

# Anlamsal Web Ortamında Çalışacak Çok-Etmenli Sistemler için bir Referans Mimarisi

Geylani Kardeş<sup>1</sup>, Oğuz Dikenelli<sup>2</sup>

<sup>1</sup> Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, 35100, Bornova, İzmir

<sup>2</sup> Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, 35100, Bornova, İzmir

<sup>1</sup>geylani.kardas@ege.edu.tr, <sup>2</sup>oguz.dikenelli@ege.edu.tr

**Özet.** Yazılım etmenleri ve bunların oluşturduğu çok-etmenli sistemler (ÇES), karmaşık yapıdaki dağıtık sistemlerin modellenmesini ve oluşturulmasını sağlayan etkili birer teknoloji olarak ortaya çıkmışlardır. Anlamsal Web evrimi de şüphesiz etmen araştırmalarına yeni bir vizyon getirmiştir. Bu ikinci nesil Web, web sayfası içeriklerini, ontolojiler kullanılarak yorumlanabilecek bir seviyeye taşımayı hedeflemektedir. Söz konusu yorumlamanın ve anlam çıkarsamaların özerk etmenler tarafından insanlar adına yerine getirileceği düşünülmektedir. Özellikle anlamsal web servisi gibi Anlamsal Web yapıları ile yazılım etmenlerinin etkileşiminin ÇES'ler tasarlanıp uygulamaya geçirilirken dikkate alınması ve sistem mimarilerinin de ilgili ihtiyaçlara cevap verecek bileşenlere sahip olması gerekmektedir. Bu bildiride Anlamsal Web ortamında çalışacak ÇES'ler için bu amaca yönelik katmanlı bir referans mimarisi önerilmektedir. Mimari etmenlerin iletişim, içsel planlama mekanizmaları ve bağlı oldukları platform bünyesinde sundukları veya hizmet aldıkları servislere yönelik sistem bileşenlerini tanımlamakta ve ilişkilerini ortaya koymaktadır.

## 1 Giriş

Yazılım etmenleri ("*software agents*") kullanıcılarının adına bir takım görevleri yerine getirmek üzere davranma yeteneği olan özerk ("*autonomous*") yazılım bileşenleri olarak tanımlanmaktadır. Öte yandan bir çok akıllı yazılım etmeninin bir araya gelerek oluşturdukları ve kendi bilgi ve bireysel yeteneklerini kullanarak çözemedikleri veya etkin bir biçimde çözemeyeceklerini düşündükleri problemlerini çözmek amacıyla birbirleri ile etkileşimde buldukları sistemler ise Çok-etmenli Sistem ("*Multi-agent System*") (ÇES) adını almaktadır [1]. Söz konusu etmen etkileşimleri [2]'de belirtildiği gibi bencil veya işbirlikçi bir yapıda olabilir. Başka bir deyişle etmenler ortak bir amacı paylaşabilir ya da serbest piyasa ekonomisinde olduğu gibi kendi çıkarlarının takipçisi olabilirler.

Weyns ve Holvoet, [3]'te bir ÇES'in belirli bir problemi çözmek için gerekli yazılımı sağladığını belirtmektedirler. Bu amaç doğrultusunda ÇES ilgili sistemi birbirleriyle etkileşim halinde olan bir dizi özerk varlıklar halinde yapılandırmakta ve sistemin işlevi ve kalitesine yönelik ihtiyaçları karşılamaktadır. Öte yandan [4]'te bir *yazılım mimarisi*, yazılım elemanlarını, bu elemanların görünür özelliklerini ve elemanlar arasındaki ilişkileri içeren bir sistemin yapı veya yapıları olarak tanımlanmaktadır. Yazılım elemanları (ya da daha genel bir deyişle mimariye ait elemanlar) sistemin işlevselliğini sağlarken ihtiyaç duyulan sistem kalite özellikleri yazılım mimarisinin yapıları

üzerinden karşılanmaktadır. [3]'teki bakış açısı dikkate alındığında ÇES'ler ile yazılım mimarisi arasında çok yakın bir ilişkinin olduğu söylenebilir. Çünkü bir ÇES asıl hedefini yerine getirmek için ne yapıyorsa bir yazılım mimarisi de onu içermektedir.

Literatürde ÇES'lerin mimari özelliklerini göz önüne alan çalışmaların yanında ÇES mimarilerine organizasyonel perspektifte, ilgi yönelimli ("*aspect-oriented*") veya model tabanlı öneriler sunan çeşitli çalışmalar (örneğin [5], [6], [7], [8], [9] ve [10]) bulunmaktadır. Her ne kadar söz konusu bu çalışmalar ilgili alana önemli katkılar sağlamış olsa da yakın gelecekte etmenlerin üzerinde çalışacağı düşünülen Anlamsal Web ("*Semantic Web*") ortamı ve ÇES'lerin bu ortam üzerinde çalışabilmesi için ihtiyaç duyulan yapıların bu çalışmalarda desteklenmediği gözlenmiştir.

Anlamsal Web evrimi [11] şüphesiz etmen araştırmalarına yeni bir vizyon getirmiştir. Bu ikinci nesil Web, Dünya Geneli Ağ'ı (WWW) web sayfası içeriklerinin ontolojiler kullanılarak yorumlanabileceği bir seviyeye taşınmayı hedeflemektedir. Söz konusu yorumlamanın ve anlam çıkarsamaların özerk etmenler tarafından insanlar adına yerine getirileceği düşünülmektedir.

Anlamsal Web ortamının kendine özgü mimari varlıklarının ve farklı bir anlamsal yapısının olduğu, bu ortam üzerinde çalışacak ÇES'ler hazırlanırken göz önünde bulundurulmalıdır. Etmen mimarilerinin, modelleme tekniklerinin ve ÇES yazılımı geliştirme çerçevelerinin bu yeni ortamı desteklemesi gerekmektedir. Bu düşünceden hareketle bu bildiride Anlamsal Web ortamında çalışacak ÇES'ler için bir referans mimari tanımlanmaktadır. Önerilen mimari, bünyesinde bir etmenin Anlamsal Web ortamında hem diğer etmenler hem de Anlamsal Web ortamına özgü anlamsal web servisleri ile etkileşimine ait servis, etmen planlama ve iletişim seviyesinde görev alan mimari bileşenlerini içermektedir. Böyle bir mimariye dayalı olarak hayata geçirilen ÇES'ler Anlamsal Web yetenekli olacaklar ve bu ÇES'lerde yer alan yazılım etmenleri de kullanıcıları adına Web içeriğini farklı kaynaklardan elde edebilecek, bilgiyi işleyebilecek ve sonuçları değiş tokuş edebileceklerdir. Ayrıca özerk etmenler bu tip ÇES'ler içerisinde anlamsal veriyi değerlendirebilecek ve içerik dilleri vasıtasıyla anlamsal web servisleri gibi anlamsal ortam elemanları ile etkileşimlerde bulunabileceklerdir.

Bir referans mimarisi, benzer özelliklere ve sistem ihtiyaçlarına sahip bir dizi uygulamanın tasarımı ve geliştirilmesi sırasında elde edilen deneyimler sonucunda şekillenmektedir [12] ve sunduğu ortak zemini paylaşan yeni yazılım mimarilerinin geliştirilmesinde yazılım mimarlarına fayda sağlamaktadır. Bildiride tanımlanan mimarinin de bu amaç doğrultusunda Anlamsal Web yetenekli ÇES'ler için somut mimariler hazırlanırken kullanılacak bir referans mimarisi olacağına inanılmaktadır.

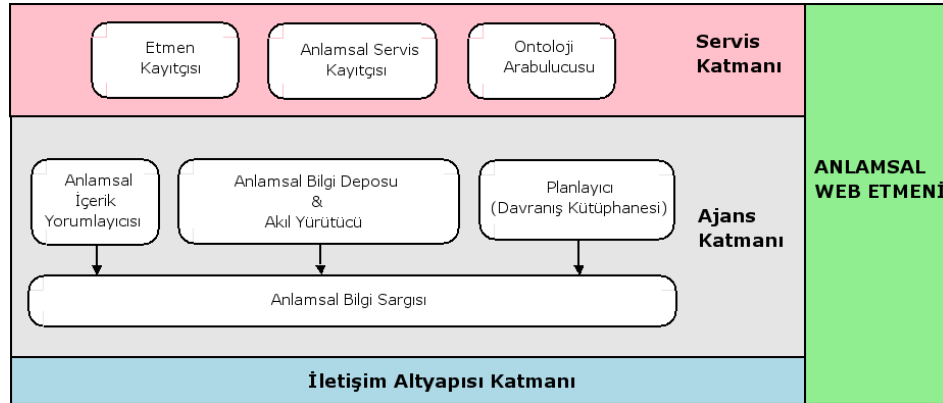
Bildirinin geriye kalan kısmı şu şekilde düzenlenmiştir: Bölüm 2'de Anlamsal Web ortamında çalışacak ÇES'lere ait referans mimarisi anlatılmaktadır. Bölüm 3'te ÇES'ler için yazılım mimarisi öneren literatürdeki diğer çalışmalar özetlenmiştir. Elde edilen sonuçlar ve hedeflenen ileriye yönelik çalışmalar ise Bölüm 4'te yer almaktadır.

## 2 Anlamsal Web Etmenleri için Referans Mimarisi

Anlamsal Web'in gerçek gücü Web içeriğini farklı kaynaklardan toplayabilen, elde ettiği bilgileri işleyebilen ve çıkarsadığı sonuçları başka ortam elemanları ile de paylaşabilen bilgisayar programlarının geliştirilmesiyle ortaya çıkacaktır [11]. Söz konusu bu bilgisayar programları yazılım etmenleridir. Yazılım etmenlerinin etkinliği makineler tarafından anlaşılabilen Web içeriği daha fazla hazır hale geldikçe ve kullanılabilir, otomatikleştirilmiş ("*automated*") servislerin sayısı çoğaldıkça artacaktır.

Yukarıda sözü edilen ortamda çalışacak etmen yazılımlarının tasarlanması ve kullanılmasına yönelik bir yazılım mimarisinin ortaya konulması gerekmektedir [13]. Bu amaç doğrultusunda önerdiğimiz referans mimari Şekil 1'de verilmiştir.

Önerilen referans mimari, [14]'te de anlatılan *Modül Bakış Tipi* ("*Module Viewtype*") göz önüne alınarak hazırlanmıştır ve katmanlı bir yapıya sahiptir. Katmanlar arası ilişki *kullanıma izinli* ("*allowed-to-use*") adı verilen ilişki gösterim tipindedir. Clements ve ark.'nın [14]'te verdikleri tanıma göre aralarında bu ilişki olan iki katmandan ilkinde yer alan herhangi bir modül ikinci katmanda yer alan herhangi bir modülü kullanma hakkına sahiptir. İlişkinin yönü aşağıya doğrudur. Bunun anlamı önerilen mimaride sadece üst seviye bir katman alt seviyedeki bir katmanın sunmuş olduğu servis veya hizmetleri kullanabilir. Tersine izin verilmemektedir. Öte yandan önerilen mimari herhangi bir *katman köprülemeyi* ("*layer bridging*") [14] de içermemektedir. Buna göre bir üst katman sadece bir sonraki alt katmanın modüllerini kullanabilir. Mimariyi resimlemek için kullandığımız gösterim Şekil 1'de görüldüğü üzere yan eklentili katmanları ("*layers with a sidecar*") [14] içermektedir. Katmanlar arası *kullanıma izinli* ilişkisi ise şekilde geometrik komşuluklarla temsil edilmektedir.



Şekil 1. Anlamsal Web ortamında çalışan ÇES'ler için katmanlı bir referans mimarisi

Önerilen mimarinin ana kısmında üç adet katman tanımlanmıştır: *Servis Katmanı*, *Ajans* ("*Agency*") *Katmanı* ve *İletişim Altyapısı Katmanı*. *Anlamsal Web Etmeni* ise bu

mimari katmanlarında bulunan tüm modülleri (bileşenleri) kullanma hakkına sahip mimarının eklenti (“*sidecar*”) bileşenidir. Bu mimariye uygun olarak geliştirilen bir ÇES’te bir grup etmen Servis Katmanı’nda tanımlı servisleri sunar. Sistemdeki her etmenin ise Ajans Katmanı’nda tanımlı bir etmen iç yapısı bulunmaktadır. Sistemdeki etmenler birbirleri ile İletişim Altyapısı Katmanı’nda tanımlanmış olan protokollere uygun olarak haberleşirler. Söz konusu bu mimari katmanları aşağıdaki alt bölümlerde detaylı olarak anlatılmaktadır.

## 2.1 Servis Katmanı

Servis Katmanı’nda bir ÇES’te yer alan anlamsal web etmenlerinin servisleri (ve/veya rolleri) tanımlanmaktadır. Servis Katmanı’ndaki tüm servisler Ajans Katmanı’nın sunmuş olduğu imkanları kullanırlar. İlgili etmen sisteminin iş alanına özgü etmen servisleri haricinde bu katmanda, etmen sistemine sağlanması gereken sarı sayfa ve arabulucu servisleri yer almaktadır.

*Etmen Kayıtçısı*, etmen platformunun diğer üyeleri için sistemde yer alan etmenlerin yeteneklerinin anlamsal olarak tanımlandığı ve ilan edildiği bir sistem servsidir. Görevlerinin işletimi sırasında platform etmenleri diğer etmenler tarafından sunulan servislere ihtiyaç duyabilir. Bu nedenle söz konusu bu servis üzerinde sorgular işletirler ve etkileşim için uygun etmenleri belirlerler.

Geleneksel bir ÇES, sistemin üyesi olan etmenlerin uygun etmen servislerini bulabilmesi amacıyla sarı sayfa hizmeti sunan bir ya da daha fazla kayıtçıya sahiptir. Örneğin FIPA<sup>1</sup> soyut mimari tanımında etmenlerin sunduğu servislerin kayıt olduğu, *Dizin Kolaylaştırıcısı (DK)* adı verilen ve her FIPA uyumlu ÇES’te olması gereken bir etmen çeşidi bulunmaktadır [15]. Bir etmen spesifik bir etmen servisini aradığında DK’dan servisi sağlayan etmenin bilgilerini (örneğin etmenin adı, adresi, vb.) elde etmekte ve görevini tamamlamak üzere ilgili servisi sağlayan etmenle iletişime geçmektedir.

FIPA uyumlu olsun ya da olmasın etkileşim içerisindeki etmenleri içeren bir ÇES’te yukarıda tarif edilen etmen kayıtçıların olması gerekliliği açıktır. Ancak etmenlerin talep ettikleri ve sundukları servislerin (bir anlamda etmen yeteneklerinin) eşlenmesi işlemi Anlamsal Web ortamında çalışacak ÇES’ler düşünüldüğünde daha karmaşık bir yapıya bürünmektedir ve yeniden tanımlamaya ihtiyaç duymaktadır. Bu tip ÇES’lerde etmen servislerinin keşfi için servis yeteneklerinin anlamsal eşlenmesine ait kriterlerin tanımlanması ve etmen servisi tanımlarının kaydedilme mekanizmalarının (bir anlamda dizin servislerinin) bu kriterlere uygun olarak tasarlanması gerekmektedir. Böylelikle aranan servis özellikleri ve ilan edilen servislerin yetenekleri arasındaki eşleme işlemi sadece özdeş (“*identical*”) servis eşlemeyi göz önünde bulundurmayarak daha etkin bir hale gelmektedir. Buradaki özdeş servis eşleme ile kastedilen standart dizin servislerinde yer alan anahtar kelime bazlı servis arama ve eşleme işlemidir. Oysa yeni

---

<sup>1</sup> FIPA (“Foundation for Intelligent Physical Agents”), etmenler ve etmen tabanlı sistemler arasında birlikte çalışabilirliği desteklemek amacıyla çeşitli standartlar ortaya koyan bir kuruluştur. Bu standartlara dayalı olarak tanımlanmış olan FIPA ÇES platformu literatürde üzerinde en çok çalışılan ve desteklenen ÇES platformudur. FIPA ÇES platformu ve FIPA standartları hakkında ayrıntılı bilgiye <http://www.fipa.org> adresinden erişilebilir.

yetenek eşleme süreci aranan ve ilan edilen iki servis arasındaki ilişkinin tipini ve derecesini *anlamsal olarak* belirleyecektir ve bu da etmen ihtiyaçlarını karşılayan en uygun servislerin bulunmasını sağlayacaktır. Tüm bu vizyona uygun olarak referans mimarisi etmen servisleri üzerinde yetenek eşlemesini yerine getirecek bir etmen kayıtçısını Servis Katmanı'nda tanımlamaktadır. Anlamsal yetenek eşlemesi ile ilgili detaylar bu bildirinin kapsamı dışındadır ancak FIPA uyumlu etmen sistemleri için örnek bir anlamsal yetenek eşleme mekanizması [16]'da anlatılmaktadır.

Öte yandan bir ÇES'te yer alan Anlamsal Web yetenekli yazılım etmenleri, yerine getirmek istedikleri görevlerinin işletimi sırasında eğer ihtiyaç hissedersen *anlamsal web servisleri* ile de etkileşime geçebilirler. Anlamsal web servisleri, bulunmaları ve otomatik olarak işletilmeleri için anlamsal bir ara yüze sahip web servisleri olarak tanımlanabilirler.

Günümüzde kullanılmakta olan web servisleri kendilerini temsil eden ve *Web Servisleri Tanımlama Dili* ("*Web Services Description Language - WSDL*") kullanılarak hazırlanan ara yüzleri sayesinde geliştirildikleri yazılım dili ve/veya ortamına bağlı kalmaksızın yine çok çeşitli ortamlarda çalışan istemci yazılımlar tarafından kullanılabilirler. Bir istemci program WSDL'i işleyerek, sunulan servise ait işlemi ve bu işlem için gerekli girdi – çıktı parametrelerini öğrenir ve servisi WSDL gibi yine web servisleri için bir standart olan *Basit Nesne Erişim Protokolü* ("*Simple Object Access Protocol - SOAP*") kullanarak çalıştırabilir. Ancak bu mevcut web servis altyapısı sadece sözdizimsel birlikte işlerliği göz önünde tutmaktadır ve [17]'de belirtildiği gibi böyle bir yaklaşım ne anlamsal birlikte işlerliği ne de web servislerinin otomatik tümleşimini mümkün kılar. Söz konusu birlikte işlerliği ve tümleşimi sağlamak amacıyla web servislerinin yeteneklerinin servis ontolojilerinde tutulması ve bu ontolojiler kullanılarak ihtiyaca en uygun servislerin bulunmasına ve dinamik çağrımının gerçekleştirilmesine çalışılmaktadır.

Servis yeteneklerinin tanımlanması ve servis çalıştırma sürecinin ifade edilmesine yönelik *Servisler için Web Ontoloji Dili* ("*OWL for Services - OWL-S*") [18] ve *Web Servis Modelleme Ontolojisi* ("*Web Service Modeling Ontology - WSMO*") [19] gibi çeşitli anlamsal web servis tanımlama ve kullanma ontolojileri literatürde bulunmaktadır. Bizim bakış açımıza göre, bu şekilde tanımlanmış servis yeteneklerinin de tıpkı etmen servisleri için olduğu gibi uygun kayıtçılarda tutulup ilan edilmesi gerekmektedir. Böylelikle bu servisler de etmenler tarafından dinamik olarak keşfedilecek ve ihtiyaçları doğrultusunda çalıştırılabilirlerdir. Önerdiğimiz mimaride bu amaca uygun olarak *Anlamsal Servis Kayıtçısı* adı verilen bir Servis Katmanı bileşeni tanımlanmıştır.

Anlamsal Servis Kayıtçısı, ilgili platforma ait anlamsal web servislerinin ara yüzlerini bu servislerin etmenler tarafından keşfedilmesi amacıyla ilan eden bir servis eşleyici olarak modellenmelidir. Örneğin OWL-S servislerini göz önüne alacak olursak, bir etmen ihtiyaç duyduğu anlamsal servisin yeteneklerini belirten OWL-S profilini bu kayıtçıya göndererek bu kayıtçı üzerinde ilgili sorgunun işletilmesini sağlar. Anlamsal Servis Kayıtçısı verilen ihtiyaç profili ve ilan ettiği servis profilleri arasında bir anlamsal yetenek eşleme işlemi gerçekleştirerek etmenin işine yarayacak uygun servisleri ilgili etmene bildirir. Etmen de uygun olan bu servis (ya da servislerle) anlaştıktan sonra görevini tamamlamak üzere ilgili servis (ya da servislerle) etkileşimde bulunabilir. Söz

konusu anlaşma ve servis çalıştırma işlemi yine anlamsal olarak tanımlanmış etkileşim protokollerine uygun olarak yerine getirilmektedir. Anlamsal web servisleri için yetenek eşleme hakkında detaylı bilgiye ve örnek bir eşleme mekanizmasına [20]'den erişilebilir. Anlamsal web servislerinin ÇES'ler bünyesinde kullanımı ve yukarıda sözü edilen Anlamsal Servis Kayıtçısı'nın somut bir uygulaması da [21]'de anlatılmıştır.

Hem Etmen Kayıtçısı hem de Anlamsal Servis Kayıtçısı bir ÇES'te servis eşleyici yerine birer aracı ("*broker*") olarak da uygulamaya geçirilebilirler. Bu durumda ilgili kayıtçılar sadece servis yetenek eşlemesini gerçekleştirmezler; buna ek olarak servise ihtiyaç duyan etmen adına servisle bizzat etkileşime geçerler ve servis çalıştırma sonucunu servisi talep eden etmene yönlendirirler.

Bir anlamsal web etmeni farklı etmen organizasyonlarındaki etmenlerle etkileşimde bulunma ihtiyacı hissedebilir. Ayrıca bu etmenlerin ve anlamsal web servislerinin farklı bilgi depolarında ya da dağıtık sistemlerde yer alan bilgi kaynaklarını kullanması gerekebilir. ÇES'ler gibi açık sistemlerde bu durumlar nedeniyle birden fazla ontoloji yer alabilir ve farklı sistem elemanları farklı ontolojileri kullanabilir. Bu nedenle farklı ontolojilerin kavram dönüşümlerini ve eşlemelerini yerine getirecek servislerin ÇES'lerde olması gerektiğine inanmaktayız. Şekil 1'de görüldüğü gibi referans mimaride bu servisi sağlayan bileşene *Ontoloji Arabulucusu* adı verilmiştir. Bir ÇES bünyesinde bir ya da daha fazla Ontoloji Arabulucusu yer alabilir.

Bir Ontoloji Arabulucusu aynı zamanda ÇES iş alanına ait ve ilgili platform içerisindeki elemanlarca kullanılan ontolojiler için merkezi bir veri havuzu görevini üstlenebilir ve ontoloji yükleme, ontoloji güncelleme ve ontolojiler üzerinde sorgu gerçekleştirme gibi temel ontoloji yönetim işlemlerini yürütebilir. Ontoloji Arabulucusu servisini sağlayan bir etmen ontoloji çevrim isteklerini özel bir kullanıcı ara yüzü vasıtası ile önceden tanımlanmış kavram eşleme bilgisine göre karşılamaktadır. Sağlanan bu ontoloji çevrim desteği ile bir etmen kendisinin üyesi olmadığı başka bir ÇES'te yer alan bir etmenle ya da kendi ortamı dışında yer alan bir servis ile farklı ontolojiler kullansalar bile iletişimde bulunabilir.

## 2.2 Ajans Katmanı

Referans mimarinin orta katmanı olan Ajans Katmanı anlamsal web etmenlerinin iç yapısını tanımlamaktadır. Özerk ve karşıt eylemli ("*reactive*") yapıdaki etmenlerin amaçlarına uygun olarak dış ortam bileşenlerini nasıl kullanacağı ve davranışları için nasıl plan yapacağı bu iç yapıya göre belirlenmektedir.

Sistemdeki her etmenin yerel ontolojilerini sakladığı bir *Anlamsal Bilgi Deposu* bulunmaktadır. Bu ontolojiler etmen platformundaki diğer etmenler veya servisler ile etkileşime geçerken kullanılmaktadır. Ontolojilerin değerlendirilmesi ve temel çıkarsama ise *Akıllı Yürütücü* modülü tarafından yerine getirilir.

*Anlamsal İçerik Yorumlayıcısı* etmen iletişimini kontrol eder. Bir anlamsal web etmeni iletişimleri sırasında diğer etmenlerden veya anlamsal servislerden doğal olarak mesajlar alacaktır. Alınan mesaj içeriğinin anlamsal uygunluğunun kontrol edilmesine ve içeriğin etmenin inançlarına ve niyetlerine uygun bir şekilde yorumlanmasına ihtiyaç vardır. Anlamsal İçerik Yorumlayıcısı söz konusu bu içerik uygunluğunun kontrolünü ve yorumlamayı gerçekleştirir.

Ajans Katmanı'nın *Planlayıcı* adı verilen modülü ihtiyaç duyulan yeniden kullanılabilir etmen planlarını ve ilgili davranış kütüphanelerini içermektedir. Yeniden kullanılabilir etmen planları bir etmenin niyetlerine uygun olarak işletilen görevlerinin birleşiminden oluşmaktadır. Planlayıcı, örneğin Hiyerarşik Görev Ağı ("*Hierarchical Task Network – HTN*") [22] gibi bir karşıt eylemli planlama ("*reactive planning*") paradigmasını temel almaktadır. Dickinson ve Wooldridge'in [23]'te belirttiği gibi karşıt eylemli planlama için bir etmene önceden tanımlı (belki de sistemin derlenme zamanında tanımlanmış) genel planlardan oluşan bir kütüphane sağlanmaktadır. Etmen ortamdan elde ettiği algılara tepki olarak bu planların birini ya da birkaçını uygulamaya geçirir.

Öte yandan bir yapay zeka planlama metodolojisi olan HTN planlama, görevlerin ayrıştırılması ("*decomposition*") ilkesine bağlı olarak planların ortaya konmasını sağlar. Bu ilkeye göre doğrudan çalıştırılabilen görevlerin oluşturduğu, hiyerarşide en üstte yer alan ana bir plan vardır. Planlama sistemi doğrudan çalıştırılacak en temel görevleri bulana kadar bu plan ayrıştırılmakta ve etmen bu görevleri yerine getirmektedir. Örneğin bir etmenin anlamsal web servisleri ile etkileşimi servis keşfi, servisle anlaşma ve servisi çalıştırma görevlerinin birleşiminden oluşan yeniden kullanılabilir bir plan olarak modellenebilir. Anlamsal web servisini kullanabilmek için etmen bu görevler kapsamında tanımlanan işlemleri yerine getirir.

Ajans Katmanı'nda bulunan *Anlamsal Bilgi Sargısı* ("*Semantic Knowledge Wrapper*") yukarıda sözü edilen ontolojilerin Ajans Katmanı'nın üst seviye bileşenleri tarafından kullanılabilmesini sağlamaktadır. Örneğin görev işletimi sırasında bir etmen bir ontoloji varlığının nesne (ya da başka bir programlanabilir yapı) gösterimine ihtiyaç duyabilir. Anlamsal İçerik Yorumlayıcısı da etmen ontolojileri üzerinde sorgu işleterek bir konu hakkında çıkarsamada bulunmak isteyebilir. Bu tip ihtiyaçları gidermek amacıyla Anlamsal Bilgi Sargısı, Ajans Katmanı'nın çalışma zamanı ortamı içerisinde ilgili ontolojilerin çizge gösterimlerini oluşturabilir. Etmen iç mimarisinde böyle bir sargının kullanılmasına dair bir örnek JENA<sup>1</sup> çerçevesine dayalıdır ve [24]'te anlatılmıştır.

### 2.3 İletişim Altyapısı Katmanı

Mimarinin en alt katmanı mimarinin iletişim altyapısı uygulamasının soyutlanmasından sorumludur. Referans mimarisinin somut bir örneğinde bu katman FIPA Etmen İletişim ve Etmen Mesaj Taşıma protokollerinin bir uygulaması olabilir. Böylece altyapı FIPA Etmen İletişim Dili'nin ("*Agent Communication Language - ACL*") ve ilgili protokolün kullanılması ile etmenler arası mesaj transferini gerçekleştirmiş olur. Fiziksel iletim iyi bilinen HTTP-IIOP ("*HTTP - Internet Inter-ORB Protocol*") üzerinden gerçekleştirilebilir. Burada asıl önem verilmesi gereken mesaj altyapısında kullanılan içerik dilinin yapısı ve zenginliğidir.

---

<sup>1</sup> JENA, Anlamsal Web uygulamalarının geliştirilmesini sağlayan ve Java programlama dili kullanılarak hazırlanmış açık kaynak kodlu bir çerçevedir. Ontolojilerin kullanılması için programlanabilir bir ortam sağlamaktadır ve kural tabanlı bir çıkarsama motoru içermektedir. Tüm çerçeveye <http://jena.sourceforge.net/> adresinden erişilebilir.

### 3 İlgili Çalışmalar

Etmen araştırmacılarının literatürde ÇES'ler için önerdikleri mimari çalışmalarının farklı bakış açılarına sahip olduğu gözlenmektedir. [5]'teki çalışmada Shehory, ÇES'lerin yazılım mühendisliğindeki rollerini değerlendirmekte ve tasarımcıların bir problemin çözümü için önerilen bir ÇES'in uygunluğunu değerlendirmede kullanabilecekleri başlangıç seviyesindeki mimari özellikleri sunmaktadır. Sunulan bu özellikler ÇES'leri bir yazılım mimarisi stili olarak karakterize etmeyi sağlamaktadırlar. Öte yandan ÇES mimarilerine organizasyonel perspektiften bakan [6]'daki çalışmada organizasyon yönetimi teorisi kavramlarını içeren bir dizi mimari stili önerilmiştir. İlgili ÇES mimari stilleri aktör, görev ve aktör bağımlılığı gibi ÇES kavramlarını ön planda tutan bir çerçeveye göre modellenmişlerdir.

[25]'te tanıtılan PROSA, imalat sistemlerini göz önüne alan koordinasyon ve kontrol uygulamaları için bir referans mimarisidir. Kaynak etmeni, ürün etmeni ve sipariş etmeni adı verilen üç temel etmen tipi üzerine kurulu PROSA'da etmenler ve aralarındaki ilişkileri belirlemede nesne yönelimli kavramlar kullanılmıştır. Weyns ve Holvoet de [7]'de yerleşik ("*situated*") ÇES'ler için bir referans mimarisi önermişlerdir. Söz konusu mimari ilgili araştırmacıların üzerinde çalıştıkları çeşitli robotik uygulamalarının ortak fonksiyonları ve yapıları belirlenerek oluşturulmuştur ve bu uygulamalar sonucunda elde edilen deneyimi yansıtmaktadır.

FIPA'nın ÇES'ler için önerdiği FIPA Soyut Mimarisi [15] farklı ileti taşıma protokolleri, farklı etmen iletişim dilleri ve farklı içerik dilleri kullanan etmenler arasında anlamlı ileti alışverişini sağlamayı amaçlamaktadır. Önerilen soyut mimari, somut mimarilerin geliştirilmesinde temel olarak alınmaktadır. Bir somut mimarinin FIPA uyumlu olabilmesi için, etmenleri kaydeden, etmenleri bulan ve etmenler arası ileti transferini gerçekleştiren mekanizmalara sahip olması gerekmektedir.

Garcia ve ark., [8]'de özerklik, öğrenme ve taşınabilirlik gibi etmen özelliklerinin örneğin [9]'da olduğu gibi klasik mimari desenlerini uygulayan yaklaşımlarla karşılanamadığını savunmuş ve mimarilerin yeniden kullanılabilirliğinin ve yönetiminin varolan bu yaklaşımlarda oldukça zor olduğunu belirtmişlerdir. Bu nedenle etmen mimarilerini yapılandırmak için ilgi yönelimli bir yaklaşım önermişlerdir.

ÇES'ler için üstmodeller tanımlayıp bu modeller arası dönüşümler sonrası etmen yazılımlarını elde etmeyi hedefleyen model güdümlü etmen mimari çalışmaları da bu alanda öneme sahiptir. [10], [26] ve [27] gibi önerilerde ortam bağımsız ya da belli bir ortama uygun ÇES'lerin model güdümlü olarak geliştirilmesi amaçlanmaktadır.

Bildirinin giriş bölümünde de belirtildiği gibi yukarıda değinilen bu önemli etmen mimarisi çalışmalarında Anlamsal Web ortamı ve ÇES'lerin bu ortam üzerinde çalışabilmesi için ihtiyaç duyulan yapıların desteklenmediği görülmektedir. Önerilen referans mimarisinin çalışmalarda gözlemlenen bu eksikliği kapamaya yönelik bir ilk adımı temsil ettiği söylenebilir.



## 4 Sonuç ve İleriye Yönelik Çalışmalar

Bu bildiride anlatılan referans mimarisi Anlamsal Web ortamında çalışacak etmen sistemlerinin tasarımı ve uygulaması için gerekli yapıları tanımlamayı hedeflemektedir. Mimarinin öncül bir versiyonu [13]'teki çalışmada da yer almaktadır. Ancak bu öncül versiyonda katmanlar arası ilişkiler belirgin değildir ve gerek katmanların gerekse de katmanlar içerisindeki modüllerin gösteriminde ihtiyaç duyulan biçimsellik eksiktir. Bu bildiride anlatılan çalışma ile mimarinin söz konusu ihtiyaçlar doğrultusunda yenilenmesi yerine getirilmiştir.

Önerilen referans mimari Bölüm 2'de de belirtildiği gibi, yazılım mimarilerini belgelemek için [14]'te yer alan bakış tiplerinden *Modül Bakış Tipi* ("Module Viewtype") göz önüne alınarak hazırlanmıştır ve gereksinimleri karşılayacağına inanılan ilgili modül bakış tipi stili (*katmanlı stil*) ile bu çalışmada anlatılmıştır. Ancak referans mimarisinin, mimariyi farklı perspektiflerden tanımlayan diğer mimari bakış tipleri ile de desteklenmesi gerekmektedir. Bu amaçla yakın gelecekte, Anlamsal Web yetenekli ÇES'lerde yer alan etmenlerin çalışma zamanı davranışlarını ifade eden stilleri içerdiği düşünülen *Bileşen ve Bağlaç Bakış Tipi* ("Component-and-Connector Viewtype") [14] göz önüne alınarak mimari başka bir perspektiften değerlendirilecek ve belgelendirilecektir. Bir sonraki adım da ise yine bu tip ÇES'lere ait bileşenlerin üstlendikleri görevler doğrultusunda sistem içerisindeki konumlarını belirlemek amacıyla *Yerleşim Bakış Tipi* ("Allocation Viewtype") [14] stillerinin incelenmesi ve uygun olan veya olanlarının kullanılarak mimarinin desteklenmesi hedeflenmektedir.

Öte yandan tanıtılan referans mimarisi modüllerinin ve modüller arası ilişkilerin bu modüller kullanılarak geliştirilen somut ÇES yazılım mimarisi çalışmalarından elde edilen deneyimlere bağlı olarak daha güncel ve etkin hale gelebileceklerine inanılmaktadır. Bu amaç doğrultusunda SEAGENT<sup>1</sup> ÇES yazılımı geliştirme çerçevesi üzerinde devam eden çalışmalarımızın yeni mimari ihtiyaçlarını belirlemede ve varolan mimariyi güncellemede fayda sağlayacağı düşünülmektedir.

## Kaynakça

1. Weiss, G., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, USA, (1999).
2. Sycara, K., Multiagent Systems, AI Magazine, 19 (4), s. 79-92, (1998).
3. Weyns, D. ve Holvoet, T., "Multiagent Systems and Software Architecture", In proceedings of the Multiagent Systems and Software Architecture (MASSA), Special Track at Net.ObjectDays 2006, Erfurt, Almanya, s. 7-30, (2006).
4. Bass, L., Clements, P. ve Kazman, R., Software Architecture in Practice, Addison Wesley Publishing Comp., (2003).

---

<sup>1</sup> SEAGENT, Anlamsal Web ortamında çalışacak yazılım etmen sistemlerinin geliştirilmesini sağlayan açık kaynak kodlu bir Java çerçevesidir. Güncel sürümüne ve ilgili dokümantasyona <http://seagent.ege.edu.tr> adresinden erişilebilir.

5. Shehory, O., Architectural Properties of MultiAgent Systems, Teknik Rapor, CMU-RI-TR-98-28, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, (1998).
6. Kolp, M., Giorgini, P. ve Mylopoulos, J., A Goal-Based Organizational Perspective on Multi-agent Architectures, Lecture Notes in Computer Science, 2333, s. 128-140, (2002).
7. Weyns, D. ve Holvoet, T., A Reference Architecture for Situated Multiagent Systems, Lecture Notes in Computer Science, 4389, s. 1-40, (2007).
8. Garcia, A., Kulesza, U. ve Lucena, C., Aspectizing Multi-Agent Systems: From Architecture to Implementation, Lecture Notes in Computer Science, 3390, s. 121-143, (2005).
9. Kendall, E. A., Malkoun, M. T. ve Jiang, C. H., Multiagent System Design Based on Object Oriented Patterns, Journal of Object Oriented Programming, 10 (3), s. 41-47, (1997).
10. Gracanic, D., Singh, H. L., Bohner, S. A. ve Hinchey, M. G., Model-Driven Architecture for Agent-Based Systems, Lecture Notes in Artificial Intelligence, 3228, s. 249-261, (2005).
11. Berners-Lee, T., Hendler, J. ve Lassila, O., The Semantic Web, Scientific American, 284 (5), s. 34-43, (2001).
12. Reed, P., Reference Architecture: The Best of Best Practices, The Rational Edge, <http://www-128.ibm.com/developerworks/rational/library/2774.html>, (2002).
13. Kardas, G., Goknil, A., Dikenelli, O. ve Topaloglu, N. Y., "Metamodeling of Semantic Web Enabled Multiagent Systems", In proceedings of the Multiagent Systems and Software Architecture (MASSA), Special Track at Net.ObjectDays 2006, Erfurt, Almanya, s. 79-86, (2006).
14. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R. ve Stafford, J., Documenting Software Architectures: Views and Beyond, Pearson Education, USA, (2003).
15. FIPA, FIPA Abstract Architecture Specification, <http://www.fipa.org/specs/fipa00001/SC00001L.pdf>, (2002)
16. Kardas, G., Gümüs, Ö. ve Dikenelli, O., Applying Semantic Capability Matching into Directory Service Structures of Multi Agent Systems, Lecture Notes in Computer Science, 3733, s. 452-461, (2005).
17. Sycara, K., Paolucci, M., Ankolekar, A. ve Srinivasan, N., Automated discovery, interaction and composition of Semantic Web Services, Journal of Web Semantics, 1, s. 27-46, (2003).
18. OWL-S Coalition, OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.1/overview/>, (2004).
19. WSMO Working Group, Web Service Modeling Ontology, <http://www.wsmo.org/index.html>, (2005).
20. Paolucci, M., Kawamura, T., Payne, T., R. ve Sycara, K., Semantic Matching of Web Services Capabilities, Lecture Notes in Computer Science, 2342, s. 333-347, (2002).
21. Gümüs, Ö., Gürçan, Ö., Kardas, G., Ekinci, E. E. ve Dikenelli, O., Engineering an MAS Platform for Semantic Service Integration based on the SWSA, Lecture Notes in Computer Science, 4805, s. 85-94, (2007)
22. Williamson, M., Decker, K. ve Sycara, K., "Unified Information and Control Flow in Hierarchical Task Networks", In Proceedings of the AAAI-96 Workshop, California, USA, s. 142-150, (1996).

23. Dickinson, I. ve Wooldridge, M., “Agents are not (just) web services: considering BDI agents and web services”, In proceedings of the workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE 2005), Utrecht, the Netherlands, (2005).
24. Dikenelli, O., Erdur, R. C., Kardas, G., Gümüs, O., Seylan, I., Gurcan, O., Tiryaki, A. M. ve Ekinçi, E. E., Developing Multi Agent Systems on Semantic Web Environment using SEAGENT Platform, Lecture Notes in Artificial Intelligence, 3963, s. 1-13, (2006).
25. Brussel, H. V., Wyns, J., Valckenaers, P., Bongaerts, L. ve Peeters, P., Reference Architecture for Holonic Manufacturing Systems: PROSA, Journal of Manufacturing Systems, 37, s. 255–274, (1998).
26. Amor, M., Fuentes, L. ve Vallecillo, A., Bridging the Gap Between Agent-Oriented Design and Implementation Using MDA, Lecture Notes in Computer Science, 3382, s. 93-108, (2005).
27. Hahn, C., Madrigal-Mora, C., Fischer, K., Elvesæter, B., Berre, AJ. ve Zinnikus, I., Meta-models, Models, and Model Transformations: Towards Interoperable Agents, Lecture Notes in Artificial Intelligence, 4196, s. 123–134, (2006).