını da Düşünme bileşenine geri besleme ile bildirmelidir. Her etkileyici giriş kapısına gelen komutları hemen uygular.



Şekil 4. 5 Amaçlanan Etmen Mimarisi

4.6.4. Düşünme Bileşeni

Düşünme bileşeni, etmenin bütün bilgilerini toplar ve bunlar üzerinde çıkarım işlemlerini yapar. Giriş kapısından gelen algılama bilgilerini, kendi zamanında bilgi tabanına ekler. Farklı çevrelere göre değişen birçok bilgi tabanlı düşünme işini yürütür. Bunlardan bazıları: Algılanan bilgileri yorumlamak; istisnai olayları tespit ve tanıma; gelecek olayları öngörmek; değişken dış sistemleri modellemek; daha uzun vadeli eylemleri planlamak; gözlemleri açıklamak; çıkarımlar ve planlar yapmak;

kendi çıkarımlarını açıklamak; tecrübelerden davranışını geliştirmeyi öğrenmek ve davranışlarını değişen çevre şartlarına uydurmak. Bunlardan başka, çoklu görevlerin global komutları hakkında düşünmek, bunların etkileşimlerini koordine etmek ve eldeki kaynaklara göre en önemli amaçlara zamanında ulaşmak gibi işleri de yapabilmektir. Düşünme bileşeni, eylem tasvirlerini kendi çıkış kapısına göndererek eylemleri başarır.

Bütün düşünme işlemleri, bütün bilgileri ve düşünme işlemlerinin sonuçlarını içine alan global bir hafıza içinde olur. Önemli bir bilgi türü, akıl yürütme işlemleri ve bunlara bağlı stratejilerdir. Bunlar, teşhis, öngörü, açıklama ve planlama gibi özel görevleri özel metotlarla yapmak için etkin hale getirilir.

Yukarıda anlatıldığı gibi, global hafıza Algılama bileşeninden gelen algıların tutulduğu giriş kapılarına sahiptir. Ayrıca, eylem sürücülerine gönderilen eylem şemalarının ve algılama ön işlemcilerine ve eylem sürücülerine gönderilen kontrol parametrelerinin tutulduğu çıkış kapılarına sahiptir. Giriş/çıkış kapıları sınırlı kapasitededir, bu yüzden en iyiye öncelik verilir, en kötü dışarı atılır.

Global hafıza aynı zamanda etmenin düşünme davranışı ile ilgili bilgileri de ihtiva eder. Bu düşünme kapısı düşünme işlemlerinin meydana getirdiği düşünme olaylarını tutar. Bu gündem (ajanda) kutusu, algılama veya düşünme olayları tarafından önerilen uygulanabilir düşünme işlemlerini tutar. Bir kontrol planı kutusu da etmenin, düşünme işlemlerince kararlaştırılan düşünme işlemlerini gösterir. Son olarak, global hafıza, gözlemler, çıkarımlar, öngörüler, ve planlar gibi düşünme işlemlerinin sonuçlarını tutar.

Düşünme bileşeni global hafızadaki bilgilere göre düşünme işlemlerini yapar ve sonuçta buradaki bilgileri değiştirir. Bunun çalışma çevrimi üç süreçten ibarettir:

1. Gündem yöneticisi, gene gündeme kaydettiği uygulanabilir düşünme işlemlerini tayin etmek ve sıralamak için en son algılama ve düşünme verilerini kullanır.
2. İş sıralayıcı tayin edilen uygulanabilir işlemlerden hangisinin ve ne zaman uygulanacağına karar verir. Bunu yaparken işlemlerin değerlerine bakar ve her birini müteakip (izleyen) işlem olarak kaydeder.
3. Uygulayıcı, kayıt edildiği şekliyle bir sonraki işlemi uygular.

5. ZEKİ ETMEN UYGULAMA ALANLARI

Etmen uygulamalarını sınıflamak için birçok yol vardır. Uygulamalar etmen türlerine göre, etmen oluşturmak için kullanılan teknolojilere göre veya etmenlerin kendi uygulama alanlarına göre sınıflandırılabilir. Bu tez çalışmasında alan türü kullanımı seçilmiştir. Çünkü tezin amacına ve yapısına en iyi bu bakışın uyduğu düşünülmüştür.

5.1. Endüstriyel Uygulamalar

Etmen teknolojisindeki endüstriyel uygulamalar diğer alanlar arasında ilk geliştirilenlerdir: ilk olarak Parunak 1987'de, üretim ortamında çalışan anlaşma ağının (contract net) görev bölüştürme protokol uygulamasıyla ilgili bir sistem geliştirdi. Bugün, etmenler endüstriyel uygulamalarda çok yaygın kullanılmaktadır [13].

5.1.1. Süreç Kontrolü

Süreç kontrolü zeki etmenler ve çoklu-etmen sistemleri için doğal bir uygulamadır, çünkü süreç kontrolcülerin kendileri bir ortama yerleştirilmiş tepkili sistemlerdir [13]. Bu nedenle, birçok sayıda etmen tabanlı süreç kontrol uygulamaları geliştirilmiştir. Bunlar içinde en iyi bilineni ARCHON, çoklu-etmen sistemlerini oluşturmak için bir yazılım platformu ve bu platformla uygulamaları oluşturmak için birleştirilmiş bir yöntemdir (Jennings). ARCHON, elektrik taşıma yöntemi (kuzey İspanya'da kullanılan uygulama) ve parçacık ivme kontrolü içeren, birçok süreç kontrol uygulamalarında kullanılır. Üstelik ARHCON dünyadaki en eski alan testli çoklu-etmen sistemlerinden biri olduğu için farklı bir yere sahiptir. ARCHON'daki etmenler oldukça iyi hesaplama sistemleridir: yüksek seviye haberleşme modülü, etmen içi iletişimi yönetir; planlama ve koordinasyon modülü, etmenin ne yapacağına karar vermesinden sorumludur; etmen bilgi yönetimi modülü, etmen dünya modelinin devam ettirilmesinden sorumludur. Son olarak, oluşturulan zeki sistem etmenin yaptığı işle ilgili alan uzmanlığını gösterir.

5.1.2. Üretim

Parunak üretim kontrolü için iyi tanınan Anlaşma Ağı (Contract Net) protokolünü kullanan YAMS'ı (Yet Another Manufacturing System) geliştirmişti [13]. Temel problem şu şekilde tanımlanabilir. Üretim yapan bir kuruluş, çalışma birimleri hiyerarşisi olarak modellenir. Burada örneğin, dişli yapımı, tornalama, bileme, boyama ve diğerleri çalışma birimleri olacaktır. Ayrıca bu çalışma birimleri esnek üretim sistemleri içinde gruplanacaktır, her biri boyayı spreyleme, ürünleri dizme gibi, bir işlevsellik sağlayacaktır. Bu tür esnek üretim sistemleri topluluğu fabrika içinde gruplanacaktır. Tek bir şirket veya organizasyon farklı birçok fabrikaya sahip olabilir, bununla beraber bu fabrikalar işlev ve yeteneklerini tekrarlayabilirler. YAMS'ın amacı bu fabrikalarda üretim işlemini verimli olarak yönetmektir. Bu işlem imal edilen ürünler, elde edilebilen kaynaklar, zaman sınırları ve diğerleri gibi, sürekli değişen bazı parametreler yoluyla belirlenir. Gerçekleştirilmesi çok karmaşık görev için YAMS çoklu-etmen sistemini kullanır, burada her fabrika ve fabrika parçası bir etmen olarak belirtilir. Her etmen, yeteneklerini belirten amaçların bir toplamına sahiptir. Anlaşma Ağı protokolü birbirinden ayrı fabrikaları görevlendirmeyi ve bu bireysel fabrikalardan aşağıya inerek esnek üretim sistemlerini ve sonrada birbirinden ayrı çalışma birimlerinin görevlendirilmesini sağlar. Anlaşma ağının temeli, görüşme (negotiation) fikri üzerine dayandırılır. Bu yüzden YAMS, bir şirketin ürün imalat ihtiyaçlarını en iyi nasıl yönetileceğini kararlaştırma problemine bakar. Wooldridge ve arkadaşları, bir fabrika için en iyi ürün serisini kararlaştırma problemini, oyun araçları (tools of games) ve görüşme teorisi kullanarak analiz eden benzer birkaç yaklaşım geliştirmişlerdir.

5.1.3. Hava Trafiği Kontrolü

Kinny ve arkadaşları, OASIS olarak bilinen çok yönlü etmenle geliştirilmiş bir hava trafiği kontrol sistemini geliştirdi [13]. Bu sistemde, Avustralya Sydney hava alanındaki alan testlerinden geçen, operasyonda hem uçak hem de çeşitli hava trafiği kontrol sistemlerini temsil eden etmenler kullanılır. Böylece modellendirilmiş gerçek dünyadaki otonom parçaların kullanışlı ve doğal yolunu etmen sağlar. Sydney hava alanına bir uçak girerken, uçak için bir etmen ayarlanır, etmen gerçek uçağa benzer bilgi ve amaçlarla donatılır. Örneğin, bir uçağın belirli bir zamanda belirli bir piste inmek için amacı olabilir. Hava trafiği kontrol etmenleri sistemin idare edilmesinden sorumludur. OASIS inanç-istek-amaç yapısı kullanılarak oluşturulmuş bir etmen sistemiyle yapılmıştır - Teorik çoklu-etmen sistemlerindeki etmenlerle ilgili neden sonuçlu en ünlü yaklaşımlardan birisidir.

5.2. Ticari Uygulamalar

5.2.1. Bilgi Yönetimi

Her gün elde ettiğimiz bilgi zenginliği ve çeşitliliği artarken, bu bilgileri idare etme ihtiyacı o kadar büyümektedir [13]. Etkili bilgi yönetim araçlarının bulunmaması fazla bilgi yüklenme sorununu artırmaktadır. İnternet ve World Wide Web (WWW) yoluyla elde ettiğimiz bilginin hepsi çok ciddi bir problemdir. WWW yüzeysel kullanan kişiden daha çok, kaynak elde etmek için kullanan herhangi birinin bu sorun hemen gözüne çarpar. Bunun bir çok sebebi vardır. Hem insan etkeni (kullanıcılar sıkılır ve dikkatleri dağılmaya başlar) hem de organizasyon etmeni (kötü bir şekilde düzenlenmiş sayfalar) sistemli bir yolla kullanıcıların kaynakları kullanmaya kalkışmalarına karşı zorluklar çıkarır. Fazla bilgi yükleme problemi iki yolla nitelendirilebilir:

- Bilgi filtreleme. Her gün, sadece çok az ilgilenilen konuyla alakalı ve önemli olan muazzam miktarda bilgiyle (posta ve haber gruplarındakiler gibi) karşı karşıya kalırız. Samandan buğdayı ayırabilmeye ve istediğimiz bilgi üzerine odaklanmaya ihtiyaç duyarız.
- Bilgi toplama. Aslında mevcut bilgi çokluğu belirli soruların cevabı olan bilgileri bulmamızı önler. İstenilen bilgi birçok siteden yalnızca toplanabilecek olsa bile, elde ettiğimiz bilginin bizim ihtiyaçlarımızı karşılamasını isteriz.

Fazla bilgi yüklenmesine katkıda bulunan en önemli neden elbette, yönetim işleminde sürekli olarak son kullanıcının idare etmesine duyulan ihtiyaçtır. Fakat prensipte, bu tür web aramalarının bazı kullanıcıların adına otonom davranan etmenler vasıtasıyla yerine getirilmemesi için bir neden yoktur. Tam olarak bu işe yönlendirilmiş birçok proje bulunur. Aşağıdaki üç proje bu türe örnektir:

- Maxims: (Maes) Maxims olarak isimlendirilen elektronik mail filtreleme etmenidir. Program kullanıcı adına mail mesajlarına öncelik vermeyi, onları silmeyi, iletmeyi, sıraya dizmeyi ve arşivlemeyi öğrenir. Kullanıcı email okuma programıyla çalışırken, kullanıcının tercihlerine bakarak çalışır ve yerine getirilen her bir kullanıcı davranışını ders olarak kullanır. Maxims devamlı kullanıcının mesajla ne yapacağını tahmin etmeye çalışır. Eğer bu tahminler doğru çıkmazsa, kendisi için bunları saklar. Fakat tahminlerini başarılı bir derecede buluyorsa, kullanıcıya ne yapması gerektiğini önermeye başlar.
- Newt:(Maes) Newt olarak isimlendirilen İnternet haber gruplarını filtreleyen bir programın örneğidir. Bu program UNIX platformunda C++ ile oluşturulmuştur,

Usenet haber grupları yazılarının bir akımını giriş olarak alır ve okuması için kullanıcıya tavsiyede bulunacağı yazıların alttakımını çıkış olarak verir. Newt etmeni eğitim örnekleri araçlarıyla programlanır; kullanıcı okuyup okumayacağı yazıların Newt örneklerini verir ve bu geri besleme kullanıcı meraklarını dahili model düzenleyerek kullanılır. Eğer kullanıcı isterse, Newt'e kesin kurallar vererek (örneğin, “‘etmen’ kelimesini içeren tüm yazıları ver” gibi.) Newt'i programlayabilir. Newt belgelerin vektör alan modelini kullanan tam-metin analizini çalıştırarak bir yazıdan merak edilen kelimeyi bulup çıkarır.

- Zuno Digital Library: Dijital kütüphane, kullanıcıya veriyi kullanmada yardımcı olan servislerle beraber, düzenlenen, yönetilen bir veri koleksiyonudur. Zuno Digital Library (ZDL) sistemi kullanıcının, web sayfaları, kullanıcının kendi verileri, yayınlanan ana sitelerdeki yazıların tümü ve bunun gibi, tutarsız, düzensiz veri kaynaklarının tek, kolay anlaşılabilir görünümünün elde edilebilmesini sağlayan bir çoklu-etmen sistemidir. ZDL'deki etmenler şu üç rolden birini oynar:
 - Alıcı - bilgiyi alabileceği düşünülen, sistemin son kullanıcılarını temsil eder.
 - Üretici - alıcıların kullandığı bilgi kendisinin olan 'içerik sağlayanları' temsil eder.
 - Kolaylaştırıcı - alıcılar ve üreticiler arasında bağlıdır.

Sistemdeki alıcı etmenleri kullanıcının meraklarını bildirmekten sorumludur. Kullanıcı modellerini korurlar ve bu modelleri kullanıcıya yardım etmek için kullanırlar, kullanıcıların istediği bilgiyi önetkin olarak temin ederler ve ilgilenmedikleri bilgilerden de kullanıcıyı korurlar. Böylece ZDL hem bilgi filtreleme hem de bilgi toplama vazifesini yapar [13].

5.2.2. Elektronik Ticaret

Günümüzde, ticaret hemen hemen tamamıyla insan etkileşimleriyle yapılır; insanlar ne zaman malları alacaklarına, ne kadar ödeme yapacaklarına karar verirler. Fakat prensipte, ticaretin otomatik olmaması için bir neden yoktur. Bunun yanında, bazı ticari kararların etmenlerin kontrolüne verilmesini amaçlarız. Mantıklı ticari bir organizasyonun bir bilgisayar programının sorumluluğunda karar almasının gerçekçi olmayacağı sanılabilir. Stokların ve hisselerin elektronik ticaretinde bugün ne olduğunu tam olarak bu durum hatırlatabilir. Son yıllarda birçok araştırmacı elektronik ticaret için pazar yapıları tasarımı ve bu tür işlerle meşgul olan kişisel çıkarları olan etmenler üzerine odaklanmıştır. Etmenlerin birbirleriyle olan

etkileşimini yöneten çeşitli protokoller (görüşme protokolleri) geliştirilmiştir. Etmenlere “Müşteri” ve “Mal Sağlayıcı” gibi roller verilmiştir. Bir örnek olarak, Chavez ve Maes, tarafından Kasbah isminde basit bir “elektronik pazaryeri” geliştirilmiştir. Bu sistem, satın alınacak veya satılacak her bir mal için sırasıyla ‘alma’ ve ‘satma’ etmenleri oluşturarak pazar-yeri gerçekleştirir. Ticari muamele bu etmenlerin etkileşimleriyle meydana gelir [13].

5.2.3. İş Süreci Yönetimi

Şirket yöneticileri fikirlerinin ve birçok departmandan alınan bilgilerin birleştirilmesi üzerine dayanarak kararları hakkında bilgi verirler [13]. İdeal olarak, konuyla ilgili tüm bilgileri fikirleriyle birleştirmeden önce kullanırlar. Fakat büyük bir şirket söz konusu olduğunda geçerli, tutarlı ve yenilenmiş bilgi elde edilmesi karmaşık ve zaman harcayan bir iştir. Bu nedenle, organizasyonlar iş süreçleri yönetimine çeşitli yönleriyle yardımcı olan bir çok bilgi sistemlerini geliştirmek için araştırma yapmak zorunda kalırlar. Jennings ve arkadaşlarının geliştirdiği ADEPT projesi, servis sağlayıcı görevi yapan etmenlerin görüşen topluluğu olarak iş sürecine bakarak bu problemi çözmeye çalışır. Her bir etmen ayrı bir rol veya girişimdeki departmanı temsil eder ve bir veya daha fazla servis sağlama yeteneği vardır. Örneğin, tasarım bölümü telekom ağını tasarlayan servisi hazırlayabilir, hukuki bir bölüm tasarımın yasal olup olmadığını kontrol eden servisi sağlayabilir ve pazarlama bölümü tasarım ücretlendirme servisini hazırlayabilir. Diğer etmenden servis isteyen bir etmen karşılıklı olarak makul fiyat, zaman ve kalite derecesini sağlayan servis için bir görüşmeye başlar. Etmenler arasındaki başarılı görüşmeler sözleşmenin bağlanmasıyla sonuçlanır. Bu etmen tabanlı yaklaşım, daha doğal işakışı çözümleri aracılığıyla probleme bakarak birçok avantaj sağlar. Etmenlerin önetkin doğası tam zamanlı biçimde (başlangıçtan önce belirlenmiş olmasından ziyade) zamanlanabilen servisleri amaçlar ve etmenlerin cevap vermeye istekli doğası, esnek usulde fark edilebilen ve yönetilebilen servis sıra dışı durumlarını amaçlar. Sistem 200 etkinlik ve dokuz bölüm içeren İngiliz Telekomu’nun iş sürecinde denenmiştir.

5.3. Tıbbi Uygulamalar

Tıbbi bilgi, bilgisayar biliminin başlıca gelişme alanıdır: sağlık endüstrisinde her gün bilgisayarlar için yeni uygulamalar bulunmaktadır. Bu yüzden, bu alanda etmenlerin uygulanması sürpriz değildir. Hasta takibi ve sağlık bakımı bu alandaki en eski uygulamalardan ikisidir [13].

5.3.1. Hasta Takibi

Hayes-Roth ve arkadaşları Ameliyat Yoğun Bakım Biriminde (SICU) hasta bakımı yönetimine yardım etmeyi amaçlayan Guardian sistemini geliştirmişlerdir [13]. Sistemin geliştirilmesine yol açan iki husus vardır: İlki, SICU'daki hasta bakım modeli aslında, hasta sağlık bakımını organize eden ayrı alanlardaki uzmanların birlikte çalışmasıyla oluşan uzmanlar topluluğunun bulunduğu bir takımdır; ikinci olarak iyi bir SICU hasta sağlık bakımdaki en önemli etmenlerden biri acil bakım grup üyeleri arasındaki yeterli bilgi paylaşımıdır. Özellikle, uzmanlar hastanın dakika dakika durumlarını izlemek için çok az imkana sahiptir; bu göreve bakmak, uygun uzmanla aynı yolla bilgiyi alan ama yorumlamak için genellikle yeterli bilgiye sahip olmayan hemşirelere düşer. Guardian sistemi, üç farklı türde olabilen birçok etmen arasında SICU hasta takibi görevini yayar:

- Algı / eylem etmenleri - Guardian ve dünya arasındaki etkileşimden sorumludur, kullanılabilir sembolik biçimdeki işlenmemiş algı girişini bağlar ve işlenmemiş etkileyici kontrol komutlarındaki Guardian'dan eylem isteklerini çevirir.
- Tümevarım etmenleri - sistemin karar alma sürecini düzenlemekten sorumludur
- Kontrol etmenleri - üst düzey kontrol sistemiyle tam olarak aynı fikirde olacaktır.

Bu etmenler hiyerarşik olarak düzenlenirler ve blackboard olarak bilinen ortak bilgi yapısındaki bilgi paylaşımıyla, farklı etmenlerin ('bilgi kaynakları') birlikte çalıştığı, temeli blackboard kontrol modeline yakın olan tam bir sistemdir.

5.3.2. Sağlık Bakımı

Huang ve arkadaşları dağıtık tıbbi bakım sistemi geliştirmiştir [13]. Bu sistem, birçok bireyler içeren, hasta yönetimi sürecini tamamlayacak şekilde etmen tabanlı olarak tasarlanmıştır. Örneğin, bir pratisyen hastada göğüs kanseri olmasından şüphelenebilir, fakat bu şüphe hastane uzmanının yardımı olmadan doğrulanamaz veya reddedilemez. Uzman hipotezi doğrularsa, diğer bireylerin olanaklarını gerektiren, hastanın tedavi edilmesi için bir bakım programı düzenlenmek zorundadır. Diğer bireyler hasta bakıcılar, hemşireler, başka uzmanlar olabilir. Bu

etmen sistemi, bireylere ve hasta bakım sürecindeki ilgili olası organizasyonlara bağlanmış etmenlerle, bu sürecin doğal gösterimine olanak sağlar. Sistemdeki etmenlerde, etmenin alan uzmanlığını tutan bilgi-tabanlı (zeki) bir altsistem bulunur. Etmenler, kullanıcı için sistem amaçlarını ekleme, kaldırma veya görüntüleme sağlayan insan-bilgisayar arayüzüne sahiptir. Sistemde etmenlerin mesaj-geçme fonksiyonelliğini gerçekleştiren bir haberleşme sistemi yöneticisi bulunur. Zeki sistem bileşeni KADS bilgi modeli üzerine dayanır ve tüm etmen mimarisi PROLOG da oluşturulur. Mesaj geçme standart elektronik postaya ek yapılarak gerçekleştirilir.

5.4. Eğlence

Bu boş zaman endüstrisi bilgisayar bilimi topluluğu tarafından ciddiye alınmaz. Buna rağmen boş zaman uygulamaları birçok durumda bilgisayarların 'ciddi' uygulamaları olarak görünürler. Aynı zamanda bilgisayar oyunları gibi boş zaman uygulamaları ticari bakımdan oldukça kazançlı olabilmektedir. Etmenler bilgisayar oyunlarında, etkileşimli tiyatrodaki ve ilgili görsel gerçek uygulamalarda belli bir role sahiptir; bu tür sistemler etmenler gibi kolay oluşturulabilen yarı otonom hareketli karakterlerle dolu olmaya meyillidir [13].

5.4.1. Oyunlar

Wavish ve arkadaşları bilgisayar oyunları için birçok etmen teknolojisi uygulamaları tanımlamıştır [13]. Örneğin, düzensiz şekilde düşen blokların dışında bir duvar yapmak için kullanıcının uğraşmak zorunda olduğu ünlü Tetris bilgisayar oyununun bir versiyonunu geliştirdiler. Oyundaki etmen, blokların düştüğü yeri kontrol etmek zorunda olan kullanıcının yerini alır. Burada geleneksel sembolik yapay zeka teknikleri kullanılarak etmen programı yapmaya çalışmak zordur. Çünkü kullanıcının rolünü oynamak için bilgiyi ortaya çıkarmak, oyunla ilgili bilgileri göstermek ve kurallar şeklinde bulunan sembolik veri yapılarıyla çalışmak gerekir. Bu yaklaşım tamamıyla, yoğun gerçek zaman sınırlı, Tetris benzeri bir oyun için gerçekçi değildir. Wavish ve arkadaşları RTA (Real Time Able - Gerçek Zaman Yetenekli) isimli alternatif tepkili bir etmen modeli kullandılar. Bu yaklaşımda, etmenler davranışlarına göre programlanır. Bu davranışlar kurallara az çok benzeyen fakat karmaşık sembolik çıkarım istemeyen basit yapılardır.

5.4.2. Etkileşimli Tiyatro ve Sinema

Etkileşimli tiyatro ve sinemayla, tiyatrolarda veya filmlerdeki gerçek, insan aktörlerin oynadığı rollere benzemeyen başka bir rolün kullanıcı tarafından gerçek insan davranışı özelliklerine sahip yapay, bilgisayar karakterleriyle etkileşimli olarak birlikte oynanmasını sağlamak amaçlanır. Tiyatrocu biçimindeki uygulamalarda insanların yerine oynayan etmenler, Joe Bates tarafından ortaya atılmış bir terim olan “inanılan etmenler” (believable agents) olarak bilinirler. Bu tür etmenlerin gelişmesini araştırmak için birçok proje oluşturulmuştur. Örneğin, Hayes-Roth ve arkadaşları doğrudan uydurma (directed improvization) sistemi geliştirmişlerdir [13]. Burada insan oyuncular, yeni, kullanıcı yönlendirmeli ‘eser’ oluşturulan, aktif bir anlatımdaki rollerin dışında oynarlar.



6. UYGULAMA

6.1. Giriş

Tez çalışmasındaki uygulama herhangi bir etmen geliştirme aracı kullanılarak geliştirilmemiştir. Bigus'ların kitabındaki uygulamalardan yararlanılmıştır (Referans [14]'e bakın). Bu kitaptaki örnek uygulamalara ek olarak zeki bir etmen uygulaması geliştirilmiştir. Örnek uygulamalar UserNotificationAgent ve SchedulerAgent'dır. Zeki etmenin görevi şunlardır: Birincisi, kullanıcı yerine kitap sitelerine girmektir. İkincisi, kullanıcının istediği kitap konusu ve yazar adına göre sitelerde sorgulama yapmaktır. Üçüncüsü, kitap indirimleriyle ilgili topladığı tüm bilgiyi, çıkarım mekanizmasını kullanarak, kullanıcının indirim kriterleriyle mukayese etmektir. Son olarak, kullanıcının istediği kitap fiyatı ve indirimlerini kullanıcıya sunmaktır. Bu görevleri yaparken zeki etmen, kullanıcı, diğer etmenler ve diğer uygulamalarla etkileşim içindedir.

Bu bölüm iki kısımdan oluşmaktadır. İlki, zeki etmen çalışma çerçevesidir. Burada uygulamanın oluşturulmasıyla ilgili ön hazırlıklar ve saptanması gereken hedefler açıklanmıştır. İkincisinde, uygulama içinde çalışan etmenlerin görevleri ayrıntılı olarak anlatılmıştır. Uygulamanın çalıştırılması şekillerle tanıtılmıştır. Karşılaşılan HTML sorununa değinilmiştir.

6.2. Zeki Etmen Çalışma Çerçevesi

Uygulamada nesneye dayalı tasarım teknikleri kullanılarak zeki etmen mimarisi geliştirilmiştir. Programlama dili olarak Java kullanılmıştır. Borland JBuilder aracı kullanılarak grafik kullanıcı arabirimi oluşturulmuştur.

6.2.1. Tasarım Amaçları

Uygulamanın çalışma biçimine göre iki tasarım amacı vardır. Bir tanesi tek başına bir uygulama diğeri ise, başka etmenler veya uygulamalarla etkileşen etmenler topluluğudur. İlki, etmenlere uygulama merkezli bakış açısidir. Burada etmenler uygulamaya ve uygulamanın kullanıcılarına yardımcı olur. Bu yaklaşım çok

karmaşık değildir. İkinci yaklaşım ise daha çok etmen merkezli bir yaklaşımdır. Burada etmenler uygulamaları kontrol eder ve yönetir. Etmen yöneticisinin kendisi başlı başına bir uygulamadır ve etmenler tarafından yönetilen diğer uygulamalarla etmenler arasında arayüzey olmak zorundadır. Bu yaklaşımın karmaşıklığı etmen yöneticisinin uygulama haberleşmeleri için genel bir mekanizmanın tanımlama zorunluluğunun olmasıdır. Üstelik tasarım, yeni uygulamaların gerçekleştirilebileceği düşünülerek, yapay zeka tekniklerini ve diğer özellikleri kolayca ekleyebilmeyi sağlayacak şekilde esnek olmak zorundadır [14].

6.2.2. Geliştirilen Zeki Etmen Mimarisi

Zeki etmen mimarisinin öngördüğü fonksiyonları sağlamak için şu tasarım kararlarını almamız gerekiyordu:

1. Java uygulamasına zeki etmenlerin eklenmesi kolay olmalıdır.
2. Başka dış Java bileşenleri ve etmenlerin birleştirilmesi için grafik oluşturma aracı bulunmalıdır.

Borland'ın JBuilder'ı, WEbGain'nin Visual Cafe'si ve IBM'in VisualAge for Java'sı Java uygulamalarının parçalardan oluşturulmasını sağlayan grafiksel geliştirme araçlarıdır. Java, java.bean paketleriyle temel bileşen modeli sağlar. JavaBean'ler bir uygulamayı oluşturmak için parçaların birlikte konulabilmesini sağlayan, yazılım fonksiyonlarına izin veren Java bileşen modelidir. Geliştirme araçları, her bir JavaBean nesnesinin idare edilebilmesi için görsel oluşturma aracına izin veren ara yüzeye sahiptir. Ayrıca araçlar, JavaBean'lerden oluşturulmuş uygulamaları çalıştırmayı sağlayan çalışma zamanlı ara yüzeye de sahiptir.

Bean'lerin iyi özelliği, etmenleri başka etmenlere bağlama yeteneğini uygulama için karşılayabilmesidir. Böylece, JavaBean'ler özel amaçlı etmenleri daha yüksek düzey etmenlerde tekrar kullanabilmemizi sağlar. Örneğin öğrenme için yapay sinir ağlarını kullanan düşük düzey etmenlerimiz olabilir, ayrıca etmenin hangi eyleme karar vereceğini belirten kuralları kullanan yüksek düzey etmenlere de sahip olabiliriz.

3. Etmenlerin olay işleme yeteneğinin desteklenmesi gerekir. Dış dünyadan veya diğer etmenlerden gelen olayları ve dış uygulamalardan gelen işaret olaylarını etmenin kontrol etmeye ihtiyacı vardır.

Olay çalışma çerçevesi tasarımı JavaBean bileşen modelinin gereksinimleri tarafından yönetilir. Bu modelin temeli, olay kaynaklarına ve olay

dinleyicilerine dayanır. Etmen bir olay dinleyicisi olabilir. Bir olay olduğu zaman etmen haberdar edilir, sonra etmen bu olayı işler. Olay nesne sınıfıyla etmen kendi olay türünü tanımlayabilir.

4. Etmene alan bilgisi if-then kuralları kullanılarak eklenebilmelidir. Algılayıcılar ve Etkileyiciler ile ileriye veya geriye doğru kural tabanlı çıkarım sağlanmalıdır.
5. Etmenler öğrenme algoritmalarını kullanarak sınıflama, kümeleme ve algılama yapmayı öğrenmelidir.
6. Çoklu etmen uygulamaları KQML benzeri mesaj protokolü kullanımını desteklemelidir.
7. Etmenler sürekli olmak zorundadır. Bir kere çalışması için oluşturulunca durumunu kaydetmeli ve daha sonra bu durumunu tekrar yüklemelidir [14].

6.2.3. Çerçeveler

Uygulama içinde üç tane çerçeve bulunur: CAgent, Rule ve PAManager. Hepsinin ayrı görevleri vardır. Java sınıflarından oluşmuş paketlerdir.

CAgent:

Çalışma çerçevesindeki etmen yöneticisidir yani tüm etmenler için genel programlama ara yüzeyi ve davranışı tanımlayan temel sınıftır. CAgent karma tasarım kullanır. Yani tek bir CAgent kullanılabileceği gibi gruplar içinde etmenler oluşturulabilir. Bu durumda grup yine tek mantıksal nesne olarak CAgent nesnesiymiş gibi davranacaktır. Bu tasarım modeli şu yönden güçlüdür; tanımlanmış diğer CAgent sınıflarını kullanan CAgent'ların hiyerarşisini kurmayı sağlar [14].

Rule:

Uygulama için mantık tabanlı çıkarım yapan ve kural tabanlı sistemlerde kullanılan iki ana çıkarım algoritması ileriye ve geriye doğru zincirlemeyi sağlar [14].

PAManager:

Kullanıcıya bazı görevleri yapmasında yardımcı olan ve zeki etmen kullanan kişisel yardımcıdır. Etmenlerin yaşadığı ve birbirleriyle çalıştığı çevre olarak hizmet eden bir etmen platformu sağlar. Burada geliştirilen etmenlerden ikisi olayları zamanlama ve olaylar meydana geldiği zaman kullanıcıyı haberdar etme gibi basit görevleri kontrol eden küçük çapta etmenlerdir. Üçüncüsü ise görev tabanlı zeki etmendir. Kullanıcının istediğine göre web sitelerindeki kitapların fiyatlarını ve indirimlerini kontrol eder. Sonra kullanıcının belirlediği indirim kriterleriyle uyuşan kitaplarla ilgili

olarak kullanıcıyı uyarır. Buradaki amaç, genel çoklu etmen platformu kavramlarını tanımlayan ve ek etmenler bağlayarak kolayca genişletilebilen esnek bir taban sağlamaktır. PAManager, etmenleri yöneten ve platformda çalışan diğer CIAgent'larla etkileşime izin veren, etmen diyaloglarından faydalanılması için grafik kullanıcı arayüzü sağlar. Kullanıcı PAManager içindeki etmenlerin çalışmasını başlatabilir, durdurabilir, tekrar kaldığı yerden çalıştırabilir, sonlandırabilir veya platformdan bir etmeni kaldırabilir.

6.3. Kişisel Etmen Yöneticisi

Bu kısımda uygulamadaki kişisel etmen yöneticisi olan PAManager paketi içinde oluşturulan etmenlerin özellikleri ve yaptıkları işler tanımlamaktadır. PAManager platformunda uygulamanın kullanıcı tarafından nasıl kullanıldığı ve PAManager çalışması anlatılmaktadır.

6.3.1. SchedulerAgent

Zeki bir etmen değildir fakat otonomdur, kullanışlı bir fonksiyona sahiptir. Belirli bir zamanda çalışması için uyarılarak kullanılabilir veya belirli bir zaman aralığında tekrar uyarılarak çalışması sağlanabilir. Bir uyarı meydana geldiği zaman kayıtlı dinleyicilerin hepsine CIAgent olayı gönderir. Sonuç olarak, belirli bir zamanda veya belirli bir aralık geçtikten sonra, etmenlere olay göndererek, diğer etmenlerin görevlerini zamanlayabilir [13].

6.3.2. UserNotificationAgent

Bu etmen çok basit bir etmendir. Haber mesajlarının görüntülenmesini sağlar. Bunları görüntülemek için merkezi bir kullanıcı arayüzü sağlar. Sistemdeki çoklu etmenler tarafından kullanılabilir. Bu durum tekrar tekrar fonksiyon ve kod yazılmasını önler. Çünkü kullanıcı için tek bir pencerede tüm mesajların görüntülenmesini sağlar [13].

6.3.3. KitapEtmen

Diğer ilk iki etmen kendi başlarına çalışabilirler ama zeki değildirler. KitapEtmen ise kullanıcının ilgilendiği kitap indirimi olduğu zaman karar vermek için kural tabanı ve ileriye zincirlemeyi kullanır.

İnternet üzerinden kitap almaya çalışırken birçok ilgili siteye girerek istediğimiz kitap hakkında bilgi toplarız. Sonra da karşılaştırma yaparız. Hangi site daha iyi indirim yapmışsa ondan kitap almak isteriz. Bu işleri yaparken bilgi toplama çok zamanımızı

alır. KitapEtmen bu aşamada bize yardımcı olur. Bizim için web sitelerini kontrol eder ve uygun bulduğumuz indirim kriterine uyan kitapları bildirir.

Bu uygulamada etmen için iki tane web sitesi seçilmiştir: www.kitap-net.com ve www.pandora.com.tr. Bu sitelere girdiğimiz zaman ana sayfalarında Şekil 6.1 de görülen arama alanlarını arama yapmak için kullanırız. Zeki etmen ise bunlarla ilgili sorgulama bağlantılarını kullanır: <http://arabul.kitap-net.com/arabul2.asp> ve <http://www.pandora.com.tr/ara.asp>. Bu sorgulama bağlantılarına aranılacak kelimeyle ilgili anahtar kelimeleri ve kelimenin kendisini yollar.



Şekil 6. 1 Kitap-Net Sitesi Ana Sayfası

Biz arama yaptığımızda Internet tarayıcısından Şekil 6.2 de görülen web sayfasındaki gibi arama sonuç bilgilerini alırız ve değerlendiririz. Etmen ise bu bilgilerin HTML formatı halindeki kaynak görüntüsünü satır satır okur. Yani Şekil 6.3 gibi arama sonuç bilgileriyle birlikte birçok HTML etiketlerini de alır. Etmen okuduğu kaynak kodlarını işler, değerlendirir ve istediği bilgileri çıkartır. Kural tabanında bilgileri kullanarak bir çıkarımda bulunur. Bulduğu sonucu kullanıcıya UserNotificationAgent kullanarak bildirir.

Aranan sözcük(ler) : özgür bir toplumda bilim
Bulunan kitap sayısı : 1
Alfabetik sıralama seçenekleri: [kitap adları] [yazarlar] [baskı tarihi]



Özgür Bir Toplumda Bilim

Paul Feyerabend; AYRINTI YAYINLARI; İnceleme

Bilim düşmanı! Akıl düşmanı! Anarşist!...Feyerabend, 1975 yılında yayımlanan Yönteme Karşı adlı yapıtıyla bu ve benzeri suçlamalara hedef olurken aynı zamanda geniş bir...

Çeviren Ahmet Kardan - 299 sayfa, 3. hamur, ISBN: 9755390057

Etiket 14.500.000 TL, %35 indirim 5.075.000 TL, İndirimli 9.425.000 TL



sepetime ekle

Şekil 6. 2 Kitap-Net Sorgulama Sonuç Sayfası

```
H=100% VALIGN=TOP ALIGN=CENTER>
LE border=0 cellspacing=0 cellpadding=5 WIDTH="100%">
<TR><TD WIDTH="80%">
t size="1" face="Verdana" color="#000044" style="FONT-FAMILY: Tahoma, Geneva; FONT-SIZE:
eneva; FONT-SIZE: 9pt" color="#000044"><B><A HREF="ayrinti2.asp?id=1609">Özgür Bir Toplu
<I>%35 indirim 5.075.000 TL</I>, indirimli 9.425.000 TL<BR></B></FONT><div align=right><
</TABLE>
</CENTER>

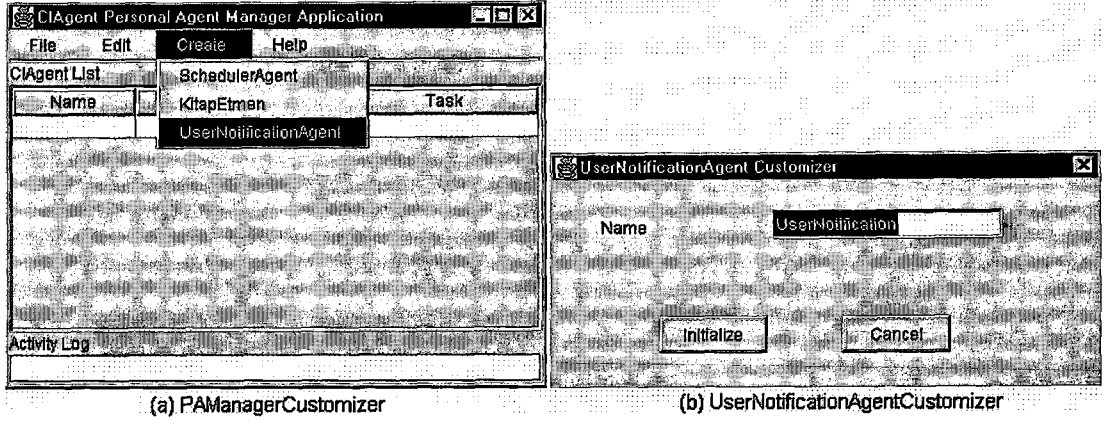
ORDER=0 WIDTH=100% cellspacing=0 cellpadding=0>
<TR>
<TD>
<CENTER>
<BR><BR>
<font size="3" face="Geneva, Arial"><B>Sosyal Yayınlar</B><BR>
<font size="1" face="Geneva, Arial">
Babıali Caddesi No:14 Cağaloğlu/İstanbul - Türkiye<BR>
Tel : (0212) 522 18 94 - (0212) 527 79 82<BR>
Fax : (0212) 527 79 82<BR>
Copyright (c) 1999,2000 Sosyal Yayınlar<BR>
e-posta için <A HREF="mailto:kitapnet@izlenim.com">kitapnet@izlenim.com</a>
```

Şekil 6. 3 Etmenin İşlediği HTML Kod Örneği

6.3.4. Uygulamanın Çalıştırılması

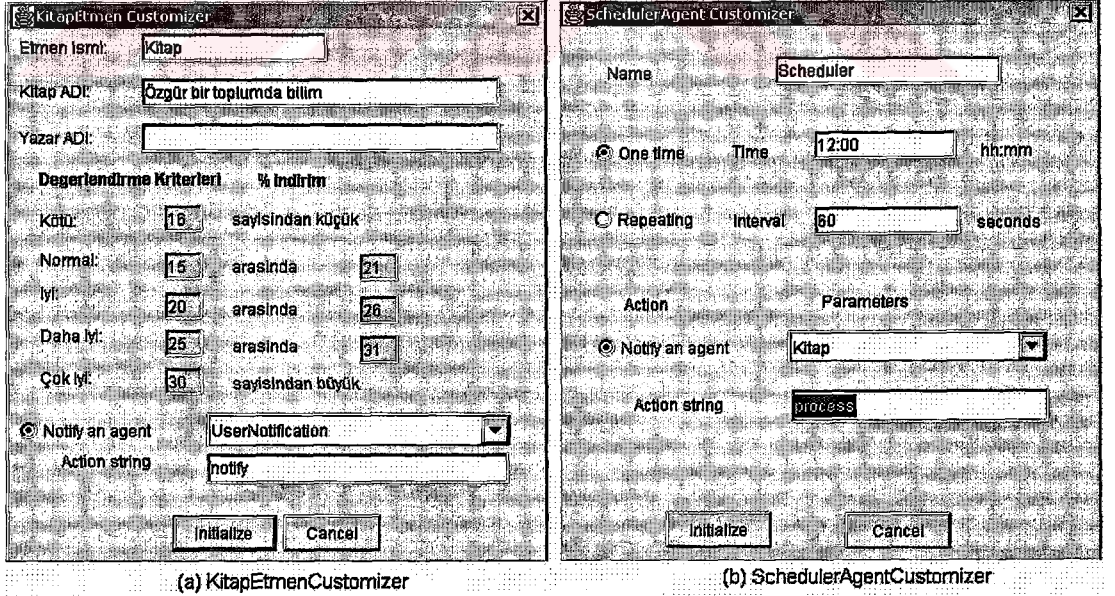
KitapEtmen, SchedulerAgent ve UserNotificationAgent ile kullanılabilir ve kullanıcıyla ilgili ne zaman bir kitap indirimi olursa kullanıcıyı bilgilendirir. Etmenler arasındaki ana etkileşimin nasıl sağlandığı Şekil 6.4'de gösterilmiştir. Uygulamanın çalıştırılması için PAManager platformunda şunlar yapılır:

1. PAManagerApp açılır ve çalıştırılır.
2. Create menüsünden önce UserNotificationAgent oluşturulmak için seçilir. Sonra Initialize tuşuna tıklanılır.(Şekil 6.4(a) ve Şekil 6.4(b))



Şekil 6. 4 UserNotificationAgent Çalıştırılması

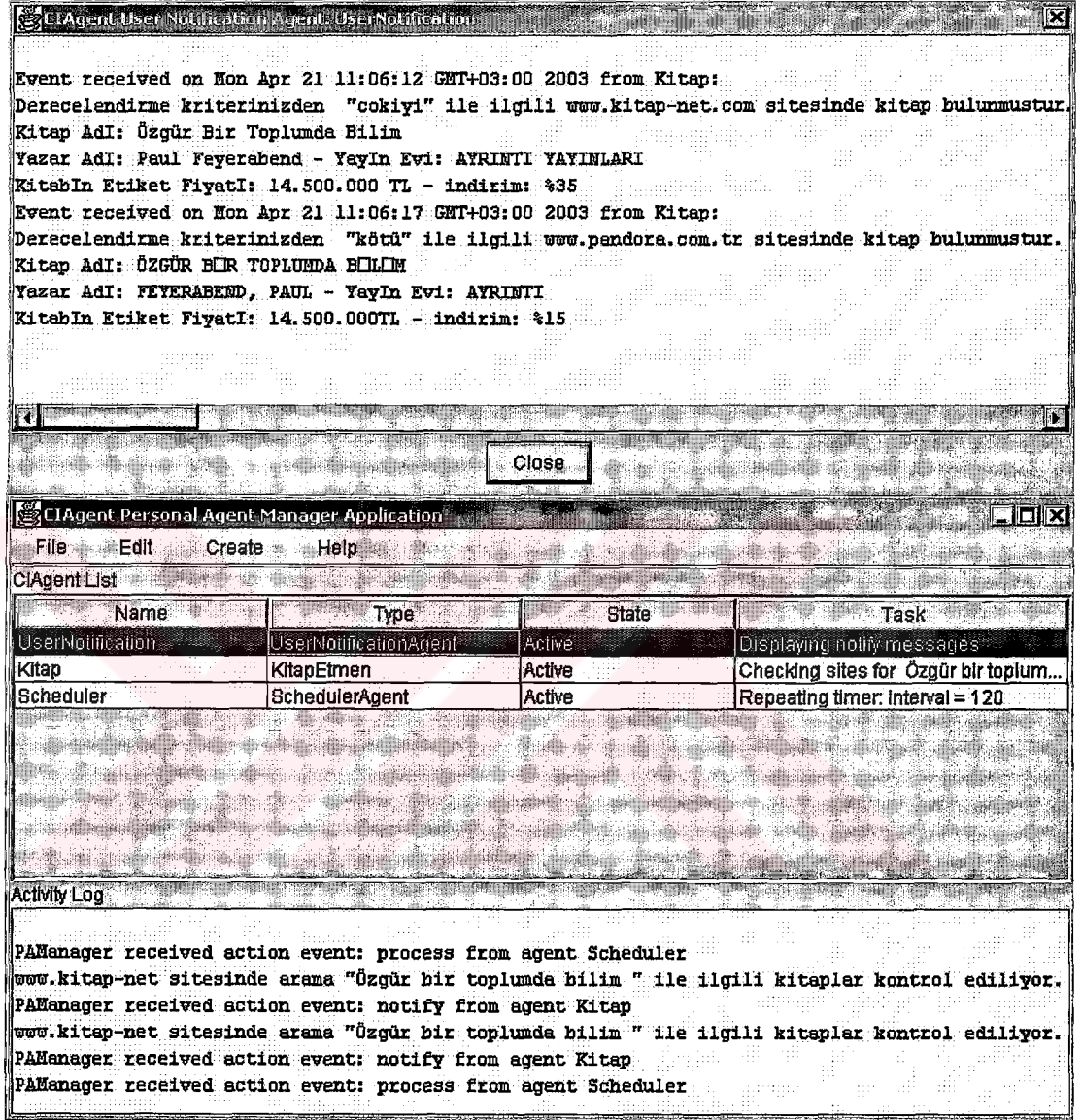
3. Create menüsünden KitapEtmen seçilir. Kitap adı ve indirim kriterleri girilir. Haber vermesi için Etmen olarak 2. Adımda oluşturulan UserNotificationAgent seçilir. Action string olarak bu yüzden notify girilir. Initialize tuşuna tıklanılır. Şekil 6.5(a)
4. Create menüsünden bu sefer SchedulerAgent seçilir. Bu etmen KitapEtmen'in istediğimiz zamanda çalışmasını sağlayacaktır. Bu yüzden uygun değerler girilir. Sonra Etmen kısmında KitapEtmen seçilir. Doğal olarak action string alanına da process girilir. Initialize tuşuna tıklanılır. Şekil 6.5(b)



Şekil 6. 5 KitapEtmen ve SchedulerAgent Çalıştırılması

5. PAManager platformunda yukarıdaki adımlarla oluşturulan üç etmen teker teker Edit menüsünden Start processing tıklanarak başlatılır.
6. Seçilen zamanda SchedulerAgent başlar ve KitapEtmen'e olay gönderir. KitapEtmen kullanıcının istediği çok iyi, daha iyi, iyi, normal, kötü indirim

kriterlerine uyan, girilen kitap ismine benzer, bilgi bulunduğunda UserNotificationAgent'a bir olay gönderir. UserNotificationAgent tarafından bilgi görüntülenir. (Şekil 6.6)



Şekil 6. 6 Bilgi Diyalogları

7. Etmenleri durdurmak ve uygulamayı sonlandırmak için File menüden Exit seçilir.

6.3.5. HTML Sorunu

HTML kodunda yazılmış olan web sayfalarının biçimi değiştirildiği zaman KitapEtmen'de yapılan okuma ve bilgi alma işlemleri çalışmayacaktır.

Bu durum web sayfası içinde bulunan bilgiye bakan ve buna göre hareket eden, web'i tarayan genel amaçlı etmen tasarımındaki en büyük engeldir. Sayfa üzerindeki bilgiyi almaya yardımcı olan sadece metinsel işaretlerin kullanılabilmesi ve bu sonuçların doğru sonuçları sağlayamaması sorun yaratır. Çünkü HTML sadece

içeriğın nasıl görüntöleneceğini belirtir. Sayfanın içeriğini ve anlamını belirten başka bir yola ihtiyaç vardır. Bu yollardan bir tanesi eXtensible Markup Language (XML) ile web sayfalarının tasarlanmasıdır.

XML, web sayfalarının içeriğini işaretleyen yeni nesil kodlamadır. XML etiketlerinin oluşturulmasına izin verir. Taşınabilirliği sağlamak için etiketlerin ne anlama geldiğini bildirecek bir yol belirler. Böylece oluşturulan belgeler başkaları tarafından okunur ve işlenebilir. Document Type Definition'lar (DTD) kullanılarak, tanıtmak için yeni eleman, eleman özellikleri ve elemanlar arasında ilişkiler belirtilebilir. DTD içinde belirtilen kurallarla, başkalarına zeki bir yolla XML belgesini okumaya yardımcı olunur ve XML okuyucular yazılabilir.

Ne yazık ki, XML halen İnternet ortamında çok yenidir. Birçok web içeriği HTML kullanılarak belirtilir, fakat az da olsa XML editörleri ve tarayıcıları bulunmaktadır. XML İnternet'de tamamıyla yaygınlaşınca, web üzerinde çalışan zeki etmenler geliştirmek çok daha kolay ve kullanışlı olacaktır [14].

7. SONUÇLAR VE TARTIŞMA

Tez çalışmasıyla kullanıcının yerine bazı görevleri yapan zeki bir etmen uygulaması geliştirildi. Bu zeki etmen, İnternet ortamında online satış yapan kitap sitelerine girmektedir. Kullanıcının belirttiği kitap konuları, yazar adları ve indirim kriterleri doğrultusunda sitelerde sorgulama yapmaktadır. Sitelerden elde ettiği sonuçları çıkarım mekanizmasında kullanarak kullanıcıya istediği bilgileri sunmaktadır.

Tez çalışmasında, zeki etmenlerle ilgili teorik bilgilerin yanında geliştirilen uygulamayla, bu tür uygulamaların nasıl oluşturulacağı ve hangi alanlarda kullanılabileceği anlatılmaya çalışılmıştır.

Oluşturulan uygulamanın benzerleri birçok bilgi toplamaya yönelik görevlerde kullanılabilir. Mesela, gün içinde çok sık değişen hisse senedi fiyatları veya döviz kurları bilgilerini toplayan ve değişimlerde kullanıcıyı uyaran zeki etmenli uygulamalar geliştirilebilir.

Uygulama geliştirme esnasında hayatımızın önemli bir parçası olan İnternet’de bilgi toplamanın, şimdi kullanılan bilgi içeriği görüntüleme teknolojisiyle, sanıldığı kadar kolay olmadığı görülmüştür. Programlamayla her sitede bilgi sorgulaması yapılamadığı ve yapılabilenlerde de farklı içerik gösterimleri yüzünden bilginin istenilen türde alınamadığı gözlenmiştir. Bu durum değişik sitelerde aynı sorgulama yaptırıldığı halde alınan bilgilerin farklı olmasından dolayı, bilgilerin karşılaştırılmasında büyük zorluklara sebep olmaktadır.

KAYNAKLAR

- [1] **Öztemel, E.**, 1996. Yapay Zeka Teknolojisi, Otomasyon Dergisi, **49**, 130–133.
- [2] **Weiss, G.**, 2000. Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence, Massachusetts Institute of Technology.
- [3] **Jennings, N. R., Wooldridge, M. and Sycara, K.**, 1998. A Roadmap of Agent Research and Development, *Int. Journal of Autonomous Agents and Multi-Agent Systems*, **1**, 7-38.
- [4] **Petrie, C.**, 1996. Agent-Based Engineering, the Web and Intelligence, *IEEE Expert*, **11**, 6, 24-29.
- [5] **Norman, D. A.**, 1997. How Might People Interact with Agents? *In Software Agents*, Ed. Bradshaw, J. M., AAAI Press.
- [6] **Kay, A.**, 1990. User Interface: A Personal View. *In the Art of Human Computer Interface Design*, pp. 191-208, Ed. Laurel, B., Addison-Wesley.
- [7] **Shoham, Y.**, 1993. Agent-Oriented Programming. *Artificial Intelligence*, **60**, 51–92.
- [8] **Sariel, S.**, 2002. Kullanıcı Kesitleriyle Yüz İfadelerini Analiz Eden Bir Çoklu Etmen Sistemi Uygulaması, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [9] **Brooks, R. A.**, 1991. Intelligence without reason. *In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, Sydney, IJCAI-91, 569–595, (in Australia).
- [10] **Brooks, R. A.**, 1991. Intelligence without reason. *In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, Sydney, IJCAI-91, 569–595, (in Australia).
- [11] **A Gentle Introduction to Agents**, <http://www.magma.ca/~mrw/agents/>
- [12] **Hayes-Roth, B.**, 1990. Architectural Foundation for Real-Time Performance in Intelligent Agents, Stanford University, Palo Alto, CA 94304, USA, 643-671.
- [13] **Jennings, N. R. and Wooldridge, M. J.**, 1998. Applications of Intelligent Agents, *in Agent Technology: Foundations, Applications and Markets*, pp. 3-28, Eds. Jennings, N. R. and Wooldridge, M. J.
- [14] **Bigus, J. P. and Bigus, J.**, 2001. Constructing Intelligent Agents using Java, Second Edition. John Wiley & Sons, Inc, Canada.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMAN

ÖZGEÇMİŞ

Gölnür BEKLERİÇ, 1975 yılında İstanbul'da doğdu. Lise eğitimi Üsküdar Kandilli Kız Lisesi'nde tamamladı. 1993 yılında İstanbul Teknik Üniversitesi Uzay Bilimleri ve Teknolojisi Mühendisliği Bölümüne girdi ve 1998 yılında mezun oldu. Aynı yıl İstanbul Teknik Üniversitesi Uzay Mühendisliği Anabilim Dalı, Uzay Mühendisliği Programında yüksek lisansa başladı.

