

FIPA Uyumlu Yazılım Etmeni Çerçevesi Gerçekleştirimi

Rıza Cenk Erdur¹, Oğuz Dikenelli²

^{1,2} Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, 35100, Bornova, İzmir

¹ erdur@staff.ege.edu.tr, ² oguzd@staff.ege.edu.tr

Özet. Yazılım etmeni çerçeveleri; bir etmenin iletişim, planlama-işletim, kullanıcı arayüzleri gibi temel modüllerini hazır olarak sunmakta ve etmen sistemi geliştiricilere yalnızca uygulama alanına özgü planların tanımlanarak bir plan editörü yardımı ile etmene tanıtılması sorumluluğunu bırakmaktadır. Böylece sistemdeki her yazılım etmeninin tüm modüllerini baştan gerçekleştirmek gerekmediği için daha güvenilir ve hızlı bir biçimde etmen sistemleri geliştirme olanağı doğmaktadır. Bu çalışmada, esnek bir mimariye sahip, farklı uygulama alanlarında etmen sistemi geliştirmede kullanılabilir ve diğer yazılım etmeni sistemleri ile birlikte-işlerliği sağlamak için FIPA standartlarına uyumlu olarak geliştirilen etmen çerçevesinin¹ yazılım mimarisi ve gerçekleştirim ayrıntıları anlatılmaktadır.

1 Giriş

Yazılım etmeni çerçeveleri (software agent frameworks), bir yazılım etmeninin geliştirilebilmesi için gerekli olan iletişim, planlama, çizelgeleme, eylem işletimi ve izleme, plan editörü, etmen-kullanıcı arayüzü gibi temel modülleri sunmaktadır. Böylece, etmen sistemi geliştiricileri, sistemdeki her bir yazılım etmenini baştan gerçekleştirmek yerine, kendilerine sunulan etmen çerçevesini kullanarak daha hızlı ve kolay etmen geliştirme olanağına sahip olmaktadır.

Bir yazılım etmeni çerçevesinin geniş bir kullanım alanı bulması için, esnek bir mimariye sahip olması ve belirli standartlara uyumlu olması gerekmektedir.

Esnek bir mimariye sahip olmak, yazılım etmeni çerçevesinin temel modüllerinin gerek etmen çerçevesi üreticisi gerekse etmen sistemi geliştiriciler tarafından kolaylıkla değiştirilmesi ve geliştirilmesi olanağını sağlamaktadır. Esnek bir mimariye sahip olmak etmen çerçevesinin farklı uygulama alanlarında kullanılabilirliğini de artıracaktır. Örneğin, anlamsal web bilgi arama amacı ile kullanılacak bir yazılım etmeninin planlama ve işletim modülü, standart bilgi arama uygulamalarında kullanılan etmenlere göre daha farklı olmalıdır. Bu durumda planlama ve işletim modülünün ilgili planlama yaklaşımını destekleyecek biçimde ve kolaylıkla değiştirilebilmesi etmen çerçevesinin anlamsal web ortamında çalışan yazılım etmenleri gerçekleştiriminde kullanılabilmesini sağlayacaktır. Diğer taraftan, farklı iletişim protokolleri veya standartları ortaya çıktığında, etmen çerçevesinin iletişim modülünün yeni protokol ve standartları da destekleyecek biçimde geliştirilebilmesi gerekmektedir. Bu açıdan bakıldığında, yazılım etmeni çerçevesi tasarımında esnek bir mimarinin önemi ortaya çıkmaktadır.

Diğer önemli bir konu, sayıları gün geçtikçe artan etmen sistemi platformlarının İnternet üzerindeki diğer etmen sistemi platformları ile birlikte-işlerliğinin (interoperability) sağlanmasıdır. FIPA (Foundation for Intelligent Physical Agents), etmen sistemleri geliştiriminin standartlarını tanımlamaya yönelik olarak çeşitli spesifikasyonlar yayınlayan, gerek akademi gerekse endüstriden birçok üyesi olan, kar amacı gütmeyen bir organizasyondur. Bir yazılım etmeninin karakteristiklerini tanımlayan soyut etmen mimarisi [3], yazılım etmenlerinden oluşan etmen platformunun yapısı ve yönetimi [4] ve etmenler arası iletişim ile ilgili spesifikasyonlar [5], [6] FIPA'nın ortaya koyduğu temel standartları tanımlamaktadır. Bu standartları ortaya koymadaki temel amaç, farklı etmen sistemleri arasındaki birlikte-işlerliği sağlamaktır. Günümüzde en çok kullanılan etmen çerçevelerinin büyük çoğunluğu [1] [8] [11] geniş bir kitle tarafından kabul görmüş olan FIPA standartlarını desteklemektedir. FIPA standartlarına uyumlu olma, yazılım etmeni çerçevesi üreticileri açısından dikkate değer ek bir çaba gerektirmekle birlikte, bu standartlara uyum çok önemlidir. Aksi durumda, farklı kişi veya kurumlar tarafından geliştirilen etmen sistemlerinin birbiriyle konuşması güçleşecek ve ancak özel köprü yazılımlar kullanarak sağlanabilecektir.

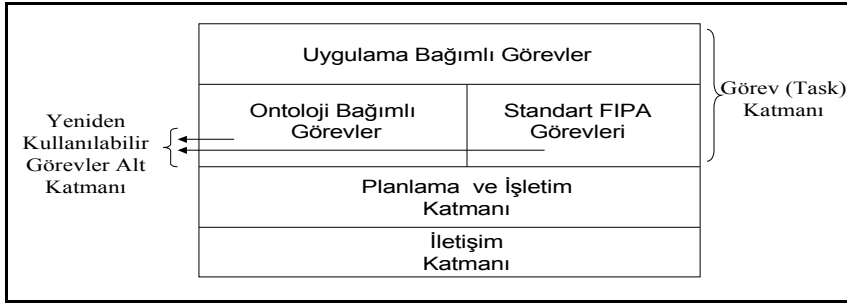
Bildirinin izleyen bölümleri şu şekilde düzenlenmiştir: 2. bölümde, geliştirilen etmen çerçevesinin katmanlı mimarisi, içsel yapısı ve işleyişi anlatılmaktadır. Katmanlı mimarinin ilk katmanı olan iletişim katmanına ilişkin gerçekleştirim ayrıntıları 3. bölümde, ikinci katman olan planlama ve işletim katmanı 4. bölümde ve son katman olan görev katmanı 5. bölümde anlatılmaktadır. Geliştirilen yazılım etmeni çerçevesinin literatürdeki yeri ve önemli özellikleri 6. bölümde tartışılmaktadır.

2 Geliştirilen Etmen Çerçevesinin Mimarisi ve İşleyiş

Geliştirilen etmen çerçevesi katmanlı bir mimariye sahiptir. Şekil 1'de görülen katmanlı mimarinin en önemli getirisi, katmanların birbirinden bağımsız olarak geliştirilebilmesini sağlamaktır.

¹ Bu çalışmada tanıtılan yazılım etmeni çerçevesi, 102E022 (GSRT) numaralı TÜBİTAK projesi olarak yürütülen DAMES-T (Development of an Agent Mediated E-Commerce System for Tourism Domain) kapsamında gerçekleştirilmektedir.

FIPA öncelikli olarak birlikte-işlerliği hedeflediği için, FIPA spesifikasyonlarının önemli bir bölümü iletişim modülü gerçekleştirimi ile ilgili konuları adreslemektedir. Bu doğrultuda, gerçekleştirilen iletişim katmanı bir etmenin gerek içinde bulunduğu etmen platformundaki gerekse diğer FIPA etmen platformlardaki etmenler ile FIPA standartlarında belirtilen protokoller ile ve FIPA-ACL (FIPA Agent Communication Language) kullanarak iletişimde bulunabilmesini sağlayan sınıflardan oluşmaktadır. Planlama ve işletim katmanı, belli bir planlama stratejisi kullanarak plan oluşturma, bu planların içindeki görevleri çizelgeleme, işletme ve görevlerin işleyişi izlemeden sorumludur. Geliştirilen etmen çerçevesinde sıradüzensel görev ağı (hierarchical task network - HTN) [10] yaklaşımı temel alınmış ancak etmen çerçevesinin anlamsal



Şekil 1. Geliştirilen etmen çerçevesinin katmanlı mimarisi.

web'te kullanılacağı göz önüne alınarak, planlama ve işletim modülünün ileride kolaylıkla ekleme yapılabilecek esnek bir yapıda tasarımı üzerinde durulmuştur. Görev (task) katmanı, gerek hazır olarak çerçeve ile birlikte sunulan gerekse daha sonra plan editörü yardımı ile girilecek olan planları tutmaktadır. Yeniden kullanılabilir görevler ve uygulama bağımlı görevler olmak üzere iki alt katmandan oluşmaktadır. Yeniden kullanılabilir görevler alt katmanı, alandan bağımsız diğer bir deyişle tüm etmenlerde ortak olarak kullanılan görevleri içermektedir. Şekil 1'den görüldüğü gibi, yeniden kullanılabilir görevler alt katmanı da kendi içinde iki görev grubundan oluşmaktadır. Ontoloji bağımlı görevler, etmenin ontoloji dilleri ile oluşturulmuş bilgilerin bulunduğu anlamsal web ortamında kullanılabilmesini sağlayacaktır. Standart FIPA görevleri, bir FIPA etmen platformunda yaşayan bir etmenin yerine getirmesi gereken temel görevlerdir. Örneğin, Etmen Yönetim Servisi (Agent Management Service-AMS) ve Dizin Servisi (Directory Facilitator-DF) [4] ile ilgili kayıt olma, sorgulama, kayıt silme gibi işlemler standart FIPA görevleri olarak nitelendirilmiştir. Görev katmanının diğer alt katmanını oluşturan uygulama bağımlı görevler ise, etmen sistemi uygulamasının geliştirilmekte olduğu alana özgü görevlerdir.

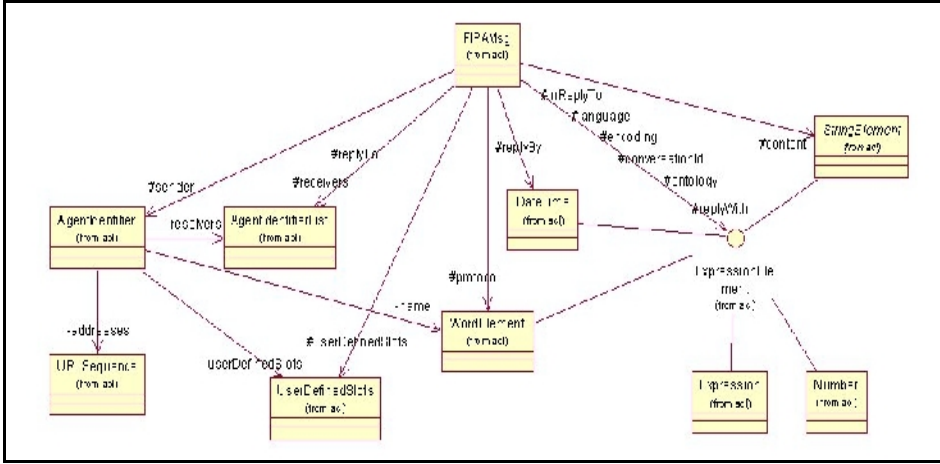
2.1 İşleyiş

Söz konusu etmen çerçevesinden türetilen bir etmenin işleyişinin daha iyi anlaşılabilmesi için, bir X etmeninin aynı platformda bulunan ve turizm alanında otel bilgileri tutan bir Y etmeninden servis alması için X ve Y etmenleri arasında geçen etkileşim ile ilgili senaryo aşağıda anlatılmaktadır:

X etmeni, rezervasyon yaptırmak istediği otel ile ilgili özellikleri belli bir içerik dilinde ifade edilen sorgu biçiminde hazırlayıp, FIPA-ACL iletilisinin içine yerleştirmektedir. Bundan sonra, oluşturulan FIPA-ACL iletilisi belli bir kodlama yaklaşımına göre kodlanmakta ve kullanılacak aktarım (transport) protokolüne bağlı olarak bir zarf oluşturulmaktadır. Bu işlemler iletişim katmanının konu edinildiği 3. bölümde daha ayrıntılı olarak anlatılmaktadır. Kodlanmış ileti ve zarf, etmen iletişim kanalı (EİK) adı verilen bir servise gönderilmekte ve bu servis de iletiyi Y etmenine iletmektedir. Y etmeni, iletişim kuyruğuna bir ileti geldiğinde iletiyi ayrıştırıp performatif ve içerik kısımlarını yorumlayıp kendisinden ne istendiğini anlamaktadır. Bu örnekte, Y etmeni kendisinden belli özelliklere sahip otelden rezervasyon yaptırılmak istendiğini anlayacaktır. Y etmeni daha sonra o anki durumuna göre kendisinden istenenleri yapıp yapamayacağını belirlemekte, eğer olumlu ise kendisinden isteneni amaç (objective) kuyruğuna yerleştirmektedir. Planlama modülü, sürekli olarak amaç kuyruğunu kontrol etmekte ve kuyruğa gelen her amaca göre bir plan yapısı oluşturmaktadır. Bu örnekte, rezervasyon yapma ile ilgili plan yapısı oluşturulacaktır. Plan yapıları, çeşitli bileşik görevlerin (complex task) [10] açılımı ile oluşan bir ağaç yapısına benzemektedir. Bu ağaç yapısındaki yaprak düğümler işletilebilir eylemlerdir. İşletilebilir eylemlerin gerçekleştirilebilmesi için gerekli şartlar sağlanınca, bu eylemler hazır kuyruğuna konmaktadır. Çizelgeleyici, hazır kuyruğundan belli bir algoritmaya göre seçtiği eylemleri eylem işleticiye iletmekte ve ağaç yapısındaki işletilebilir her görev bitene kadar bu zincir devam etmektedir. Son olarak, rezervasyon yapma planına bağlı olarak, X etmenine rezervasyona ilişkin bilgiler geri döndürülebilecektir.

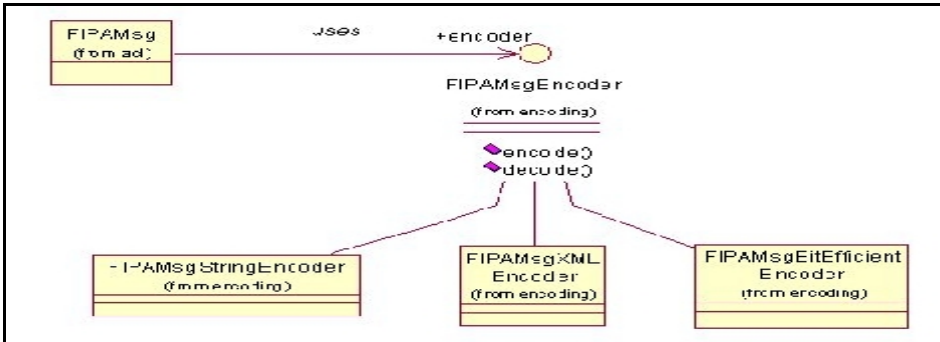
3 İletişim Katmanı

FIPA etmen sistemlerinde, etmenler FIPA-ACL (FIPA Agent Communication Language) dilini kullanarak iletişimde bulunmaktadır. FIPA-ACL dili konuşma-edin (speech-act) kuramına dayanmakta ve performatif, gönderici, alıcı, ontoloji, içerik dili, içerik gibi sahalardan oluşmaktadır [6]. FIPA-ACL iletilerinin ağ üzerinde farklı yerlerde bulunan etmenler arasında gönderilip alınabilmesi için, FIPA-ACL iletilerinin ilk önce belli bir kodlama (encoding) [7] stratejisine göre (karakter katarı, bit gösterim, XML, vb.) kodlanması gerekmektedir. Bu kodlamalar için gerekli olan EBNF tanımları ilgili FIPA spesifikasyonlarında belirtilmiştir. Şekil 2’ de, ilgili FIPA spesifikasyonundaki [7] EBNF tanımına uygun karakter katarı (string) kodlama için kullanılan sınıflar görülmektedir.



Şekil 2. FIPA ACL iletilerinin karakter katarı (string) olarak kodlanması.

Farklı kodlama stratejisi desteği ortak bir arayüzden türetilmiş sınıflarda sağlanmaktadır. Herhangi bir kodlama stratejisi için ilgili kodlayıcı sınıflar şekil 3’te görülen hiyerarşik yapıdaki gibi gerçekleştirilmelidir.



Şekil 3. FIPA-ACL iletilerinin farklı stratejilere göre kodlanması.

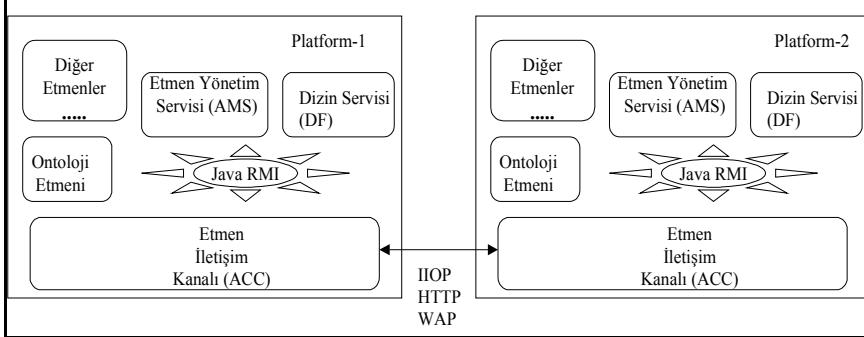
Kodlanmış FIPA-ACL iletileri “payload” olarak adlandırılmaktadır. “Payload” kısmına aktarım katmanına özgü gönderici, alıcı adresleri ve bazı diğer parametreler içeren bir zarf [5] eklenerek “Aktarım İletisi (Transport Message)” oluşturulur. Etmenler arasında alışverişi yapılan gerçek iletiler bu aktarım iletileridir. Diğer bir deyişle, aktarım iletileri adreslenmiş “payload” içerisindedir.

FIPA uyumlu etmen sistemleri, çeşitli servislerin bulunduğu ve “platform” olarak adlandırılan ortamlar olarak gerçekleştirilmektedir [4]. FIPA etmen platformlarında, platform içi ve platform dışı olmak üzere iki çeşit iletişim gerçekleştirilmektedir. FIPA etmen platformlarında iletişim şekli 4’te görülmektedir.

Etmen yönetim servisi (Agent Management Service-AMS), platform üzerindeki etmenlerin bilgilerini tutmakta ve

yaşam döngüsünü kontrol etmektedir. Dizin servisi (Directory Facilitator-DF), etmenlerin sundukları servisleri tutmaktadır. Ontoloji etmeni, platformdaki ontolojileri tutmakta, güncellemeleri kontrol etmekte ve ontolojiler arası çevrim gibi servisler vermektedir.

FIPA standartlarına göre etmenlerin birbirleri ile iletişiminin etmen iletişim kanalı -EİK (Agent Communication Channel-ACC) adı verilen servis üzerinden gerçekleştirilmesi önerilmektedir. FIPA, platform içi iletişimde kullanılacak



protokolün seçimini

Şekil 4. FIPA etmen platformu.

geliştiricilere bırakmıştır. Geliştirilen etmen çerçevesinde, platform içi iletişim için saf Java yaklaşımı ile dağıtık programlamayı destekleyen Java RMI kullanılmaktadır. Ancak, platformlar arası iletişimde kullanılabilir protokoller FIPA tarafından belirlenmiştir. Geliştirilen etmen çerçevesi, platformlar arası iletişimde IOP (Internet Inter ORB Protocol) protokolünü Java RMI over IOP teknolojisi ile desteklemektedir. Diğer bir platformdaki bir etmene mesaj göndermek isteyen bir etmen, ilk olarak kendi platformundaki EİK servisine Java RMI ile mesajı bırakır. EİK daha sonra mesajın diğer platformdaki bir etmene ait olduğunu anlayarak, IOP protokolü kullanarak mesajı diğer platformun EİK servisine iletmektedir. Son olarak, diğer platformun EİK servisi mesajı kendi platformundaki ilgili etmene iletmektedir.

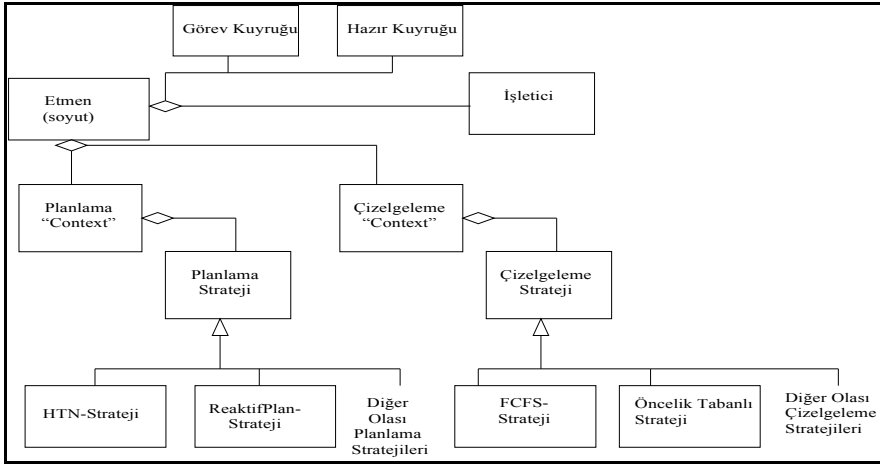
RMI ve IOP protokol desteği EİK içinde hazır olarak gelmektedir. Ancak, WAP, SOAP gibi diğer protokol desteklerinin daha sonra eklenebilmesi için EİK içinde dinamik bir kayıt servisi oluşturulmuştur. Her yeni protokole ilişkin sunucu kendisini önceden tanımlanmış arayüzü kullanarak EİK servisine tanıtmaktadır.

4 Planlama ve İşletim Katmanı

Planlama ve işletim katmanı, farklı planlama ve çizelgeleme stratejilerini destekleyecek biçimde esnek bir yapıda tasarlanmıştır. Bu amaçla, strateji tasarım deseninden [9] yararlanılmıştır. Strateji deseni, birkaç algoritma arasından dinamik olarak seçim yapmayı kolaylaştırmaktadır. Stratejiler arasında geçişleri "context" nesnesi olarak adlandırılan sınıf sağlamaktadır. Strateji deseninin planlama ve işletim katmanında kullanımı şekil 5'de görülmektedir.

Geliştirilen etmen çerçevesinin şu andaki sürümünde sıradüzensel görev ağı (HTN) planlama stratejisi kullanılmaktadır. Esnek yapı sayesinde yeni bir planlama stratejisi eklenebilecektir. Örneğin, anlamsal web ortamı için yeni bir planlama stratejisi tanımlanıp kolaylıkla eklenebilecektir. Benzer şekilde, şu anki sürümdeki çizelgeleme modülü, ilk gelen ilk hizmet alır (FCFS) mantığına dayanmaktadır. Ancak, öncelik temelli veya başka bir çizelgeleme stratejisi de kolaylıkla eklenebilecektir.

Planlama, çizelgeleme ve işletim modülleri ayrı birer iş-parçası (thread) olarak çalışmaktadır ve bu iş parçaları görev kuyruğu, hazır kuyruğu gibi veri yapıları ile senkronize edilmektedir. Örneğin, çizelgeleyici sürekli olarak görev kuyruğunu takip etmekte ve ön şartları sağlanmış olan görevleri hazır kuyruğuna yerleştirmektedir. İşleyici de, hazır kuyruğundan aldığı görevlere ilişkin eylemleri (kod parçalarını) çalıştırmaktadır.



Şekil 5. Planlama katmanının esnek tasarımı.

5 Görev Katmanı

Katmanlı mimarinin tanımlandığı şekil 1’den de görüldüğü gibi görev katmanı, iki alt katmandan oluşmaktadır. Yeniden kullanılabilir görevler alt katmanı, her etmende ortak olarak kullanılacak görevleri içermektedir. Bu katman, kendi içinde ontoloji bağımlı görevler ve standart FIPA görevlerini içeren iki kısma ayrılmıştır.

Geliştirilen etmen çatısının en önemli özelliklerinden birisi anlamsal web ortamında kullanılabilmesidir. Anlamsal web üzerinde yer alan bilgiler ontolojiler kullanılarak modellendiği ve ontoloji dilleri ile gösterildiği için, ontoloji bağımlı görevlere gereksinim duyulmaktadır. Ontoloji bağımlı sorgu arayüzlerinin ve ontolojiler ile yerel bilgi kaynaklarının bilgi modelleri arasında eşleme(mapping) yapmayı sağlayacak arayüzlerin oluşturulması ontoloji bağımlı görevlere ilişkin örneklerdir [2].

Standart FIPA görevleri, FIPA uyumlu her etmenin desteklemesi gereken görev yapılarına karşılık gelmektedir. Örneğin, yeni oluşturulan bir etmenin kendisini etmen yönetim servisine tanıtmaya veya yeteneklerini platformun izin servisine tanıtmaya standart FIPA görev yapılarına ilişkin örneklerdir. Ayrıca, FIPA tarafından tanımlanan FIPA İngiliz, Hollanda pazarlık modelleri, anlaşma ağı (contract net) gibi etkileşim protokollerinin etmen tarafından desteklenmesini sağlayan görev yapıları da bu kategori altında toplanmıştır.

Yeniden kullanılabilir görevler etmen çerçevesi ile hazır olarak gelmekte iken, uygulama bağımlı görevler etmen geliştiricileri tarafından tanımlanıp, plan editörü yardımı ile etmene girilmektedir. Örneğin, etmen tabanlı yazılım bileşeni pazaryeri sisteminden elde edilen bir yazılım bileşenin kullanılabilirliğinin değerlendirilmesi ile ilgili görev yapıları, yazılım bileşeni pazaryeri uygulamalarına özgü olup, bu alanda etmen sistemi geliştiren kişiler tarafından tanımlanmalıdır.

6 İlgili Çalışmalar ve Değerlendirme

Günümüzde, bazıları ticari amaçlı bazıları ise açık kod mantığına göre geliştirilen çeşitli etmen çerçeveleri bulunmaktadır. Son zamanlarda geliştirilen çerçevelerin çoğunluğu FIPA uyumludur. JADE [1], ZEUS[11] ve FIPA-OS [8] serbest olarak kullanılabilen FIPA uyumlu çerçeveler arasında en önemlilerdendir. DECAF [10], “Belief Desire Intention - BDI” içsel mimarisi temel alınarak geliştirilen ilk etmen çerçevelerinden biri olması ve diğer araştırmacılara örnek olması açısından önemlidir. FIPA desteği olmamasına rağmen, FIPA destekli bazı çerçeveler ile iletişimde bulunabilmesi için köprü yazılımlar geliştirilmiştir.

Geliştirilen etmen çerçevesinin literatürdeki diğer çerçevelere göre en önemli üstünlükleri şunlardır:

*Platformdaki etmen iletişim kanalında RMI dışında farklı aktarım protokollerini (HTTP, WAP, SOAP, VB.) dinamik olarak destekleyebilme özelliği.

*Anlamsal web ile ilgili görevlerin, standart FIPA görevleri ile birlikte gruplanarak, her etmende ortak olarak kullanılan görevleri içeren “yeniden kullanılabilir görevler” alt katmanının oluşturulması düşüncesi geliştirilen etmen çerçevesine özgü bir düşüncedir.

Teşekkür

DAMES-T projesinde öğrenci arařtırmacı olarak alıřan, İnan SEYLAN ve Önder GÜRCAN, geliřtirilen yazılım etmeni çerçevesinin iletiřim modülünün kodlama ařamasında önemli katkılarda bulunmuřlardır.

Kaynaka

1. Bellifemine, F., Poggi, A., Rimassa, G.: Developing Multi-agent Systems with a FIPA-compliant Agent Framework. *Software Practice and Experience*, 31, 2001, s. 103-128.
2. Erdur, R.C. ve Dikenelli, O., “A Standards Based Agent Framework for Instantiating Adaptive Agents”, *Proceedings of Int. Conference Autonomous Agents and Multi-agent Systems (AAMAS 2003)*, 2003, s.984-985.
1. FIPA(a), “FIPA Abstract Architecture Specification”, <http://www.fipa.org>
2. FIPA(b), “FIPA Agent Management Specification”, <http://www.fipa.org>
3. FIPA(c), “FIPA Agent Message Transport Service Specification”, <http://www.fipa.org>
4. FIPA(d), “FIPA ACL Message Structure Specification”, <http://www.fipa.org>
5. FIPA(e), “FIPA ACL Message Representation in String Specification”, <http://www.fipa.org>
6. FIPA-OS, <http://www.emorpha.com/research>.
7. Gamma, E., vd., “Design Patterns:Elements of Reusable Object Oriented Software”, Addison Wesley, Reading (MA), 1995.
8. Graham, J.R., Decker, K.S., Mersic, M., “DECAF – A Flexible Multi Agent System Architecture”, *Autonomous Agents and Multi-Agent Systems*, 7, 7-27, 2003.
9. Nwana, H.S., Ndumu, D.T., Lee, L.C., Coll, J.C., “ZEUS: A Tool-kit for Building Distributed Multi-agent Systems”, *Applied AI Journal*, 13(1), 1999, s.129-186.