İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DAĞITIK ÇOKLU ETMEN MÜZAYEDE SİSTEMİ

YÜKSEK LİSANS TEZİ
Müh. Ali ÜLLENOĞLU

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

Tez Danışmanı: Prof. Dr. Nadia ERDOĞAN

HAZİRAN 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DAĞITIK ÇOKLU ETMEN MÜZAYEDE SİSTEMİ

YÜKSEK LİSANS TEZİ
Ali ÜLLENOĞLU
(504061502)

Tezin Enstitüye Verildiği Tarih : 30 Nisan 2009
Tezin Savunulduğu Tarih : 04 Haziran 2009

Tez Danışmanı : Prof. Dr. Nadia ERDOĞAN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Coşkun SÖNMEZ (YTÜ)
Yrd. Doç. Dr. Sanem S. TALAY (İTÜ)

HAZİRAN 2009

Aileme,

ÖNSÖZ

Tez çalışmam süresince yardımlarını ve desteğini benden esirgemeyen değerli hocam sayın Prof. Dr. Nadia Erdoğan'a çok teşekkür ederim. Ayrıca, öncelikle çalışmam boyunca bana destek olan aileme, beni motive eden tüm arkadaşlarıma ve çalışmamı daha rahat sürdürebilmem için ellerinden geldiğince yardımcı olan iş arkadaşlarıma teşekkür ederim.

Mayıs 2009

Ali Üllenoğlu

(Bilgisayar Mühendisi)

İÇİNDEKİLER

KISALTMALAR

ACL	**:** Agent Communication Language
API	**:** Application Programming Interface
AMS	**:** Agent Management System
ATM	**:** Asynchronous Transfer Mode
CD	**:** Compact Disc
DF	**:** Directory Facilitator
FIPA	**:** Framework for Intelligent Physical Agents
JADE	**:** Java Agent Development Framework
JSP	**:** Java Server Pages
LEAP	**:** Lightweight Extensible Agent Platform
MASFIT	**:** Multi Agent System for Fish Trading
MASON	**:** Multi Agent Simulator of NeighbourHoods
MTS	**:** Message Transport Service
OASIS	**:** Optimal Aircraft Sequencing Using Intelligent Scheduling
RMI	**:** Remote Method Invocation
SL	**:** Semantic Language
SQL	**:** Structured Query Language
TILAB	**:** Telecom Italia Lab
XML	**:** Extensible Markup Language

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

DAĞITIK ÇOKLU ETMEN MÜZAYEDE SİSTEMİ

ÖZET

Internet üzerinden yapılan müzayedeler, elektronik ticaret sistemlerinin en önemli parçalarından birisidir ve çok geniş bir kullanıcı kitlesine sahiptir. Bir ürün ya da hizmet satın almak isteyen müşteri, Internet tabanlı bir müzayede evine bağlanarak müzayedelere katılıp bu ürün ya da servisleri satın alabilmektedir. Çevrimiçi müzayede evlerinin çokluğu ve bu müzayede evlerinde birçok ürünün aynı anda birden fazla müzayededede satılabiliyor olması, kullanıcıların alışveriş yapmasını zorlaştırabilmekte ve daha ucuza alabilecekleri ürünü pahalıya alabilmelerine neden olmaktadır. Bu nedenle, kullanıcının adına çevrimiçi müzayede evlerine bağlanıp; bu müzayede evlerinde alınmak istenen ürünün satıldığı müzayedelere kullanıcının adına katılacak ve müzayedeleri gerçek zamanlı bir şekilde takip edecek sistemlere ihtiyaç doğmuştur.

Bu çalışmada bir dağıtık çoklu etmen müzayede sistemi yapısı önerilmekte ve bu sistemin uygulama ayrıntıları anlatılmaktadır. Etmenler, bağımsız hareket etme ve mantıklı karar verebilme yeteneklerine sahip olan yazılımlardır ve bu özellikleri göz önünde bulundurulduğunda, kullanıcı adına müzayedeye katılıp teklif verme problemi için uygun bir çözüm olarak değerlendirilebilirler. Etmenler, gerçekleştirmeyi istedikleri hedefler doğrultusunda çevrelerindeki değişiklikleri de göz önünde bulundurarak ve bu etkilere göre davranışlarını değiştirerek çalışabilirler. Bir çoklu etmen ortamında birbirleri ile haberleşerek, pazarlık veya işbirliği yaparak çalışabilirler.

Sistemde her rol için bir etmen tipi belirlenmiştir. Müşterilerin müzayedelerde satıcı olarak bulundukları rolü bir satıcı etmen, alıcı olarak bulundukları rolü ise bir alıcı etmen temsil eder. Müzayede evi yöneticisi rolünü müzayede evi etmeni, tek bir müzayedenin yöneticisi rolünü de müzayede etmeni temsil eder. Müzayedelere katılan alıcı etmenlerin takip edecekleri farklı stratejiler belirlenerek, kullanıcının istekleri doğrultusunda etmenlerin daha esnek çalışabilmeleri sağlanmıştır.

Sistemin gerçeklenmesi sırasında etmenlerin yaratılmasını, yönetilmesini ve birbirleri ile haberleşmesini sağlayan araçlar ve yardımcı kütüphaneler sunan JADE ortamı ve kütüphanesi kullanılmıştır. JADE, etmen ve çoklu etmen sistemlerinin standartlarını belirleyen FIPA tarafında oluşturulan standartları destekleyen bir çoklu etmen geliştirme ortamı ve kütüphanesidir. Etmenler arası haberleşme ve etmenlerin davranışlarını kontrol etme için esnek ve genişletilebilir bir yapı sunmaktadır.

A DISTRIBUTED MULTI AGENT AUCTIONING SYSTEM

SUMMARY

Internet based auctions are one of the most important parts of electronic commerce systems and they have a wide range of users. A customer who wants to buy a service or product can connect to a Internet based auction house, participate in the auctions and buy the auctioned services and products. There are many online auction houses and a product can be sold from more than one auction house with more than one auction. This situation complicates the buying and selling process and may cause the user buy a product or service with a high price which he could buy it with a lower price. Therefore, a need has born for a system which connects to auction houses, participates in the auctions that are arranged in them and follows these auctions in real time.

In this work, a multi agent auctioning system structre is proposed and its implementation details are described. Agent are software programs which are able to behave autonomously and make logical decisions. If these properties are taken into account, an agent system is a suitable solution for participating in an auction and bidding in the name of a user. Agents can behave towards their goals taking into account the changes in their environments and they can change their behaviours according to these changes. They can work together or negotiate by communicating in a multi agent environment.

In the system, an agent type is defined for every role. A seller agent is defined for the role of customers who are selling a product and a buyer agent is defined for he role of customers who are buying a product. Auction house agent is defined for auction house role and auction agent is defined for auction manager role. Flexibility of the agents' behaviours is achieved by defining strategies and settings which are selected by the user.

JADE environment and library is used for creating, managing and supporting the communication of the agents in the implementation of the system. JADE environment and library supports these operations by providing various tools and libraries. JADE is a multi agent system development enviroment and library which supports the standards defined by FIPA, which is the organization that defines the standards for multi agent systems. It provides a flexible and extensible structure for agent communication and controlling behaviours of agents.

1. GİRİŞ

Bu bölümde, çoklu etmen metodolojisi kullanılarak gerçeklenen dağıtık çoklu etmen tabanlı müzayede sisteminin tanımı ve ihtiyaçları anlatılacaktır.

1.1 Problemin Tanımı

Son yıllarda Internet üzerinde yapılan müzayedeler, elektronik ticaret sistemlerinin önemli parçalarından birisi haline gelmiş ve çok geniş bir kullanıcı kitlesine sahip olmuştur. Kullanıcılar Internet'te istedikleri ürünü, çok sayıda bulunan çevrimiçi müzayede evlerine bağlanıp müzayedelere katılarak satın alabilmektedirler. Çevrimiçi müzayede evlerinin çokluğu ve kullanıcının maddi olanak ve ihtiyaçlarının çeşitliliği, kullanıcıların bu müzayede evlerinden alışveriş yapmasını zorlaştırmaktadır. Kullanıcı bir ürün satın almak istediğinde birçok müzayede evine bağlanıp müzayedenin durumunu takip etmeli; kendisinin maddi olanakları, müzayedenin koşulları ve müzayedenin teklif tarihçesine göre bir teklif vermelidir. Bu işlemleri bir kullanıcının tek başına yapmasının zorluğu, müzayedeye kullanıcı adına katılacak ve kullanıcının tercih ve seçeneklerine göre otomatik teklif verecek yazılımlara ihtiyaç duyulmasına neden olmuştur. Etmen tabanlı müzayede sistemleri, bu ihtiyaca tam olarak cevap verebilecek bir sistemdir.

Kullanıcı adına müzayedeye katılacak etmen sisteminin sahip olması gereken en önemli özellik, bir etmenin müzayede evlerini dolaşarak kullanıcının ihtiyacı olan ürünün müzayede bilgilerini alması, kullanıcının tercihlerini göz önünde bulundurarak yeni teklif oluşturması ve bu teklifleri kullanıcının onayına ihtiyaç duymadan müzayede evlerine iletebilmesidir. Verilecek teklifin belirlenmesi için kullanıcıya çeşitli stratejiler sunulmalı; kullanıcı adına teklif veren etmen, seçilen strateji seçeneklerine göre daha istekli veya isteksiz bir strateji izlemelidir. Sistemin sahip olması gereken diğer özelliği, bir etmenin sistemde kullanıcı adına yeni bir müzayede yaratabilmesi ve bu müzayedeyi takip edebilmesidir.

1.2 Çözüm Önerisi

Tez çalışmasında öne sürülen modelde, üç adet uygulama bulunmaktadır: veritabanı uygulaması, müzayede evi uygulaması ve müşteri uygulaması.

Müşteri uygulamasında her kullanıcıya karşılık bir satıcı ve birçok alıcı etmen düşmektedir. Satıcı etmen, kullanıcı adına müzayede yaratmakla ve yarattığı müzayedelerin durumlarını kontrol etmekle yükümlüdür. Alıcı etmen ise, kullanıcının adını girdiği bir ürünü sisteme kayıtlı müzayede evlerinde aramakla, kullanıcının seçenekleri ve tercihleri doğrultusunda müzayede etmenlerine teklif vermekle ve müzayedenin durumunu izlemekle yükümlüdür. Kullanıcının satın almak istediği her ürün için bir alıcı etmen yaratılmaktadır. Ayrıca kullanıcının sisteme üye olmasını ve kendisini sisteme tanıtmasını amaçlayan bir yetkilendirme etmeni tasarlanmıştır.

Müzayede evi uygulamasında, müzayede evini temsil eden ve ürün aramalarına cevap veren bir müzayede evi etmeni bulunmaktadır. Müzayede evi uygulaması, yeni bir müzayede açıldığında bir müzayede etmeni yaratır. Müzayede evine bağlı birden fazla müzayede etmeni bulunabilir. Müzayede tamamlandığında bir adet ödeme etmeni yaratılır, bu ödeme etmeni alıcı ve satıcı kullanıcıların girdiği ödeme bilgileri doğrultusunda ödeme işlemini gerçekleştirir. Ödeme işlemleri çalışmanın kapsamı dışında olduğu için, ödeme etmeni her zaman ödemenin başarılı olduğunu bildiren bir sonuç döndürür.

Veritabanı uygulamasında bir adet veritabanı etmeni bulunur. Veritabanında sisteme kayıtlı ürünlerin, müzayedelerin ve etmenlerin durum bilgileri tutulur. Bu bilgilerin kaydedilmesi, güncellenmesi, silinmesi ve okunması işlemleri veritabanı etmeni aracılığıyla yapılır. Tüm veritabanı erişimleri müzayede evi ve müzayede etmenleri tarafından yapılır. Böylece müşterilerin veritabanına erişerek müzayede bilgilerini okuması veya değiştirmesi engellenmiştir.

Sistemin kötü amaçla kullanılmaması için gerekli olan mesaj şifreleme gibi güvenlik önlemleri çalışmanın kapsamının dışında bırakılmıştır. Çalışmanın hedefi kullanıcıya mümkün olduğunca fazla seçenek sunarak, kullanıcının ürün alma isteğini müzayede tekliflerine doğru olarak yansıtabilmektir.

2. ETMEN

2.1 Etmen Tanımı

Etmenler hakkında evrensel olarak kabul edilmiş bir tanım yoktur. Genel olarak kabul edilen tek etmen özelliği kendi iradesi ile hareket edebilme yeteneğidir. Etmenlerin bunun dışındaki özellikleri kimi uygulamalarda kullanıldığı, kimi uygulamalarda kullanılmadığı için etmen tanımı içerisinde değerlendirilmeleri de tartışmalıdır. Örneğin bir etmenin öğrenme ve deneyim kazanma özelliği bir uygulama için gerekli olabilirken, başka bir uygulama için tamamen gereksiz olabilir [1]. Bu bilgiler bağlamında etmen tanımı, tasarım hedefleri bağlamında bir ortam içerisinde bağımsız hareket edebilen bir bilgisayar sistemi olarak yapılabilir [2]. Bir etmen sistemi oluşturmak için, ortamda çalışan ve ortamla iletişim halinde olan tek bir etmen yeterli olsa da, genellikle birçok etmen aynı ortamda birlikte çalışır. Birden fazla etmenin; aynı ortamda, birbirleriyle çelişen veya uyum halinde olan hedefleri bağlamında çalıştıkları ve birbirleriyle pazarlık veya mesajlaşma yoluyla iletişim kurabildikleri sistemlere ise çoklu etmen sistemleri adı verilir.

2.2 Etmenlerin Özellikleri

Etmenlerin, değişik tiplerine göre sahip oldukları bazı özellikler vardır. Etmenler, bu özelliklerin bir ya da birkaçına sahip olabilirler. Bu özellikler şu şekilde sıralanır:

- Bağımsız hareket etme: Kendi başına karar verme yeteneğidir.
- Mantıklılık: Davranışlarını, hedeflerini eksiksiz veya en uygun bir şekilde başarabilmek için değiştirebilme yeteneğidir.
- Reaktiflik: Çevresindeki değişikliklere tepki verebilme yeteneğidir.
- Proaktiflik: Çevresini etkileyen işlemler yapabilme yeteneğidir. Proaktif etmenler sadece çevrelerine tepki vermezler, bulunduğu ortamda değişiklikler yaparak ortamdaki diğer etmenlerin tepki vermesine neden olabilirler.

- Haberleşme: Başka kullanıcılar ve etmenlerle haberleşerek birlikte çalışabilme yeteneğidir. Bu özelliğe sahip olan etmenler, belirlenmiş bir iletişim dili kullanarak haberleşirler.
- Süreklilik: Uzun bir süre sonlanmadan çalışabilme yeteneğidir. Bir etmen, bulunduğu ortamda hedefine ulaşana kadar sonlanmadan çalışabilmelidir.
- Hareketlilik: Bir etmen ortamından başka bir etmen ortamına taşınabilme yeteneğidir. Bu ortamlar farklı fiziksel konumlarda bulunabilir. Taşınma sırasında etmenin tüm kodları ve durumu taşınabilir.
- Adaptiflik: Etmenin çevresine uyumlu bir şekilde çalışabilme yeteneğidir. Uyumlu çalışması için, çevresinin durumunu öğrenme özelliği olmalıdır.
- Karaktere sahip olma: İnanılabilir bir şekilde tasarlanmış kişiliğe sahip olma özelliğidir [3].

2.3 Etmen Tipleri

Etmenler sahip oldukları değişik özelliklere göre sınıflandırılırlar. Tüm etmenler bağımsız hareket etme özelliğine sahiptir. Bu özelliğe farklı özellikler eklendikçe farklı etmen tipleri meydana gelir.

2.3.1 Akıllı etmenler

Bir etmen, bulunduğu çevre içerisinde esnek bir şekilde hareket edebiliyorsa akıllı etmen olarak sınıflandırılır. Esnek hareketin özellikleri de şu şekilde sıralanır [2]:

- Reaktiflik (Tepki verebilme) : Akıllı etmenler sürekli çevresi ile iletişim halindedir ve çevredeki değişikliklere tepki verebilirler.
- Proaktiflik (Önleyici tedbirler alabilme) : Proaktif etmenler sadece çevresindeki değişikliklere tepki vererek değil, inisiyatif alarak da çalışabilirler. Akıllı etmenler, aynı ortamda bulunan diğer etmenlerin kararlarında değişiklik yapmalarına neden olabilecek değişiklikler yapabilirler.
- Sosyallik (Birlikte hareket edebilme) : Sosyal etmenler başka bir etmenle birlikte çalışabilirler ve bir etmen iletişim dili aracılığıyla başka etmenlerle haberleşebilirler.

2.3.2 İşbirliği etmenleri

Bu etmenlerin temel özellikleri, bir hedefe ulaşmak için birlikte çalışabilme, bağımsız hareket etme, reaktiflik ve proaktifliktir. Etmenler çeşitli konularda, zaman-kısıtlı bir çoklu etmen ortamında birbirleri ile haberleşerek ve pazarlık yaparak kendi hedeflerine ulaşmaya çalışırlar. Öğrenme yeteneği genellikle gerekli değildir veya kısıtlıdır [4].

2.3.3 Arabirim etmenleri

Kullanıcı için bir kişisel asistan görevi gören etmenlerdir. Bu nedenle öğrenme ve bağımsız hareket etme özelliklerine sahiptirler. Arabirim etmenleri aynı anda hem kullanıcı ile hem de başka etmenler ile haberleşebilirler. Kullanıcının eğilimlerini öğrenerek onun adına daha iyi kararlar verebilir veya kullanıcının kararlarını daha iyi vermesi için ona önerilerde bulunabilirler [4].

2.3.4 Hareketli etmenler

Bir ortamdan başka bir ortama kendiliğinden kodunu ve durumunu taşıyarak diğer ortamda çalışmaya devam etme yeteneği olan etmenlere hareketli etmenler adı verilir. Hareketli etmenler bir kere taşındıktan sonra diğer ortamda çalışmaya devam ettikleri için ağdaki mesajlaşmalardan dolayı oluşan yükü azaltırlar. Ayrıca ağ gecikmeleri nedeniyle oluşan performans problemleri aşılmış olur [5].

2.3.5 Bilgi etmenleri

Bilgi etmenlerine olan ihtiyaç günümüzdeki yoğun bilgi akışının ihtiyaçlara göre düzenlenmesine olan ihtiyaçtan doğmuştur. Bilgi etmenleri, çeşitli dağıtık kaynaklardan gelen bilgileri yönetir ve düzenlerler. Durgun ya da hareketli olabilirler. Birlikte çalışabilirler veya tek başlarına çalışabilirler. Öğrenme yeteneğine ihtiyaçları her zaman yoktur. Yani çalışmaları için tam olarak belirlenmiş standart özellikler yoktur [4].

2.3.6 Reaktif etmenler

Reaktif etmen, özel bir etmen tipidir. Reaktif etmenler, bulundukları ortamların dâhili ve sembolik modellerine sahip değillerdir. Sadece bulundukları ortamdan gelen etkilere karşı tepki vererek çalışırlar. Genelde çok basit olurlar ve çevresindeki etmenlerle karmaşık olmayan yöntemlerle haberleşirler [4].

2.3.7 Melez etmenler

Tanımlanmış etmen tipleri, her türlü uygulamada ihtiyacı karşılamayabilir. Bu nedenle farklı etmen tiplerinin çeşitli özelliklerini aynı anda bulunduran etmenlere sıklıkla ihtiyaç duyulur. Aynı anda birden fazla etmen tipinin özelliklerini barındıran etmenler melez etmenlerdir. Örneğin, hareketli bir bilgi etmeni melez etmenlere bir örnektir [4].

2.4 Etmenlerin Uygulama Alanları

Çoklu etmen sistemlerinin endüstriyel ve ticari alanlarda kullanılması önerilen ve kullanılan birçok uygulaması vardır. Bu uygulama alanlarına verilebilecek örnekler şu şekilde sıralanabilir:

- Alıcı ve satıcı etmenlerin kullanıcılar adına ürün satın alabildiği elektronik ticaret uygulamaları [2]. Bu uygulamalara örnek olarak kullanıcı adına en ucuz CD'leri araştıran BargainFinder sistemi ve çevrimiçi mağazalarda ürünlerin bulunabilirlik ve fiyat bilgilerini araştıran Jango sistemi verilebilir [6].
- Şehir ve hava trafiğinin akışını yönetme ve geliştirme. Bu sistemlere örnek olarak Avustralya'da, Sidney Havalimanı'nda kullanılan OASIS sistemi verilebilir. OASIS, etmen tabanlı bir hava trafiği yönetim sistemidir. Bir uçak Sidney hava sahasına girdiğinde, bu uçak için yeni bir etmen oluşturulur ve gerçek uçağın bilgi ve hedefleri bu etmene yüklenir. Örneğin, bir uçağın havalimanındaki belirli bir piste belirli bir saatte inme planı olabilir. Hava trafik kontrol etmenleri ise sistemi yönetmekle sorumludur [6].
- Ev içi, şehir içi, ulusal ya da uluslararası taşıma sistemlerinin modellenmesi ve iyileştirilmesi. Bu uygulamalarda etmenler; nakliyat araçlarını, nakliye edilecek ürünleri veya taşınacak müşterileri temsil edebilirler [2]. Etmenlerin taşıma sistemlerinde kullanımı alanında, birlikte işe giden çalışanların arabalarını ortak kullanarak masraflarını bölüştükleri bir sistem gerçeklemek için, bir çoklu etmen sistemi mimarisi önerilmiştir [7]. Bu sistemde iki tip etmen bulunur. Birincisi araba servisi sunan veya isteyen müşteri etmenlerini, ikincisi ise müşterilerin birlikte işe gitmek için buluştukları merkezleri temsil eden etmenlerdir. Müşteri etmenleri, kendilerine uygun olan merkezlere,

nereye ve ne zaman gitmek istediklerini bildirirler. Merkez etmenleri ise bu isteğe uygun bir araba var ise bu arabaya ait bilgileri müşteri etmenlerine gönderirler.

- İletişim ağlarının gerçek zamanlı izlenmesi ve yönetimi. Etmenler bu sistemlerde arama yönlendirme, sinyal anahtarlama ve mesaj taşıma gibi işlemlerden sorumludurlar [2]. Bu uygulamalara örnek olarak ATM ağlarında iletişim kanallarını trafik kaynaklarına dağıtan bir dağıtık sistem verilebilir [8]. Bu sistemde, her trafik kaynağına bir etmen atanmıştır. Bir trafik kaynağından yeni bir çağrı yapıldığında, o trafik kaynağına atanmış olan etmen en az kullanılmakta olan iletişim kanalına istek yapar. Başka bir örnek olarak da, iletişim ağlarında yük dengelemesi yapmak için önerilen bir hareketli çoklu etmen sistemi verilebilir [9]. Bu sistemde iki tip hareketli etmen bulunur. Birinci tip etmen, ağdaki düğümler arasındaki en kısa yolları, sürekli ağı dolaşarak bulur. İkinci tip etmen ise, ağda düğümler arasında dolaşarak ağı en çok kullanan ve meşgul eden düğümleri tespit eder ve yük dengelemesi için gerekli işlemleri yapar.
- Kullanıcılar adına bilgi toplama, toplanan bilgileri düzenleme ve filtreleme [2]. Bu alandaki uygulamalara örnek olarak kullanıcı adına hisse senedi değerlerini, ekonomi haberlerini ve ekonomi analistlerinin raporlarını izleyip takip ederek kullanıcının borsa portföyünü yönetmesine yardımcı olan WARREN uygulaması ve kullanıcının ilgi alanlarını öğrenerek kullanıcıya özel bir gazete oluşturan WEBMATE sistemi verilebilir [6].

3. JAVA AGENT DEVELOPMENT FRAMEWORK (JADE)

3.1 Etmen Geliştirme Platformları

Etmen sistemleri geliştirmek için, geliştiricilerin işlerini kolaylaştırıcı birçok etmen geliştirme platformu mevcuttur. Bu platformlara örnek olarak aşağıdakiler verilebilir:

- JADE (Java Agent Development Framework): Java dili ile 2001 yılından beri geliştirilmekte olan etmen geliştirme platformudur. FIPA standartlarına uygun olarak etmen geliştirme, hata ayıklama ve kurulumu için birçok görsel araç bulundurur [10].
- Repast Simphony (Recursive Porous Agent Simulation Toolkit): Etmenleri modellemeyi, yaratmayı ve simülasyonunu kolaylaştırmak için geliştirilmiş bir araçtır. Çeşitli programlama dillerinde gerçeklenmiştir [11].
- Swarm: Bir çoklu etmen simülasyon paketidir. Etmenler arasındaki haberleşmenin simülasyonunu gerçekleştirmek için kullanışlı bir platformdur [12].
- Netlogo: NetLogo, bir çoklu etmen programlama ve modelleme ortamıdır. Java ve Scala dillerinde yazılmıştır. Java sanal makinesi üzerinde çalışır [13].
- MASON (Multi-Agent Simulator of Neighbourhoods): MASON, bir çoklu etmen simülasyon kütüphanesi çekirdeğidir. Java dilinde yazılmıştır. Bir model kütüphanesi ve isteğe bağlı görsel araçlar bulundurur [14].

3.2 JADE Platformunun Özellikleri

JADE, TILAB tarafından geliştirilen ve dağıtılan, noktadan noktaya iletişim mimarisini temel alan, FIPA standartlarına uygun olarak tasarlanmış bir dağıtık çoklu etmen geliştirme platformudur [15]. Yürütme anında bir etmen platformu, ağdaki aynı işletim sistemine sahip olmayan farklı makineler üzerine dağıtılabilir ve uzaktan bir grafik arabirimi ile yönetilebilir. İletişim mimarisi, etmenler arasında esnek ve verimli bir şekilde, eş zamanlı ve eş zamanlı olmayan haberleşmeye izin verir. FIPA

etmen ve iletişim modeli tamamen desteklenecek şekilde gerçeklenmiştir ve sistemle bütünleştirilmiştir. JADE, FIPA üyesi olan veya olmayan birçok şirket ve akademik grup tarafından kullanılmaktadır. JADE'in kullanıcılarına, kullanımı ve özelleştirmeyi kolaylaştırmak için hazır olarak sunduğu özellikler aşağıdaki gibi sıralanabilir:

- Etmenler, tam bir dağıtık sistem oluştururlar. Her etmen, farklı bir iplik olarak çalışır ve farklı makineler üzerinde çalışabilir. Etmenler bulundukları makinenin ve işletim sistemlerinin özelliklerinden bağımsız olarak JADE'in haberleşme altyapısını kullanabilirler [16].
- JADE, FIPA standartlarıyla tamamen uyumludur. Böylece aynı standardı kullanan farklı etmenler, birbirleriyle kolayca haberleşebilirler. FIPA iletişim şartnamesinin kullanılması için gerekli olan kütüphaneler kullanıma hazır şekilde geliştiricilere sunulmuştur [15].
- Eş zamanlı olmayan mesajların etkin bir şekilde taşınabilmesi sağlanmıştır. Ontoloji (Ontology) nesneleri ve içerik dilleri kullanılarak, Java nesnelerinin FIPA-uyumlu mesajlara otomatik olarak dönüştürülmesi, FIPA-uyumlu mesajların Java nesnelerine otomatik olarak dönüştürülmesi sağlanmıştır [16].
- JADE, davranış modeli ile etmenlerin çoklu, paralel ve eş zamanlı işlem yapmasını destekler. Etmenler, görevlerini ve çalışmalarını davranış adı verilen nesneler aracılığıyla yaparlar. Etmenin bir davranışı, başka bir davranışının çalışmasını bölemez. Bu yöntem ile etmenin farklı davranışlarının aynı veri üzerinde işlem yapmaları engellenmiştir [17].
- Etmenlerin, dağıtık etmen platformunda benzersiz bir şekilde isimlendirilmesi JADE tarafından otomatik olarak yapılmaktadır. Her etmen yaratıldığında, ona benzersiz bir isim atanır ve arama servisine kaydedilir. Etmenler, sağlanan API'ler ve görsel araçlar sayesinde yerel ortamdan veya uzaktan başlatılabilir, durdurulabilir, dondurulabilir, devam ettirilebilir, taşınabilir, kopyalanabilir ve öldürülebilir [16].
- Programcılara hata giderme ve yönetme işlemlerinde kolaylık sağlamak için birçok görsel araç bulundurur [16].

- JSP, servlet, applet gibi ağ tabanlı teknolojileri destekler [16].
- Etmen olmayan ve JADE ortamına bağlanmamış dış uygulamalar tarafından da etmenlerin yaratılmasını sağlayan bir arabirim mevcuttur [17].
- Java gibi yaygın, güçlü ve esnek bir programlama dili ile gerçeklenmiştir ve açık kaynak kodludur [15].

3.2.1 FIPA

FIPA, etmenler ve etmen tabanlı sistemlerin birlikte çalışabilmesi için gerekli standartları belirleyerek etmen sistemler endüstrisini teşvik etmeyi amaçlayan uluslararası bir organizasyondur [18]. FIPA'nın belirlediği standartlar, soyut bir etmen sistemi mimarisi belirler ve geliştirilecek mimariye en az sayıda kısıt getirmeyi hedefler. FIPA'nın belirlediği standartlarda en çok, ticari ve endüstriyel kullanım amaçlı etmen ortamları hedeflenmiştir [19]. JADE, FIPA tarafından belirlenen soyut etmen mimarisine göre tasarlanmıştır ve bileşenleri bu mimariye göre belirlenmiştir.

3.2.2 Etmen platformu

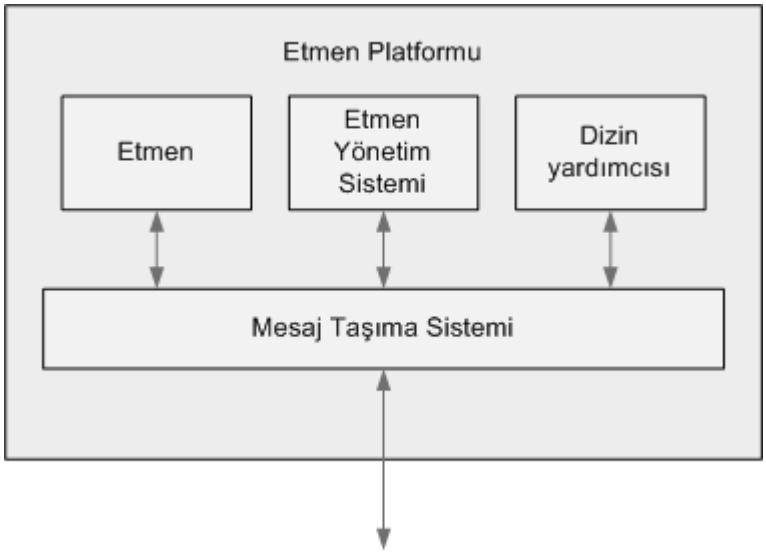
JADE çalışma ortamının her çalışan örneğine Taşıyıcı (Container) adı verilir. Her taşıyıcı birden fazla etmeni aynı anda bulundurabilir. Aktif taşıyıcılar topluluğuna Platform adı verilir. Bir platformda her zaman bir ana taşıyıcı aktif halde bulunur ve diğer taşıyıcılar çalışmaya başladıkları anda platforma kayıt olurlar. Bir platformda açılan ilk taşıyıcı, ana taşıyıcıdır ve daha sonra başlatılan taşıyıcılar normal taşıyıcılardır. Bir normal taşıyıcı başlatılırken, kayıt olacağı ana taşıyıcının adresinin ona bildirilmesi gereklidir [18].

Etmen platformu, FIPA mimarisinde tanımlanan ve Şekil 3.1'de görülen bileşenlerden oluşur. Etmen Yönetim Sistemi (AMS), etmen platformuna erişim üzerinde yönetici yetkiye sahip olan etmendir. Bir etmen platformunda en fazla 1 adet AMS etmeni bulunabilir. Etmen ortamına beyaz-sayfa (white-pages) ve yaşam-döngüsü (life-cycle) servislerini sağlar. Ayrıca etmen kimliklerinin (AID) ve durumlarının dizinini tutar. Yaratılan her etmenin geçerli bir kimlik alması için AMS'ye kayıt olması gerekir.

Dizin Yardımcısı (DF), platform üzerinde sarı-sayfalar (yellow-pages) servisini sağlar. Etmenler kayıt olma, kayıt silme, kayıt güncelleme ve arama işlemlerini

yapmak için DF ile haberleşirler. Etmen servis sunmak istediğinde DF'ye kayıt olur. Bu hizmeti kullanmak isteyen etmen ise, hizmeti veren etmenin kimlik bilgisini öğrenmek için DF'yi kullanır.

Mesaj Taşıma Sistemi (MTS) ise, platform üzerindeki etmenler arasında mesaj iletimini kontrol eder. Bu mesajlara, uzak platformlardan alınan ve uzak platformlara gönderilen mesajlar da dâhildir [17].



Şekil 3.1 : FIPA Etmen Platformu Örnek Mimarisi.

JADE platformu başlatıldığında ilk olarak AMS ve DF etmenleri yaratılır. Mesaj Taşıma Sistemi de mesaj iletimini sağlayacak duruma getirilir. Her konak üzerinde sadece bir Java uygulaması çalıştırılır. Bu uygulamadaki her etmenin çalışması birer iplikle sağlanır. Bu şekilde aynı konak üzerinde birden fazla etmenin etkin bir şekilde çalışması sağlanır. Her etmenin benzersiz kimliği JADE tarafından sağlanan AID nesnesi ile sağlanır. AID nesnesi, etmenin benzersiz kimliğini *<yerelad>@<platformadı>* şeklinde tutar. Platform adı, konak ada JADE RMI kayıt defterinin port numarası ve JADE karakter katarı eklenerek oluşturulur. Örneğin, *Alıcı* etmeninin, JADE RMI kayıt defterinin 1099 numaralı porta bağlı olduğu *AnaBilgisayar* platformundaki benzersiz ismi *Alıcı@AnaBilgisayar:1099/JADE*'dir [16].

3.2.3 Etmen yönetimi

Jade platformunda etmenler, jade.core.Agent sınıfı kullanılarak yaratılırlar. Bu sınıftan kalıtımla türetilen sınıflar ile özelleştirilmiş etmenler yaratılır. Etmen ilk

çalıştırıldığında bu sınıfın *setup* metodu çağırılır. Bu metot ile etmen, kullanacağı kaynakları oluşturur, eğer bir servis sunacak ise kendisini DF'ye kayıt ettirir ve diğer ilklendirme işlemlerini yapar. Etmen yok edilirken *takeDown* metodu çağırılır. Bu metot ile etmen kullandığı kaynakları iade eder ve DF'ye kayıt olduysa bu kaydını siler [16].

FIPA standartlarına göre bir etmen çalışma süresi boyunca çeşitli durumlarda bulunabilir. Bu durumlar arasında geçişler *doDurumAdı* metotları aracılığıyla olur. Etmenin bulunabileceği bu durumların listesi aşağıdaki gibidir [17]:

- Başlatılmış (Initiated): Etmen oluşturulmuş, fakat AMS'ye kayıt olmamıştır. Henüz bir ismi ve adresi yoktur ve diğer etmenler ile haberleşemez.
- Etkin (Active): Etmen oluşturulmuş ve AMS'ye kayıt olmuştur. İsmi ve adresi vardır. Tüm JADE özelliklerine erişebilir durumdadır.
- Askıda (Suspended): Etmenin çalışması durdurulmuştur. Etmenin çalıştığı ipliğin çalışması ve etmenin davranışları durdurulmuştur.
- Beklemede (Waiting): Etmen bloke olmuş durumdadır ve çalışmaya devam etmek için bir koşulun gerçekleşmesini beklemektedir.
- Silinmiş (Deleted): Etmenin çalışması tamamen sonlanmıştır. Etmenin AMS kaydı silinmiş durumdadır.
- Geçiş (Transit): Bir hareketli etmen başka bir konuma taşınırken bu duruma geçer. Sistem bu etmene gönderilen mesajları tampon belleğe alır ve geçiş tamamlandığında mesajları etmene iletir.

3.2.3.1 Etmen yaratma

JADE platformu, yeni bir etmen yaratma işlemi için şu adımları uygular: etmen kurucu metodu çalıştırılır, etmene jade.core.AID sınıfı aracılığıyla benzersiz bir isim verilir, yeni etmen AMS'ye kayıt edilir, etmenin durumu "Etkin" yapılır ve son olarak etmenin *setup* metodu çağırılır. FIPA standartlarına göre bir etmenin evrensel benzersiz bir isime ve başka etmenlerin onunla haberleşebilmek için kullanacakları bir adres kümesine sahip olması gerekir.

Geliştirici, etmeni ilklendirmek için *setup* metodunu kullanmak zorundadır. Etmenin JADE platformu tarafından tanımlanmış şekilde yürütülmesi bu metodun çağrılması

ile başlar. Etmenin *setup* metodu çağrıldığı anda, etmen AMS'ye kayıt olmuş ve "Etkin" durumdadır. Programcı *setup* metodunu şu işleri yapmak için kullanabilir:

- Gerekli görüldüğü durumlarda AMS'ye kayıt edilmiş verileri değiştirmek.
- Gerekli görülen durumlarda, etmenin açıklamasını ve ortama sunduğu servisleri ayarlamak. Ayrıca eğer ihtiyaç duyulursa etmeni birden çok DF'ye kayıt ettirmek.
- Görev kuyruğuna *addBehaviour* metodunu kullanarak yeni davranışlar eklemek. Davranışlar, *setup* metodu sonlandıktan sonra çalıştırılırlar.

setup metodunda en az bir adet davranış eklemelidir. *setup* metodunun sonlanması ile otomatik olarak hazır davranış kuyruğundan alınan ilk davranış çalıştırılır ve davranışlar sonlandıkça kuyruktan alınan sıradaki davranış çalıştırılır. Bir davranış çalışırken başka bir davranış onun çalışmasını kesemez. Davranışların zamanlanması round-robin algoritmasına göre yapılır [17].

3.2.3.2 Etmen sonlandırma

Etmenin *doDelete* metodu çağrıldığında etmen sonlandırılır. Etmenin kullandığı kaynakları temizlemesi ve kayıt olduğu servislerden kaydını silmesi işlemlerinin yapılabilmesi için otomatik olarak *takeDown* metodu çağırılır. *takeDown* metodunun çağrılması ile etmen "Silinmiş" durumuna geçer; yani yok edilecektir. *takeDown* metodunun çağrıldığı sırada etmenin AMS kaydı silinmemiş haldedir. Başka etmenlere mesaj gönderebilir ve sonlandırılmakta olduğunu haber verebilir. *takeDown* metodu sonlandıktan sonra, etmen için yaratılan iplik sonlandırılır [17].

3.2.3.3 Etmen davranışları

Bir etmenin görevlerini yerine getirmesi, Davranışlar (Behaviour) aracılığı ile olur. Bir davranış, kalıtım aracılığı ile jade.core.Behaviours.Behaviour sınıfının genişletilmesi ile tanımlanır. Bu davranışın etmen tarafından yürütülebilmesi için etmen sınıfı içerisinde *addBehaviour* metoduna parametre olarak eklenerek, *addBehaviour(b)* şeklinde çağrılması gerekir. Davranışlar, etmenin açılışında çalıştırılan *setup* metodunda veya daha sonra, başka bir davranış içerisinden eklenebilir.

Behaviour sınıfını genişleten her sınıf, *action* ve *done* özet metotlarını gerçeklemek zorundadır. *action* metodunda, davranışın çalışması sırasında yapılacak işler tanımlanır. *done* metodu ile, davranışın çalışmasının sonlanıp sonlanmadığı bilgisi *true* veya *false* dönüş değerleri ile JADE platformuna bildirilir. Eğer etmen sonlanmadıysa, *action* metodunun tekrar çalıştırılması için davranış görev kuyruğuna atılır. Bir davranışın *action* metodu sonlanmadan başka bir davranışın *action* metodu çalıştırılmaz. Şekil 3.2'de gösterilen bu yöntemin faydaları aşağıdaki gibi sıralanabilir [17]:

- Her JADE etmeninin bir iplik olarak çalıştırılabilmesini sağlar, böylece performansı az olan sistemlerin de desteklenmesi sağlanır.
- Davranışlar arasındaki geçişler, Java iplikleri arasındaki geçişlerden daha hızlı olduğu için performans artışı sağlanır.
- Aynı anda birden fazla davranış aynı veri üzerinde işlem yapamayacağı için eş zamanlı çalışma sorunları için önlem almaya gerek yoktur.
- Davranışlar arasında bir geçiş olduğunda herhangi bir yığın bilgisi kayıt edilmediği için, etmenin anlık durum bilgisi alınabilir. Böylece etmenin daha sonra çalışmasına devam etmesi için sabit bir ortama kaydedilmesi ve hareketli etmenlerin durumlarını ortamlar arasında taşıması kolaylaşır.

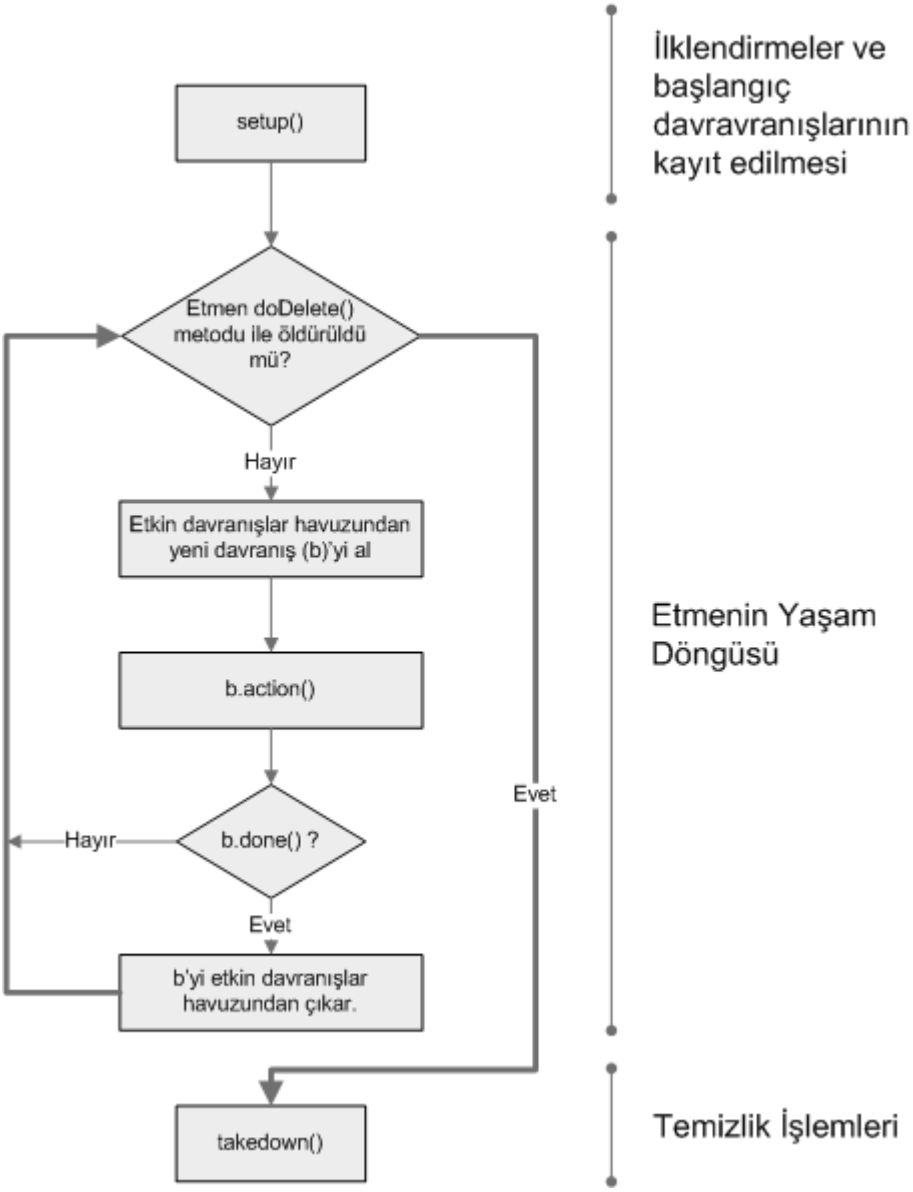
JADE kütüphanesi, davranışlarla çalışmayı kolaylaştırmak için farklı özelliklere sahip çeşitli davranış sınıfları bulundurur. Bu sınıflar kullanım amaçlarına göre doğrudan kullanılabilir veya özelleştirilmiş bir çalışma şekli için ilgili özet metotları gerçeklenerek genişletilebilir.

Behaviour

Behaviour sınıfının genişletilmesiyle, genel amaçlı davranış sınıfları oluşturulur. Oluşturulan sınıf, içerisinde bir durum değişkeni tutar ve bu değişkenin değerine göre yapacağı işleme veya davranışın sonlanmasına karar verir [18].

OneShotBehaviour

OneShotBehaviour sınıfının genişletilmesiyle, bir defa çalışan ve hemen sonlanan davranışlar oluşturulur [18]. Bu davranışlar tüm işlemlerini *action* metodunun sadece bir defa çağrılacağını göz önünde bulundurarak yapmalıdır. Davranışın sadece bir defa çağrılması, *done* metodunun varsayılan değer olarak *true* döndürmesi ile sağlanmıştır.



Şekil 3.2 : Etmenin Yaşam Döngüsü.

CyclicBehaviour

CyclicBehaviour sınıfının genişletilmesiyle oluşturan davranışlar, hiç bir zaman sonlanmazlar. Davranışın her çağrılmasında *action* metodundaki aynı işlemler tekrarlanır [16]. Davranışın sonlanmaması, *done* metodunun sürekli *false* değerini döndürmesi ile sağlanmıştır. Davranış, hazır davranışlar havuzundan *removeBehaviour* metodu ile çıkarılabilir.

TickerBehaviour

TickerBehaviour sınıfının genişletilmesiyle oluşturulan davranışlar, belirli zaman aralıklarıyla gerçekleştirilmesi gereken görevler için kullanılır [16]. Davranışın kurucu fonksiyonuna parametre olarak verilen zaman aralığı her dolduğunda, *onTick*

metodu çağırılır. Bu metot sonlandığında, davranış ayarlanan zaman aralığı tekrar dolduğunda çalışacak şekilde görev kuyruğuna eklenir.

WakerBehaviour

WakerBehaviour sınıfının genişletilmesiyle oluşturulan davranışlar, belirli bir süre geçtikten sonra veya belirli bir tarih ve saatte gerçekleştirilmesi istenen görevler için kullanılır [16]. Davranışın çalışması istenen tarih ve saat geldiğinde, davranışın *onWake* metodu çalıştırılır ve bu metodun çalışması bittikten sonra etmen sonlandırılır.

SequentialBehaviour

Birden fazla davranışın sırayla çalıştırılmasını sağlayan davranıştır. SequentialBehaviour sınıfı, kalıtımla genişletilmeden kullanılır. Bu sınıfın *addSubBehaviour* metodu kullanılarak, çalıştırılması istenen davranışlar sırayla eklenir. Davranış çalışmaya başladığında tüm alt davranışlarını sırayla çağırır. Bir alt davranışın *done* metodu *true* döndürdüğünde alt davranış sonlanmış kabul edilir ve sıradaki alt davranışa geçilir [16].

ParallelBehaviour

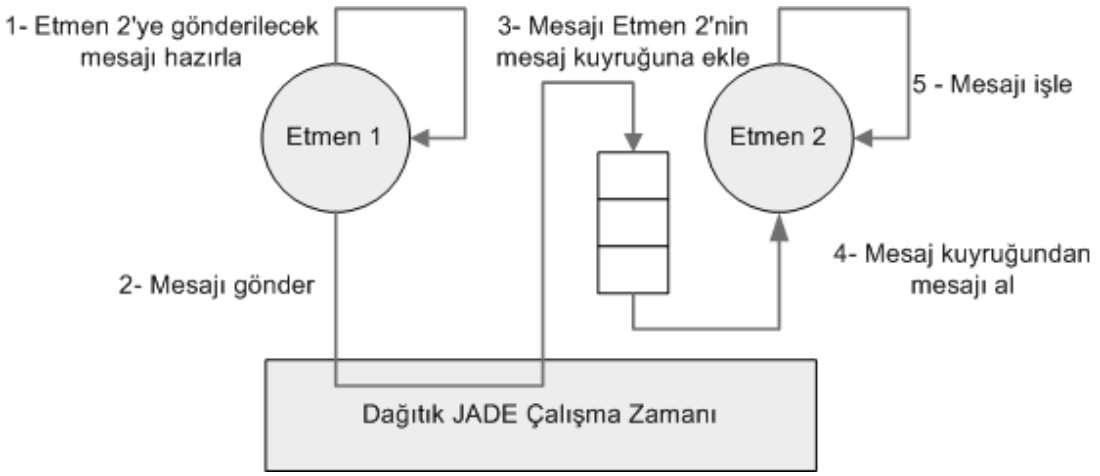
Birden fazla davranışın paralel olarak çalıştırılmasını sağlayan davranıştır. Kalıtımla genişletilmeden kullanılır. Bu sınıfa *addSubBehaviour* metodu ile çalıştırılması istenen davranışlar eklenir. Paralel davranışlar, geliştiricinin tercihine bağlı olarak bir alt davranış sonlandığında veya tüm alt davranışlar sonlandığında sonlanabilirler. Bu çalışma koşulu, kurucu fonksiyona verilen bir parametre ile belirlenir [16].

FSMBehaviour

Birden fazla davranışın, sonlu durumlu makine modeline göre çalıştırılmasını sağlayan davranıştır. FSMBehaviour sınıfı kalıtımla genişletilmeden kullanılır. Sonlu durumlu makine davranışını sağlamak için bir durum geçiş tablosu tutulur. Bu durum geçiş tablosundaki her düğüm bir alt davranışı temsil eder. Alt davranış sonlandığında, *onEnd* metodu çağrılarak dönüş değeri alınır ve çalıştırılacak yeni davranışı belirlemek için durum tablosundaki değerlerle karşılaştırılır. Karşılaştırma sonucunda çalıştırılacak yeni alt davranış belirlenir veya eğer sonlanma durumuna ulaşıldıysa ana davranış sonlandırılır [18].

3.2.4 Etmen haberleşmesi

Etmen haberleşmesi JADE'in en temel özelliğidir ve FIPA iletişim modeline uygun olarak tasarlanmıştır. İletişimin temelinde eş zamanlı olmayan mesaj taşıma işlemi vardır. Her etmenin, başka etmenlerin gönderdiği ve JADE çalışma zamanı tarafından iletilen mesajların ulaştığı bir posta kutusu (etmen mesaj kuyruğu) vardır. Posta kutusu mesaj kuyruğuna yeni bir mesaj ulaştığında etmen uyarılır. Yeni mesajın posta kutusundan ne zaman okunacağı veya okunup okunmayacağı etmen sisteminin tasarımcısı tarafından belirlenir. JADE haberleşme mimarisi, ana hatlarıyla Şekil 3.3'te verilmiştir.



Şekil 3.3 : JADE Haberleşme Mimarisi.

3.2.4.1 FIPA iletişim modeli

Etmen sistemleri için oluşturulmuş FIPA modelinin merkezinde, etmenlerin uygulama tarafından talep edilen işlemleri yapabilmeleri için birbirlerine anlamlı mesajlar aktarabilmesini sağlayan etmen haberleşmesi bulunur [19]. Bu nedenle etmenler arasında mesaj aktarım yöntemi, mesajların temsil edilme yöntemini (XML, ikili nesneler, gibi) ve diğer özellikleri tanımlayan etmen mesajlaşma dili (ACL) oluşturulmuştur. FIPA standartlarına göre bir ACL mesajında bulunan alanlar aşağıdaki gibidir [20]:

- performative: Mesajın iletişimsel eylemini belirtir.
- sender: Mesajı gönderen etmeni belirtir.
- receiver: Mesajın gönderileceği etmeni veya etmenler kümesini belirtir.
- reply-to: Mesaja verilecek cevabın gönderilmesi gereken etmeni belirtir.

- content: Mesajın içeriğini bulundurur. İçerik bilgisi alıcı etmen tarafından yorumlanır.
- language: Mesajın içeriğini bulunduran *content* alanının biçimsel dilini (SQL, FIPA-SL, Prolog, vb.)belirtir.
- encoding: *content* alanının karakter kodlamasını belirtir.
- ontology: Mesajdaki *content* alanındaki sembollere anlam verebilmek için hangi kelimeler kümesinin kullanıldığını belirtir.
- protocol: Eğer mesaj FIPA tarafından tanımlanmış bir protokole göre gönderiliyorsa, bu protokolü belirten alandır.
- conversation-id: Etmenin, yaptığı farklı mesajlaşmaları birbirinden ayırt etmek için kullandığı alandır.
- reply-with: Alıcı etmenin cevap verirken kullanması gereken ifadeyi belirtir.
- in-reply-to: Mesajın, hangi tipten bir mesaja cevap olarak gönderildiğini belirtir.
- reply-by: Alıcı etmenin, aldığı mesaja son cevap verme tarihini belirtir.

FIPA standartlarına göre bir mesajın sahip olabileceği iletişimsel eylem tipi aşağıdakilerden birisidir [21]:

- accept-proposal: Daha önce *propose* ile yapılan bir önerinin kabul edildiğini belirtir.
- agree: Daha önce *request* ile yapılması istenen bir eylemin yapılmasının kabul edildiğini belirtir.
- cancel: Bir etmenin, diğer etmene daha önce gönderdiği bir işlemin yapılması isteğinin artık geçerli olmadığını belirtmesini sağlar.
- CFP: Bir etmenin, mesajı alan etmenlerden bir öneri beklediğini belirtir.
- confirm: Göndericinin, alıcı etmene bir önermenin doğru olduğunu bildirdiğini belirtir.
- disconfirm: Göndericinin, alıcı etmene bir önermenin yanlış olduğunu bildirdiğini belirtir.

- failure: Yapılması istenen bir eylemin başarısız olduğunu belirtir.
- inform: Göndericinin, alıcı etmene bir önermenin doğru olduğunu bildirdiğini belirtir.
- inform-if: Göndericinin, alıcı etmene bir önermenin doğru veya yanlış olduğunu bildirdiğini belirtir.
- not-understood: Göndericinin, alıcının yaptığı bir eylemi anlamadığını belirtir.
- propagate: Alıcı etmenin, mesajın içeriğine gömülü bir mesajı, belirtilen etmenlere göndermesi gerektiğini belirtir.
- propose: Göndericinin, alıcı etmenin belirli koşullar altında bir eylem yapmasını önerdiğini belirtir.
- proxy: Alıcı etmenin, verilen bir açıklamaya göre gerekli etmenleri seçerek gömülü olan mesajı seçtiği etmenlere göndermesi gerektiğini belirtir.
- query-if: Göndericinin, alıcı etmene bir önermenin doğru olup olmadığını sorduğunu belirtir.
- query-ref: Göndericinin, nesnenin kaynağını belirten bir ifade ile alıcı etmenden bir nesneyi istediğini belirtir.
- refuse: Yapılması istenen bir eylemin reddedildiğini belirtir.
- reject-proposal: Daha önce *propose* ile yapılan bir önerinin kabul edilmediğini belirtir.
- request: Göndericinin, alıcı etmenden bir eylem veya başka bir iletişimsel eylem yapmasını istediğini belirtir.
- request-when: Göndericinin, alıcı etmenden bir koşul gerçekleştiğinde bir eylem yapmasını istediğini belirtir.
- request-whenever: Göndericinin, alıcı etmenden bir koşul her gerçekleştiğinde bir eylemi tekrar yapmasını istediğini belirtir.
- subscribe: Göndericinin, alıcıdan belirli bir değer her değiştiğinde haberdar edilmesini istediğini belirtir.

Bir etmen, başka bir etmenden mesaj aldığında bu mesajı anlamak ve içerisindeki bilgileri almak için, mesajın içeriğini kullanılan biçimsel dile göre ayrıştırabilmelidir. Ayrıca bu dilin kullandığı yapıları ifade etmek için kullanılan kavram ve sembollerin isimlerini de anlayabilmelidir. Bu kavram ve semboller kümesine ontoloji adı verilir. Örneğin, alıcının anlayabileceği bir biçimsel dilde gönderilmiş olan *Kitap:baslik"KitapAdı"* mesajını anlayabilmek için, alıcı etmenin *Kitap* ve *baslik* sembollerinin anlamlarının tanımlanmış olduğu bir ontolojiye ihtiyacı vardır. Gerekli ontoloji tanımına sahip olan alıcı etmen, böylece aldığı mesajda "Kitap" kelimesinin bir Kitap nesnesini ifade ettiğini ve "baslik" kelimesinin bu sınıftaki Baslik isimli bir üye değişkeni ifade ettiğini anlayabilir. [16].

3.2.4.2 ACLMessage sınıfı

JADE kütüphanesinde, bir ACL mesajı jade.lang.acl.ACLMessage sınıfı ile tanımlanır. Bu sınıf, FIPA modeline göre gerekli olan tüm alanları bulundurur. Bu alanların her biri için *get* ve *set* metotları gerçeklenmiştir ve bu alanların değerleri bu metotlarla okunur ve yazılır. FIPA tarafından tanımlanan tüm iletişimsel eylem tipleri, ACLMessage sınıfı içerisinde birer sabit üye değişken olarak tanımlanmıştır [17]

3.2.4.3 Mesaj gönderme ve alma

Mesaj gönderme işlemi, ACLMessage sınıfının *send* metodu kullanıklarak, mesaj alma işlemi ise *receive* ve *blockingReceive* metotları kullanılarak yapılır. Eş zamanlı mesaj alma işlemi için *blockingReceive*, eş zamanlı olmayan mesaj alma işlemi için ise *receive* metodu kullanılmalıdır. Eş zamanlı çalışma mümkün olduğu sürece kullanılmamalıdır çünkü bir davranış bitmeden başka bir davranış çalışamayacağı için; mesaj beklerken çalışması duran davranış, aynı etmenin diğer davranışlarının çalışmasını engelleyecektir [16]. Eş zamanlı olmayan çalışmada, *receive* metodu etmenin mesaj kuyruğunda yeni mesaj varsa alınan mesajı, eğer yeni mesaj yoksa *null* değerini döndürür. Eğer *null* değeri dönerse, etmenin çalışması *block* metodu ile durdurulmalıdır. Bu metot, davranışın çalışmasını mesaj kuyruğuna yeni bir mesaj gelene kadar durdurur. Etmen yeni bir mesaj aldığında davranış çalışmasına kaldığı noktadan devam eder. Ardından *receive* metodu tekrar çalıştırılır ve yeni mesaj okunur [18].

3.2.4.4 Biçimsel dil ve ontoloji desteği

JADE, biçimsel dil desteği için SL ve LEAP olmak üzere iki dil için kodlayıcı ve kod çözücü bulundurur. SL dili, karakter katarı tabanlı, insanlar tarafından okunabilecek şekilde tasarlanmıştır. Genellikle açık kaynaklı ve akademik çalışmalarda kullanılmak için uygundur. İnsanlar tarafından okunabilmesi, hata giderme ve test işlemlerinde oldukça yararlıdır. LEAP ise, bayt olarak kodlanmış ve insanlar tarafından okunamayacak şekilde tasarlanmıştır. SL diline göre daha az bellek harcar. Bu nedenle kullanılabilir bellek miktarı az olan, cep telefonu gibi sistemlerde kullanılması daha uygundur [22].

JADE'de ontolojiler jade.content.onto.Ontology sınıfının kalıtım yoluyla genişletilmesiyle oluşturulur. Bu sınıfta; mesajlarda kullanılan yüklemler, eylemler ve kavramların şemaları ontolojiye eklenir. Bu şemaları tanımlamak için de PredicateSchema, AgentActionSchema ve ConceptSchema arabirimlerinin kullanılması yeterlidir [17]. Ontolojiyi tanımlarken her şemanın her değişkenine, mesajlarda bu değişkeni temsil edecek ismi tek tek atamak yerine, geliştirme sürecini kısaltmak için BeanOntology eklentisi mevcuttur. Bu eklenti ile mesajlarda bir değişkeni veya nesneyi temsil edecek ad, değişkenin veya nesnenin şemadaki sınıfından otomatik olarak oluşturulur [22].

4. MÜZAYEDE

Müzayede, çeşitli ürün veya servisler için teklif verme, teklif alma ve kazanan teklif sahibine ürünü satma işlemlerinden oluşan alışveriş tipidir. Bu alışveriş işlemi için Internet çok elverişli bir ortamdır. Bu nedenle müzayede sistemleri, elektronik ticaretin önemli bir parçası haline gelmiştir. Ayrıca etmen ve yapay zekâ teknolojilerini uygulama konusunda umut verici bir alandır [23].

4.1 Müzayede Tipleri

Tek bir ürün tipinden bir adet ürünün satıldığı dört çeşit müzayede tipi mevcuttur. Bunlar, İngiliz tipi, ilk fiyat kapalı teklif tipi, Hollandalı tipi ve ikinci fiyat kapalı teklif tipi müzayedelerdir.

4.1.1 İngiliz tipi

İngiliz tipi müzayede, açık arttırma olarak da bilinir [24]. Her teklif veren, teklifini arttırmada özgürdür. Hiçbir alıcı teklif vermediğinde açık arttırma kapanır. En fazla teklif veren alıcı, açık arttırmayı kazanır [2]. Alıcıların izlemesi gereken strateji, eğer kendisinin teklifi en yüksek teklif değilse, kendisinin verebileceği en fazla teklife kadar en yüksek tekliften az miktar fazla bir teklif vermektir [23].

4.1.2 İlk fiyat kapalı teklif tipi

İlk fiyat kapalı teklif tipi müzayedelerde, her alıcı diğer alıcıların tekliflerinden habersiz olarak tek bir teklif verir. En fazla teklifi veren alıcı müzayedeyi kazanır ve verdiği teklif miktarını öder [2]. Alıcıların bu tipteki müzayedelerde izlemesi gereken strateji, müzayedeye çıkan ürün veya hizmetin gerçek değerini iyi bilmesi ve diğer alıcıların muhtemel tekliflerini doğru tahmin etmesidir [23].

4.1.3 Hollandalı tipi

Hollandalı tipi müzayedelerde satıcı çok yüksek bir fiyattan başlar ve bir alıcı mevcut fiyattan ürünü satın alana dek ürünün fiyatını düşürür [2]. Stratejik olarak ilk fiyat kapalı teklif tipinden farksızdır. Alıcının ürünün gerçek değerini iyi bilmesi ve diğer alıcıların muhtemel tekliflerini doğru tahmin etmesi gerekir [25]. Eğer doğru tahmin yapılmazsa, ürün gerçek değerinden çok yüksek bir fiyattayken satın alınabilir.

4.1.4 İkinci fiyat kapalı teklif tipi

Vickrey tipi müzayede olarak da bilinir. Her alıcı, diğer alıcıların tekliflerinden habersiz olarak tek bir teklif verir. En fazla teklif veren teklif sahibi müzayedeyi kazanır ama ikinci en fazla teklifin miktarını öder [2]. Bu tip müzayede gerçek hayatta çok fazla kullanılmamaktadır [25]. İzlenmesi gereken strateji, alıcının ürünün değerine dair kendi belirlediği fiyata teklif vermesidir [24].

4.2 Müzayede Uygulamaları

Internet üzerinde müzayede uygulamaları son yıllarda giderek daha fazla yaygınlaşmış ve eBay, Amazon.com, Yahoo!Auction, PriceLine, Ubid ve benzerleri gibi geniş bir kullanıcı kitlesi olan birçok internet tabanlı müzayede evi açılmıştır.Bu müzayede evlerine kullanıcılar Internet üzerinden bağlanarak müzayedelere katılıp ürün veya hizmet satın alabilmektedir [24]. Fakat Internet üzerindeki müzayede uygulamalarının fazlalığı nedeniyle her kullanıcının bütün müzayede evlerindeki müzayedeleri sürekli takip etmesi ve etkin teklifler oluşturması mümkün değildir. Bu nedenle müzayede takip etme ve teklif miktarı belirleme gibi işlemleri kullanıcı adına yapacak uygulamalara ihtiyaç doğmuştur. Bu uygulamalarda her etmen bir kullanıcı rolüne girerek kullanıcı adına müzayedelere katılabilir ve yeni müzayede yaratabilir.

Internet üzerindeki müzayedelere kullanıcı adına katılan çeşitli etmen sistemleri tasarlanmıştır. Bunlardan birisi, aynı anda birçok müzayede sitesine bağlanabilen BiddingBot uygulamasıdır. Bu uygulamada bir lider etmen ve her müzayede sitesine bağlanan farklı bir teklif verici etmen bulunur. Lider etmen, teklif verici etmenler arasında haberleşmeyi ve işbirliğini sağlar. Kullanıcılar ilk olarak lider etmene aranacak ürün hakkında bilgi girerler ve her teklif verici etmen kendisine atanmış

müzayede evine bağlanarak arama sonuçlarını lider etmene döndürür. Bu noktada kullanıcı her ürün için teklif parametrelerini girmek zorundadır. Bu işlemden sonra bu parametrelere göre teklif verici etmenler, aralarında işbirliği yaparak yeni teklifler oluştururlar ve müzayedelere katılırlar [26].

Müzayedeler, sadece Internet üzerinde alışverişte kullanılmamakta; çeşitli etmen tabanlı özel amaçlı müzayede sistemleri de bulunmaktadır. Bu sistemlere örnek olarak MASFIT (Multi-Agent System for Fish Trading) sistemi verilebilir. Bu sistemde, yazılım etmenleri yardımıyla alıcıların limanda yapılan toptan satış müzayedelerine katılmasına olanak sağlanmıştır [27]. Sardine isimli başka bir uygulamada, havayolu biletleri için bir etmen tabalı müzayede sistemi gerçeklenmiştir. Bu sistemde, kullanıcı kendisine uygun uçuş bilgilerini girer ve alış işlemini kullanıcı adına yapacak olan bu etmen, bu bilgileri kullanarak kullanıcı için en uygun uçuş zamanlarını bulur ve havayolu etmenlerine tekliflerini gönderir. Her havayolu için bir etmen bulunur ve bu etmenler işlem sonucunu alıcı etmenlere geri döndürürler. Bu şekilde başarılı olan teklifin rezervasyonu yapılmış olur. Son kullanıcı rezervasyonu yapılmış olan uçuşları inceleyerek hangisinin satın alınacağına karar verir [28].

5. MÜZAYEDE SİSTEMİ

Etmen sistemlerinin en etkin bir şekilde uygulanabileceği müzayede tipi, İngiliz tipi müzayedelerdir. Bu müzayedeler açık arttırma usulüne göre çalıştığı için, etmenler kullanıcı adına açık arttırmalara katılıp teklif vererek alışveriş yapabilir. İngiliz tipi müzayede sistemini geliştirmek için üç adet uygulama geliştirilmiştir:

- Veritabanı uygulaması
- Müzayede evi uygulaması
- Müşteri uygulaması

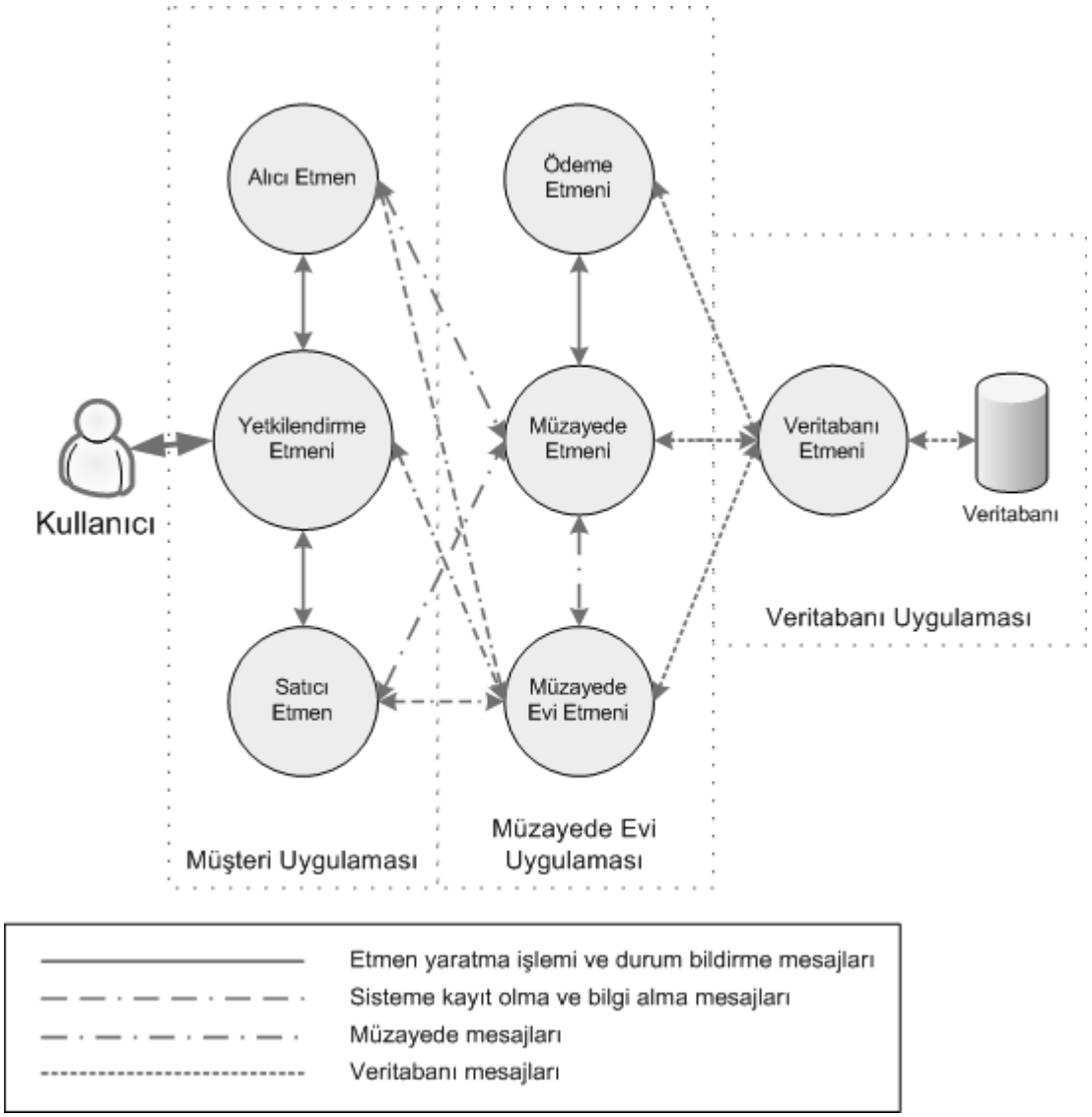
Bu üç uygulama farklı sistemlerde çalışabilir. Sistem JADE platformu üzerinde geliştirildiği için bu uygulamalar tamamen farklı mimarilerde ve farklı fiziksel konumlarda bulunabilir. Sistemin genel modeli Şekil 5.1'de verilmiştir.

Veritabanı uygulaması; bir adet veritabanı ve veritabanı etmeninden oluşur. Veritabanı etmeni, müzayede evi uygulamasından gönderilen mesajları işler ve bu mesajlara cevap verir.

Müzayede evi uygulaması; müzayede evi etmeni, müzayede etmenleri ve ödeme etmenlerinden oluşur. Müzayede etmeni, yeni müzayede açılmak istendiğinde müzayede evi etmeni tarafından yaratılır. Müzayede satıcının belirlediği zamanda veya müzayede için belirlenmiş hemen alma fiyatına erişildiğinde sonlandırılır. Müzayede sonlandığında, müzayede etmeni bir ödeme etmeni yaratır ve ödeme işlemlerinin yapılması işlemini bu etmene devreder.

Müşteri uygulaması; yetkilendirme etmeni, satıcı etmen ve alıcı etmenlerden oluşur. Kullanıcı, yetkilendirme etmeni ile etkileşim halindedir. Sisteme giriş, yeni müzayede yaratma veya müzayedeye katılma isteklerini yetkilendirme etmeni üzerinden alıcı ve satıcı etmenlere bildirir. Satıcı etmen yeni bir müzayede yaratma isteğini müzayede evi uygulamasına bildirir ve sonucu beklemeye başlar. Alıcı etmenler ise, kendilerine atanan müzayedeleri ortamdaki müzayede evlerinden

ararlar ve kendilerine cevap olarak bildirilen müzayedeleri yöneten müzayede etmenleri ile haberleşerek müzayedelere katılırlar.



Şekil 5.1 : Dağıtık Etmen Tabanlı Müzayede Sistemi Modeli.

5.1 Haberleşme Altyapısı

5.1.1 Biçimsel dil ve ontolojiler

Bütün uygulamalar, birbirleriyle haberleşmek için SL dilini kullanılırlar. Haberleşme esnasında kullanılan yapıları isimlendirmek için üç farklı ontoloji kullanılır. Bu ontolojiler, sistemdeki üç farklı mesaj grubunu oluştururlar. Her ontolojide kavram ve mesaj sınıfları tanımlanmıştır. Aktarılan her mesaj için bir sınıf tanımlıdır. Mesajda iletilecek bilgiler, ilgili mesaj sınıfına üye değişken olarak eklenen kavram

sınıfları ile belirtilir. Gönderilen mesajlar ve bu mesajlarda bulunan kavram sınıfları, etmenlerin gönderdikleri mesajların tanımlandığı bölümlerde belirtilmiştir.

5.1.1.1 Yetkilendirme ontolojisi (AuthorizationOntology)

Yetkilendirme ontolojisi, etmenler arasında aktarılan yetkilendirme mesajlarını tanımlayan ontolojidir. Her üç uygulama tarafından da kullanılır. Bu ontolojideki kavramlar ve bu kavramların içerdikleri bilgiler aşağıdaki gibi sıralanır:

- UserData: Mesajlarda kullanıcı bilgilerini temsil etmek için kullanılır. Kullanıcının ismi, şifresi, kimlik bilgileri ve iletişim bilgilerini içerir.
- DealerAgentData: Mesajlarda müşterinin satıcı etmeninin bilgilerini temsil etmek için kullanılır. Etmenin ismini, durumunu ve kendisinin açtığı müzayedelerin listesini içerir.
- BuyerAgentData: Mesajlarda müşterinin alıcı etmeninin bilgilerini temsil etmek için kullanılır. Etmenin ismini, durumunu, almak istediği ürünün ismini, izin verilen en fazla teklifi miktarını, teklif arttırma oranı bilgisini, müzayedeleri yeniden kontrol etme süresini, teklif verme stratejisini ve son satın alma tarihini içerir.
- StoreAgentData: Mesajlarda müzayede evi etmeninin bilgilerini temsil etmek için kullanılır. Etmenin isim bilgisini içerir.
- AuctionAgentData: Mesajlarda müzayede etmeninin bilgilerini temsil etmek için kullanılır. Etmenin isim bilgisini içerir.
- BankAccount: Mesajlarda bir alıcı müşterinin banka hesap bilgilerini temsil etmek için kullanılır. Hesap sahibinin ismini, hesabın tutulduğu bankayı ve banka hesap numarasını içerir.
- CreditCard: Mesajlarda bir satıcı müşterinin kredi kartı bilgilerini temsil etmek için kullanılır. Kredi kartı sahibinin ismini, kart numarasını, kartın son kullanma tarihini ve kartın CVC2 bilgisini bulundurur.

5.1.1.2 Müzayede ontolojisi (AuctionOntology)

Müzayede ontolojisi, müzayedenin yürütülmesi esnasında etmenler arasında aktarılan mesajları tanımlayan ontolojidir. Müşteri uygulaması ve müzayede evi

uygulaması tarafından kullanılır. Bu ontolojideki kavramlar ve bu kavramların içerdikleri bilgiler aşağıdaki gibi sıralanır:

- Auction: Mesajlarda bir müzayedenin bilgilerini temsil etmek için kullanılır. Müzayede evi etmeninin ismini, müzayede evinin ismini, alıcı ve satıcı etmenlerin isimlerini, müzayedenin açılış ve kapanış tarihlerini, müzayedede satılan ürünün adını, müzayedenin başlangıç fiyatını, hemen alma fiyatını, müzayedenin durumunu ve müzayedeye verilmiş en yüksek teklifi içerir.
- Bid: Müzayedeye yapılan bir teklifi temsil etmek için kullanılır. Teklifi veren etmenin ismini ve verdiği teklifi bulundurur.

5.1.1.3 Veritabanı ontolojisi (DatabaseOntology)

Veritabanı ontolojisi, veritabanına bilgi kaydetme veya veritabanından bilgi okuma esnasında aktarılan mesajları tanımlayan ontolojidir. Müzayede evi uygulaması ve veritabanı uygulaması tarafından kullanılan bu ontolojiye ait bir kavram yoktur, yalnızca mesajlar tanımlıdır.

5.2 İsimlendirme

Yeni yaratılan bir etmenin, aynı sistemdeki diğer etmenlerle çakışmayacak bir şekilde benzersiz bir isme sahip olması gerekmektedir. Bu nedenle etmen yaratılmadan önce, yaratılacak etmen için benzersiz bir isim oluşturulur. Bu benzersiz isim; etmeninin tipi, etmenin yaratıldığı sistemde oluşturulan benzersiz bir karakter katarı ve etmenin yaratıldığı sistemin IP adresi birleştirilerek oluşturulur.

5.3 Müşteri Uygulaması

Kullanıcılar müzayede sistemine bağlanmak için müşteri uygulamasını kullanırlar. Müşteri uygulaması müşteri grafik arabiriminden, yetkilendirme, satıcı ve alıcı etmenlerden oluşur. Bir müşteri aynı anda farklı müzayedelerde hem satıcı, hem alıcı rolünde bulunabilir. Kullanıcıyı satıcı rolünde olduğu müzayedelerde tek bir etmen, alıcı rolünde bulunduğu müzayedelerde ise her müzayede için farklı bir etmen temsil eder. Kullanıcının sisteme giriş yetkisini bir müzayede evinden istemesinden, sisteme kayıt olmasından ve alıcı ve satıcı etmenlerin yaratılmasından yetkilendirme etmeni sorumludur.

5.3.1 ClientManager sınıfı

Müşterinin sisteme bağlanmak için kullandığı sınıftır. Görevi, tercihler dosyasından müşterinin tercihlerini okumak ve bu tercihlere göre müzayede sistemine bağlanmak için yetkilendirme etmenini yaratmaktır. Yetkilendirme etmeni yaratıldıktan sonra uygulamanın yönetimi yetkilendirme etmenine geçer.

5.3.2 Yetkilendirme etmeni

Yetkilendirme etmeni, müşteri uygulamasının yöneticisidir. Her müşteri uygulamasında sadece bir adet yetkilendirme etmeni bulunur. Görevleri şu şekilde sıralanır:

- Ana kullanıcı arabirimini açmak ve yönetmek.
- Kullanıcının sisteme kayıt olmasını sağlamak.
- Kullanıcının sisteme giriş yapmasını sağlamak.
- Kullanıcının, istediği müzayede evinde müzayede yaratmasını sağlayacak olan satıcı etmeni yaratmak.
- Kullanıcı yeni bir müzayedeye katılmak istediğinde, onun için yeni bir alıcı etmen yaratmak.

5.3.3 İlklendirme ve çalışma

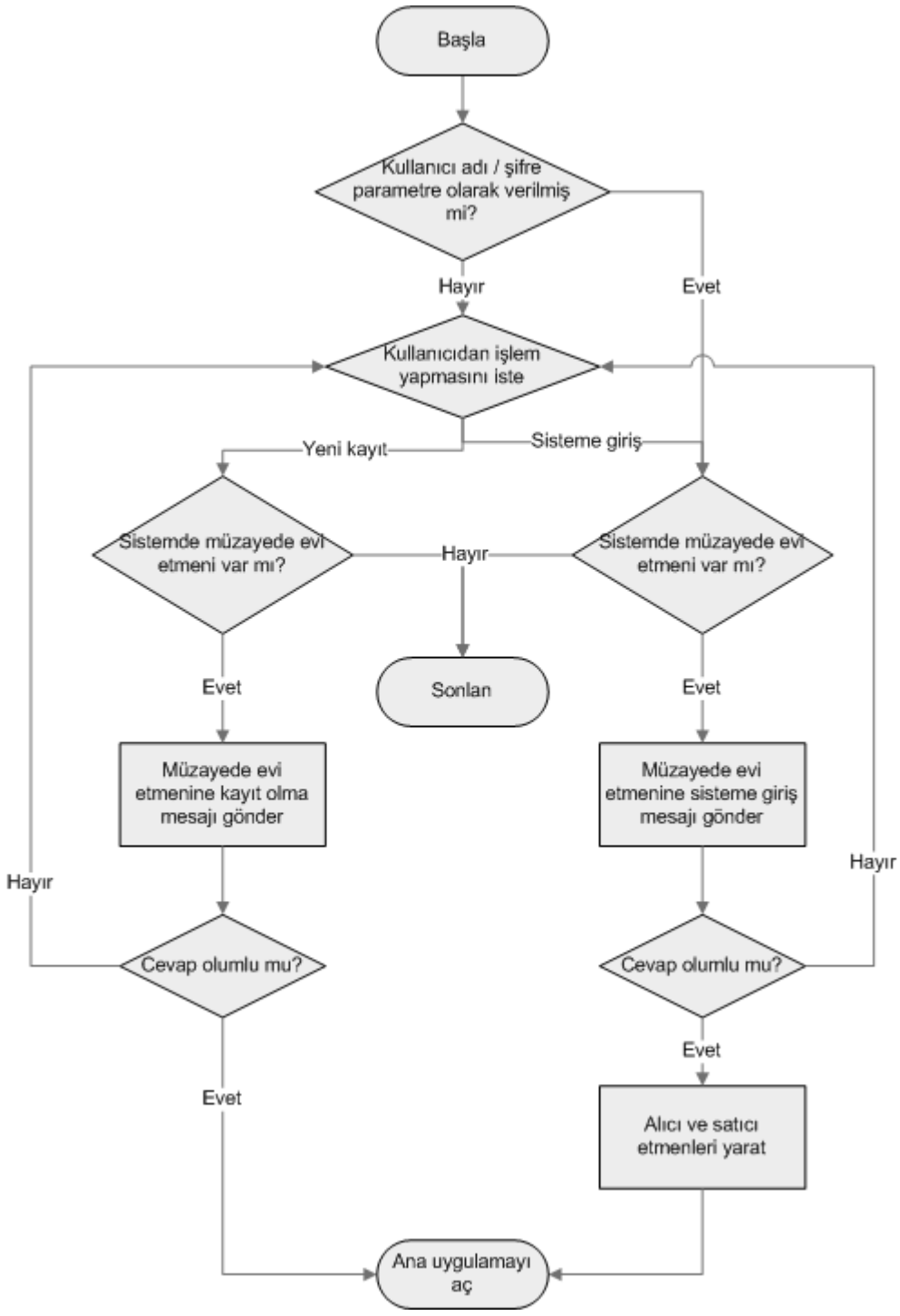
Yetkilendirme etmeninin ilklendirme işlemleri *setup* metodu ile yapılır. *setup* metodunda ilk olarak etmene verilen kullanıcı adı ve parametreleri okunur. Eğer bu parametreler etmene verilmemişse bir grafik pencere gösterilerek kullanıcının, kullanıcı adı veya şifre girmesi istenir. Kullanıcı gerekli bilgileri girdiğinde SendSignInUserBehaviour davranışını çalıştırarak sisteme giriş yapmaya çalışır. Eğer kullanıcı sisteme kayıtlı değilse, kayıt penceresine geçer ve kullanıcı gerekli bilgileri doldurduğunda SendSignUpUserBehaviour davranışını çalıştırır ve sisteme kayıt olur. Etmenin akış diyagramı Şekil 5.2'de verilmiştir.

5.3.4 Davranışlar

Yetkilendirme etmeni, iki adet davranışa sahiptir.

- SendSignUpUserBehaviour: Kullanıcının sisteme, herhangi bir müzayede evi etmenine bağlanarak kayıt olmasını sağlar. Sisteme bağlı tüm müzayede evi

etmenleri veritabanı etmeni ile haberleşebildiği için rasgele bir müzayede evi etmeni seçilir ve kayıt işlemi bu müzayede evi etmeni üzerinden yapılır.



Şekil 5.2 : Yetkilendirme Etmeninin Akış Diyagramı.

- SendSignInUserBehaviour: Sisteme daha önce kayıt olmuş bir kullanıcının, herhangi bir müzayede evi etmenini kullanarak sisteme bağlanmasını sağlar. Etmenin yetkilendirme isteğine verilen onay cevabında, kullanıcının sisteme

en son bağlandığında sisteme kayıtlı olan alıcı ve satıcı etmenlerinin bilgileri bulunur. Yetkilendirme etmeni bu bilgileri kullanarak alıcı ve satıcı etmenleri yaratır.

5.3.5 Mesajlar

Yetkilendirme etmeni, mesaj içeriği için SL dilini ve AuthorizationOntology ontolojisini kullanır. Gönderilen mesajlara ait diyalog belirteci olan conversation-id parametresi, komut için belirlenen bir komut adı katarı ve evrensel benzersiz bir karakter katarının birleştirilmesi ile oluşturulur. Yetkilendirme etmeni tarafından gönderilen mesajlar ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği aşağıda belirtilmiştir.

- SignUpUser: Yetkilendirme etmeninin, bir müzayede evi etmeni aracılığıyla sisteme kayıt olmak istediğini belirten mesajdır. Mesaj içeriğinde kullanıcı bilgisinin belirtilmesi zorunludur. Ayrıca kullanıcının tüm bilgilerinin tam olarak girilmiş olması zorunludur. Bu mesaja karşılık olarak SignUpUserReply cevap mesajı beklenir.

  sender : Yetkilendirme etmeni

  receiver : Sisteme bağlı herhangi bir müzayede evi etmeni

  conversationId : “sign-up-user-”

  performative : PROPOSE

  language : SLCodec

  ontology : AuthorizationOntology

  content : UserData nesnesi

- SignInUser: Yetkilendirme etmeninin, bir müzayede evi etmeni aracılığıyla sisteme giriş yapmak istediğini belirten mesajdır. Mesaj içeriğinde kullanıcı adı ve şifre bilgilerinin girilmiş olması zorunludur. Bu mesaja cevap olarak SignInUserReply cevap mesajı beklenir.

  sender : Yetkilendirme etmeni

  receiver : Sisteme bağlı herhangi bir müzayede evi etmeni

  conversationId : “sign-in-user-”

performative : PROPOSE

language : SLCodec

ontology : AuthorizationOntology

content : UserData nesnesi

5.3.6 Satıcı etmen

Satıcı etmen, kullanıcı adına müzayedeleri açan ve açtığı müzayedeleri izleyen etmendir. Müşteri uygulamasında bir adet bulunur ve kullanıcı tarafından açılan bütün müzayedeleri izler. Görevleri şu şekilde sıralanır:

- Kullanıcıya satış arabirimini sunar.
- Kullanıcının, tercih ettiği müzayede evinde yeni bir müzayede açmasını sağlar.
- Müzayedeleri izleyerek, müzayedelerin son durumunu takip eder.
- Kullanıcının, açtığı müzayedenin durumunu istediği zaman kontrol etmesini sağlar.
- Müzayede sonucunu kullanıcıya bildirir.

5.3.6.1 İlklendirme

Satıcı etmenin ilklendirme işlemi *setup* metodu aracılığıyla yapılır. Bu metot içerisinde, ilk önce etmene parametre olarak verilen DealerAgentData sınıfı işlenerek etmenin adı, kullanıcı bilgileri ve daha önce açılmış müzayedeler etmenin içyapılarına kaydedilir. Bu bilgiler kullanılarak grafik arabirim açılır ve arabirimde bu bilgilerin görüntülenmesi sağlanır. Satıcı etmenin, açtığı müzayedeleri kontrol eden müzayede etmenleri tarafından aranabilmesi için dizin yöneticisine kayıt yapılır ve etmene gelen bilgilendirme mesajlarını alabilmek için çevrimsel HandleInformMessagesBehaviour davranışı çalıştırılır.

5.3.6.2 Çalışma

Satıcı etmenin kullanıcı arabiriminde, etmenin aktif müzayedelerinin listesi görüntülenir. Kullanıcı bu listeden bilgilerini görmek istediği müzayedeyi seçerek, seçtiği müzayedenin bilgilerini görüntüleyebilir.

Kullanıcı müzayede yaratmak istediğinde grafik arabirimine müzayededede satacağı ürünün ismini, müzayedenin açılış fiyatını, hemen alma fiyatını ve müzayedenin sonlanma tarihini girer. Kullanıcı bu bilgileri girip onayladığında satıcı etmen SendCreateNewAuctionBehaviour davranışını çalıştırarak, kullanıcının seçtiği müzayede evi etmenine CreateNewAuction mesajını gönderir. Bu mesajda açılacak müzayedeye ait bilgiler bulunur. Eğer müzayede evi etmeni, yeni müzayedenin açılmasını uygun bulduğuna dair bir cevap verirse, alınan müzayede bilgileri ile birlikte kullanıcı arabirimi güncellenir ve yeni açılan müzayede aktif müzayedeler listesine eklenir.

Bir müzayedeye yeni teklif verildiğinde, müzayede etmeni satıcı etmene InformAuctionUpdate mesajı gönderir. Bu mesajda yeni verilen teklifin bilgileri bulunur. Bu bilgilerle kullanıcı arabirimi güncellenir. Müzayede tamamlandığında ise müzayede etmeni, sonlanmadan önce satıcı etmene InformAuctionResult mesajı ile müzayedenin tamamlandığını bildirir. Bu mesajı alan satıcı etmen, kullanıcıya müzayedenin tamamlandığını belirten bir uyarı gösterir ve müzayedeyi aktif müzayedeler listesinden çıkarır. Bir satıcı etmeninin, yarattığı tek bir müzayedenin başlamasından sonlanmasına kadar izlediği program akışı Şekil 5.3'te verilmiştir.

5.3.6.3 Davranışlar

Satıcı etmenin müzayede yaratma ve izleme görevlerine yerine getirebilmesi için çeşitli davranışları mevcuttur.

- SendCreateNewAuctionBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Kullanıcı, grafik arabirimini kullanarak yeni bir müzayede açtığında çağrılan davranıştır. Kullanıcının seçtiği müzayede etmenine bir mesaj gönderilerek müzayede açma isteği müzayede evine bildirilir. İstek kabul edilirse, cevap olarak alınan müzayede bilgileri ile kullanıcı arabirimi güncellenir.
- SendRegisterDealerAgentBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Kullanıcı yeni bir müzayede açtığında; etmen ilk defa müzayede yaratıyorsa, bilgilerini veritabanına kaydetmek için müzayede evine kayıt olma isteği gönderir. İstek kabul edilirse yeni müzayede yaratmak için SendCreateNewAuctionBehaviour davranışı çalıştırılır.

- HandleInformMessagesBehaviour: CyclicBehaviour sınıfının genişletilmesi ile gerçeklenmiştir. Çevrimsel bir şekilde, etmene gelen INFORM mesajlarını dinleyen davranıştır. Mesaj geldiğinde mesajın içeriği kontrol edilir ve içeriğe göre HandleInformAuctionUpdateBehaviour veya HandleInformAuctionResultBehaviour davranışlarından birisi çalıştırılır ve tekrar yeni bir mesaj dinlenmeye devam edilir.
- HandleInformAuctionUpdateBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Yeni bir müzayede güncelleme bilgisini içeren InformAuctionUpdate mesajı alındığında çalıştırılır. Grafik arabirimindeki müzayede bilgilerini günceller.
- HandleInformAuctionResultBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Sahip olunan bir müzayedenin sonuçlandığı bilgisini içeren InformAuctionResult mesajı alındığında çalıştırılır. Grafik arabiriminde müzayedenin sonucunu bildirir.

5.3.6.4 Mesajlar

Satıcı etmen; mesaj içeriği için AuctionOntology, AuthorizationOntology ontolojilerini ve SL dilini kullanır. Gönderilen mesajlara ait diyalog belirteci olan conversation-id parametresi, komut için belirlenen bir komut adı katarı ve evrensel benzersiz bir karakter katarının birleştirilmesi ile oluşturulur. Satıcı etmen tarafından gönderilen mesajlar ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği, aşağıda belirtilmiştir.

- CreateNewAuction: Satıcı etmenin, müzayede evi etmenine yeni bir müzayede açmak için gönderdiği mesajdır. Müzayede bilgilerini bulunduran Auction nesnesinin mesaja eklenmesi zorunludur. Bu mesaja karşılık olarak işlemin sonucunu içeren CreateNewAuctionReply cevap mesajı beklenir.

  sender : Satıcı etmen

  receiver : Müzayede evi etmeni

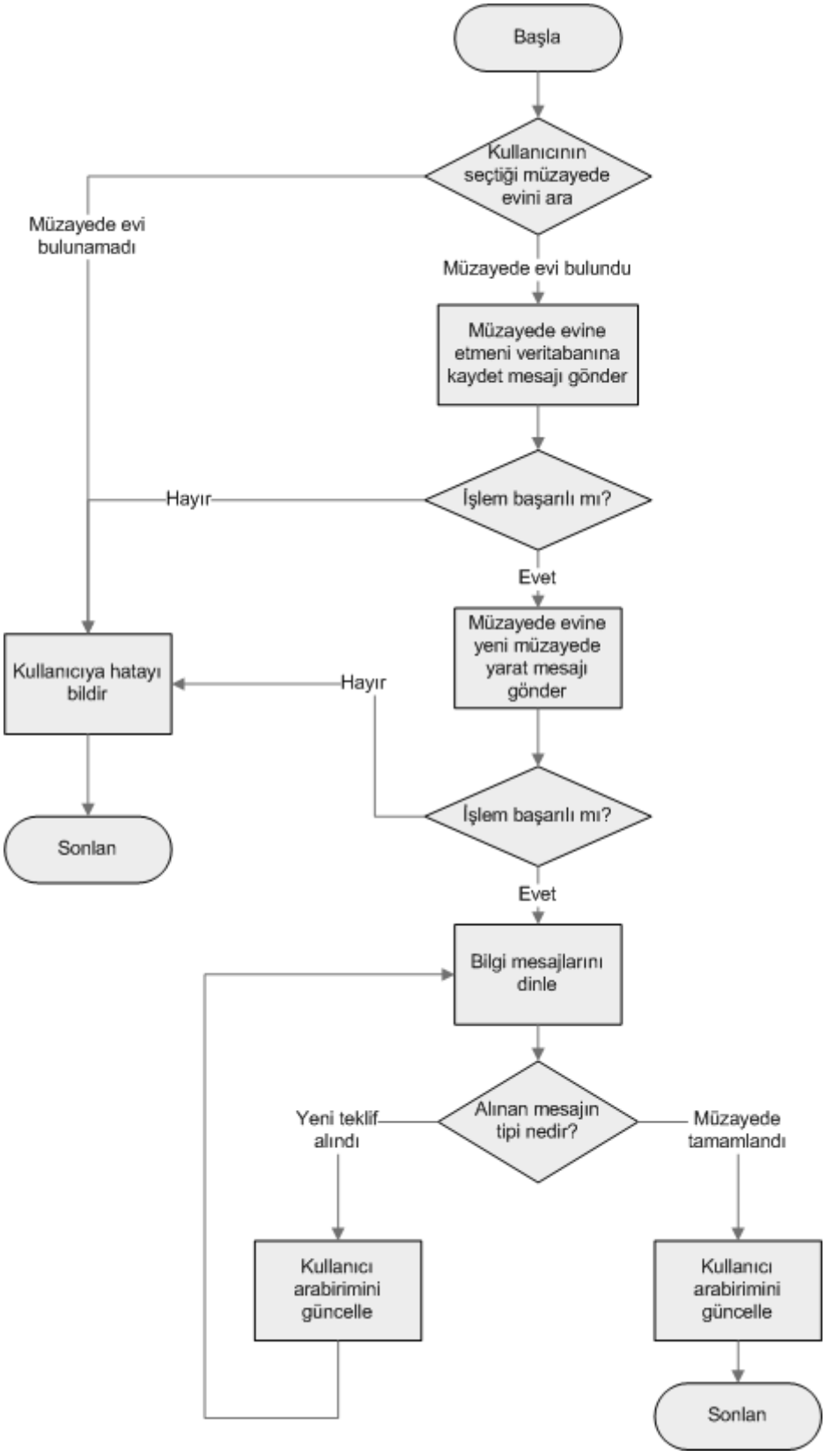
  conversationId : “create-new-auction-”

  performative : PROPOSE

  language : SLCodec

ontology : AuctionOntology

content : Auction nesnesi



Şekil 5.3 : Satıcı Etmenin Akış Diyagramı.

- RegisterDealerAgent: Satıcı etmenin, müzayede evi etmenine bilgilerini veritabanına kaydetmesi için gönderdiği mesajdır. Etmen bilgilerini bulunduran DealerAgentData nesnesinin mesaja eklenmesi zorunludur. Bu mesaja karşılık olarak işlemin sonucunu bildiren RegisterDealerAgentReply cevap mesajı beklenir.

  sender : Satıcı etmen

  receiver : Müzayede evi etmeni

  conversationId : "register-dealer-agent-"

  performative : REQUEST

  language : SLCodec

  ontology : AuthorizationOntology

  content : DealerAgentData nesnesi

5.3.7 Alıcı etmen

Alıcı etmen, kullanıcı adına müzayedelere katılır ve kullanıcının tercihleri doğrultusunda alışveriş işlemlerini yapar. Kullanıcının satın almak istediği her ürün için bir alıcı etmen oluşturulur. Bu alıcı etmen birden fazla müzayedeye katılabilir ve teklif verebilir. Alıcı etmenin görevleri şu şekilde sıralanır:

- Kullanıcıya ürün alım arabirimini ve tercihlerini sunmak.
- Ürünün satıldığı müzayedeleri, müzayede evlerine bağlanarak aramak.
- Birden çok müzayedede satışa çıkarılmış bir ürünün her müzayede için teklifini belirlemek ve müşteri için en uygun olan müzayedeye teklif vermek.
- Müzayedelerin durumunu müşterinin tercihlerine göre belirli aralıklarla takip etmek ve gerekli bir durum oluştuğunda yeni bir teklif vermek.
- Müzayedelerin sonucunu kullanıcıya bir grafik arabirim kullanarak göstermek.

5.3.7.1 Satın alma parametreleri

Kullanıcı arabiriminde, etmenin teklifi kullanıcının tercihlerine göre belirleyebilmesi için çeşitli parametreler bulunur. Bu parametreler şu şekilde sıralanır:

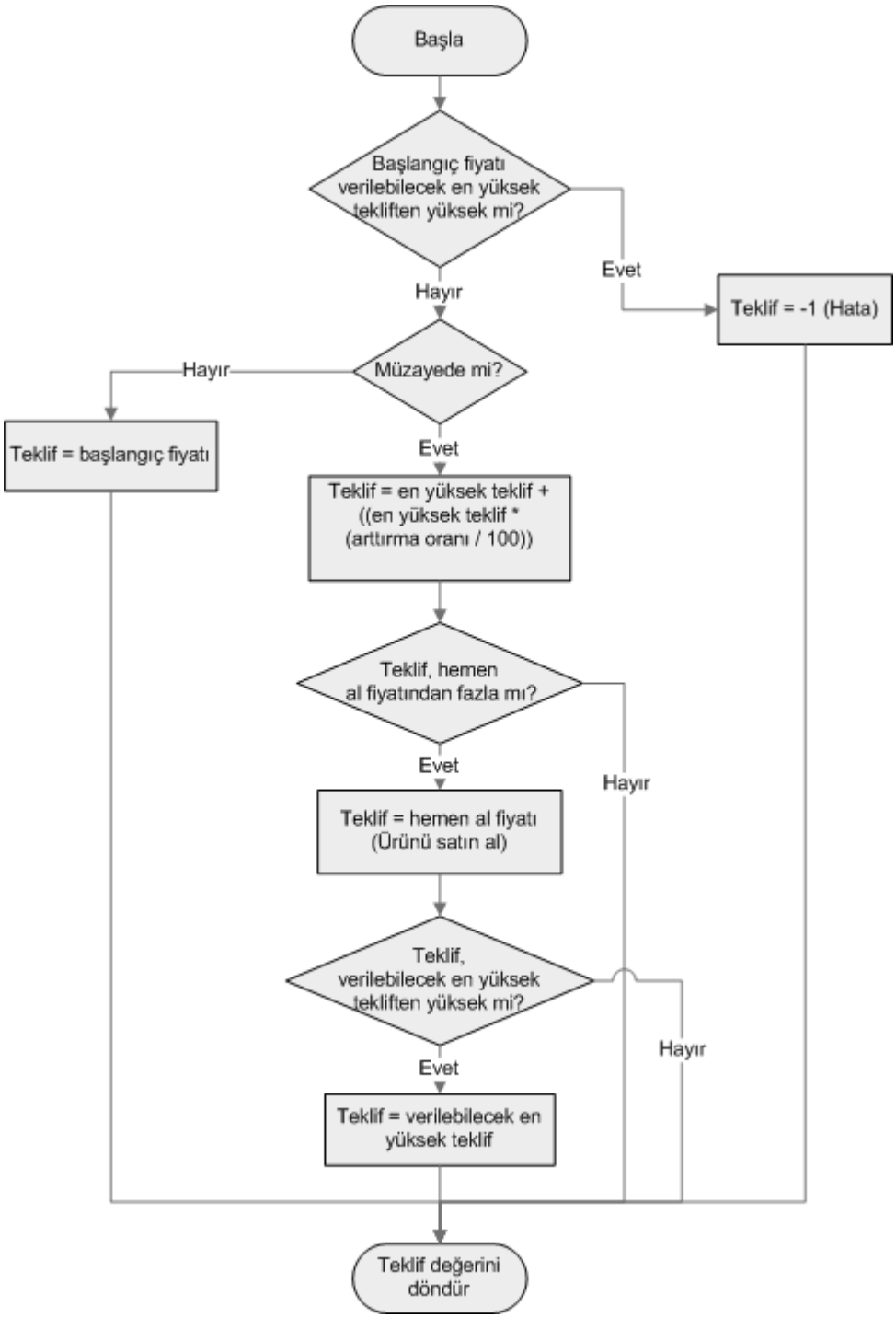
- Ürün adı: Alınacak ürünün tam adıdır. Etmenin müzayedeye katılması için, ürün adıyla açık arttırmaya çıkarılan ürünün isminin tam olarak eşleşmesi gerekmektedir.
- Verilebilecek en fazla teklif: Etmenin müzayede etmenine verebileceği en fazla tekliftir. Müzayede için hesaplanan teklif miktarı bu miktarı geçiyorsa müzayedeye teklif verilmez. Zorunlu bir parametre değildir, müşteri alıcı etmenin belirli bir fiyat sınırlaması olmadan teklif yapmasını tercih edebilir.
- Arttırma oranı: 1 ile 100 arasında değer alabilen bu alanın etkisi seçilen stratejiye göre değişiklik gösterir. Genel olarak arttırma oranı ile hesaplanan teklif miktarı doğru orantılıdır. Yüksek bir değer seçilirse, yeni hesaplanan teklif daha fazla olacaktır.
- Yeniden kontrol etme süresi: Etmenin müzayedenin durumunu sürekli kontrol edebilmesi için belirli zaman aralıklarıyla müzayede etmenine bağlanması ve müzayede durumunu sorgulaması gerekir. Bu değer dakika cinsinden müzayede etmenine bağlanma aralığını belirtir.
- Son alım tarihi: Ürünün son alım tarihini belirtir. Bir müzayedenin sonlanma tarihi, etmenin son alım tarihinden sonra ise müzayedeye girilmez. Bu kurala tek istisna, ürünü hemen alma stratejisindedir. Ürünün doğrudan alınması halinde müzayedenin sonlanma tarihi beklenmeyeceği için bu değer önemsenmez.

5.3.7.2 Teklif belirleme stratejileri

Alıcı etmenin müzayedede izleyebileceği dört farklı teklif belirleme stratejisi vardır. Strateji tipi, müzayedenin durumu ve satın alma parametreleri kullanılarak yeni teklif oluşturulur.

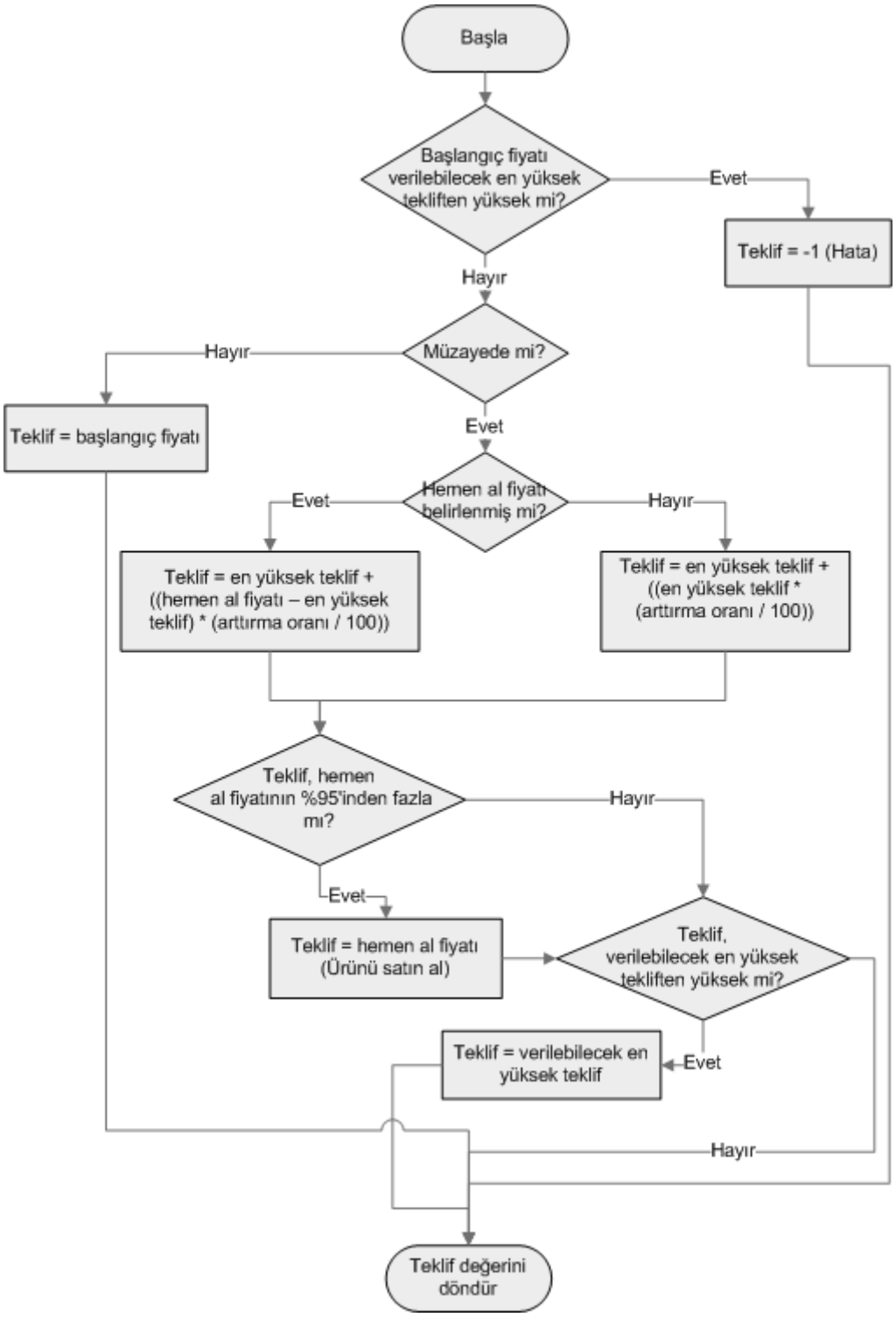
Tüm algoritmalarda ortak olarak, eğer müzayedenin başlangıç fiyatı etmenin verebileceği en fazla tekliften fazla ise teklif verilmez. Eğer müzayedenin başlangıç ve hemen alma fiyatları eşit ise, bu bir müzayede değildir ve verilecek teklif müzayedenin başlangıç fiyatı olarak belirlenir. Eğer hesaplanan teklif, ürünün hemen alma fiyatından fazlaysa; teklif hemen alma fiyatına düşürülür. Ayrıca hesaplanan teklif, verilebilecek en fazla tekliften fazlaysa; teklif, verilebilecek en fazla teklif fiyatına düşürülür.

İlk strateji, ürüne verilmiş en fazla teklife göre teklif belirleme stratejisidir. Algoritması Şekil 5.4'te gösterilen bu stratejide verilecek teklif, en fazla teklif verilen fiyat ile arttırma oranı değerinin 100 ile bölümünün çarpımıyla bulunur. Bu stratejide, her teklifte bir önceki tekliften daha çok arttırma yapılır. Eğer daha önce ürüne teklif verilmemişse, başlangıç fiyatı en fazla teklif olarak alınır.



Şekil 5.4 : Verilen En Yüksek Teklife Göre Teklif Belirleme Stratejisi.

İkinci stratejide, ürünün hemen alma fiyatı da göz önünde bulundurulur. Algoritması Şekil 5.5'te gösterilen bu stratejide, verilecek teklif hemen alma fiyatı ile en fazla teklif verilen fiyatın farkının, arttırma oranının 100 ile bölümünün çarpımıyla bulunur. Bu stratejide, her teklifte bir önceki tekliften daha az teklif yapılır. Ürünün hemen alma fiyatı yoksa en fazla teklife göre teklif verilen birinci strateji uygulanır.



Şekil 5.5 : Hemen Alma Fiyatına Göre Teklif Belirleme Stratejisi.

Üçüncü stratejide, diğer teklifler göz önünde bulundurularak teklif yapılır. Alıcı etmen, müzayede etmeninden müzayede bilgisini her aldığında ürüne verilmiş en fazla teklifi kaydeder. Yeni teklifi ise, kaydettiği eski arttırma oranlarının ortalamasını, arttırma oranının 100 ile bölümü ile çarparak belirler. Bu stratejide diğer etmenler tekliflerinde küçük arttırmalar yapıyorlarsa müşterinin etmeni de küçük arttırmalar yapar, diğer etmenler tekliflerinde büyük arttırmalar yapıyorlarsa müşterinin etmeni de büyük arttırmalar yapar. Eğer ürün için ilk teklif yapılıyor ise, ürünün başlangıç fiyatının yüzde biri kadar arttırma yapılır. Bu stratejiye göre teklif verme algoritması, Şekil 5.6'da gösterilmiştir.

Dördüncü strateji hemen alma stratejisidir. Algoritması Şekil 5.7'de gösterilen bu stratejide, eğer müzayedenin hemen alma fiyatı, alıcı etmenin en fazla verebileceği fiyattan yüksek ise ürün satın alınmaz; düşük veya eşitse ürün satın alınır.

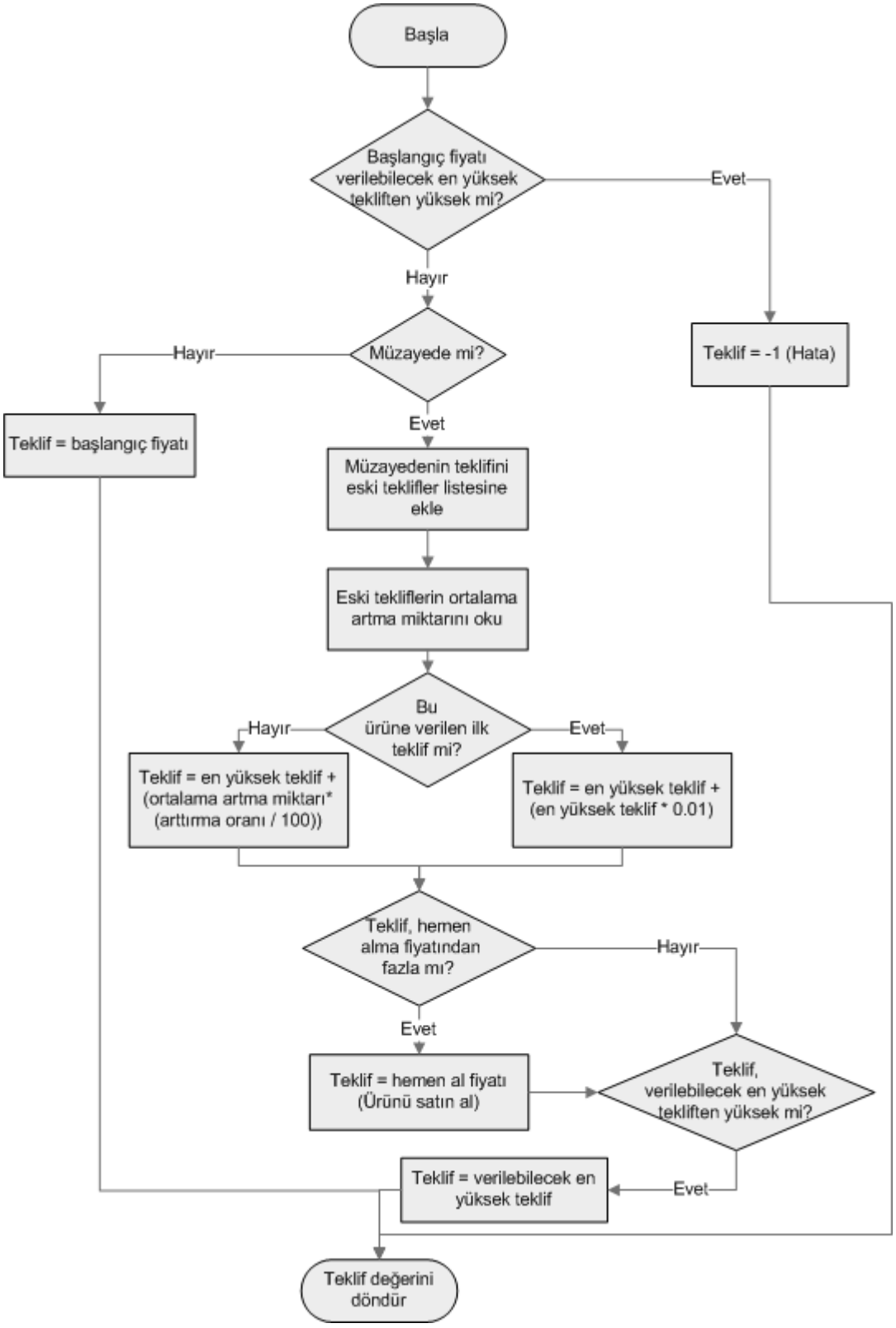
5.3.7.3 İlklendirme

Alıcı etmenin ilklendirme işlemi *setup* metodu aracılığıyla yapılır. *setup* metodunda ilk olarak etmene aktarılan parametreler okunur ve alıcı etmen daha önce yaratılmış ama satın alma işlemi sonlanmadan kapanmış ise, eski etmene ait veriler bu etmenin içyapılarına eklenir. Eğer daha önce yaratılan bir alıcı etmenin devamı değilse, müşteri kullanıcı formuna alınacak ürün için gerekli bilgileri girene kadar beklenir. Ardından müzayede evlerinde ilgili ürün aranarak teklif verme safhasına geçilir.

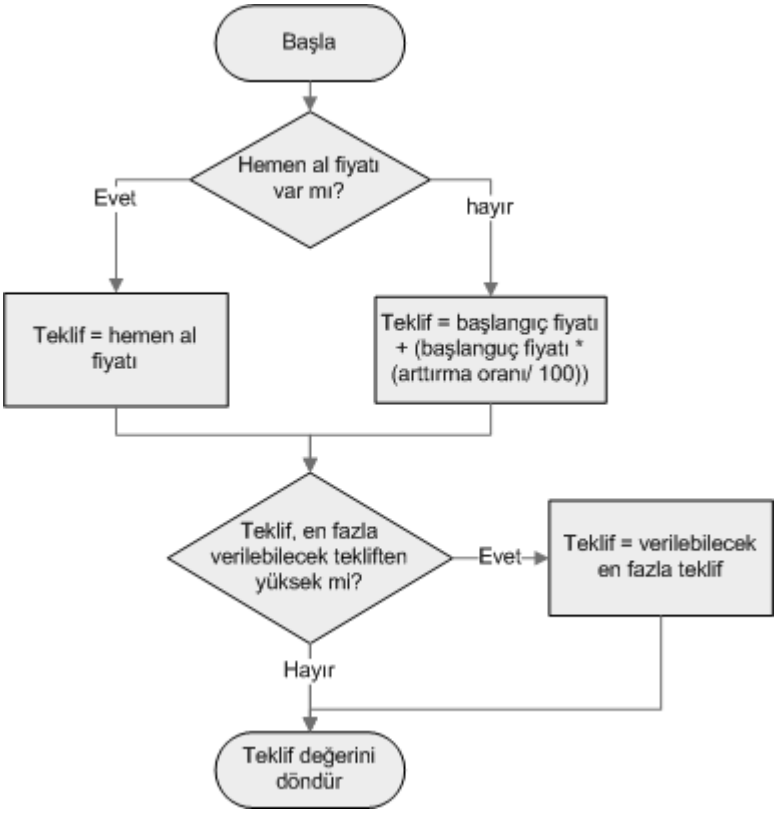
5.3.7.4 Çalışma

Alıcı etmen, müşteriden bir ürün alımına başlama komutu alındığında ilk olarak sistemdeki tüm müzayede evi etmenlerine arama yapmak için SearchItem mesajını gönderir ve aldığı cevaplardan ürünün hangi müzayedelerde satıldığını öğrenir. Arama sonucunda müzayedelere ait bilgileri de aldığı için, bu müzayede bilgilerine, kullanıcı tarafından belirlenen satın alma parametreleri ve stratejilerine göre her müzayede için yeni teklifi belirler. Belirlenen tekliflerden en düşük fiyat verilecek olan müzayedeye ProposeNewBid mesajı ile teklif verilir. Eğer daha önce teklif verilen bir müzayedeye verilmiş olan teklif hala geçerliyse, yeni teklif hesaplanmaz ve verilmez. Teklif hesaplama ve verme aşamasından sonra, alıcı etmen kullanıcının arabirimden girdiği yeniden kontrol etme süresi dolana kadar bekler. Bekleme süresi dolduğunda tekrar aynı süreç başlar. Etmenin çalışması, son alım tarihinde başarısız olarak, müzayede etmeni tarafından gönderilen InformAuctionResult mesajı ile

ürünün satın alındığı bilgisi alındığında başarılı olarak sonlanır ve kullanıcı arabirimi ile müşteri bilgilendirilir. Alıcı etmenin çalışmasını gösteren akış diyagramı Şekil 5.8'de verilmiştir.



Şekil 5.6 : Diğer Tekliflerin Ortalamasına Göre Teklif Verme Stratejisi.

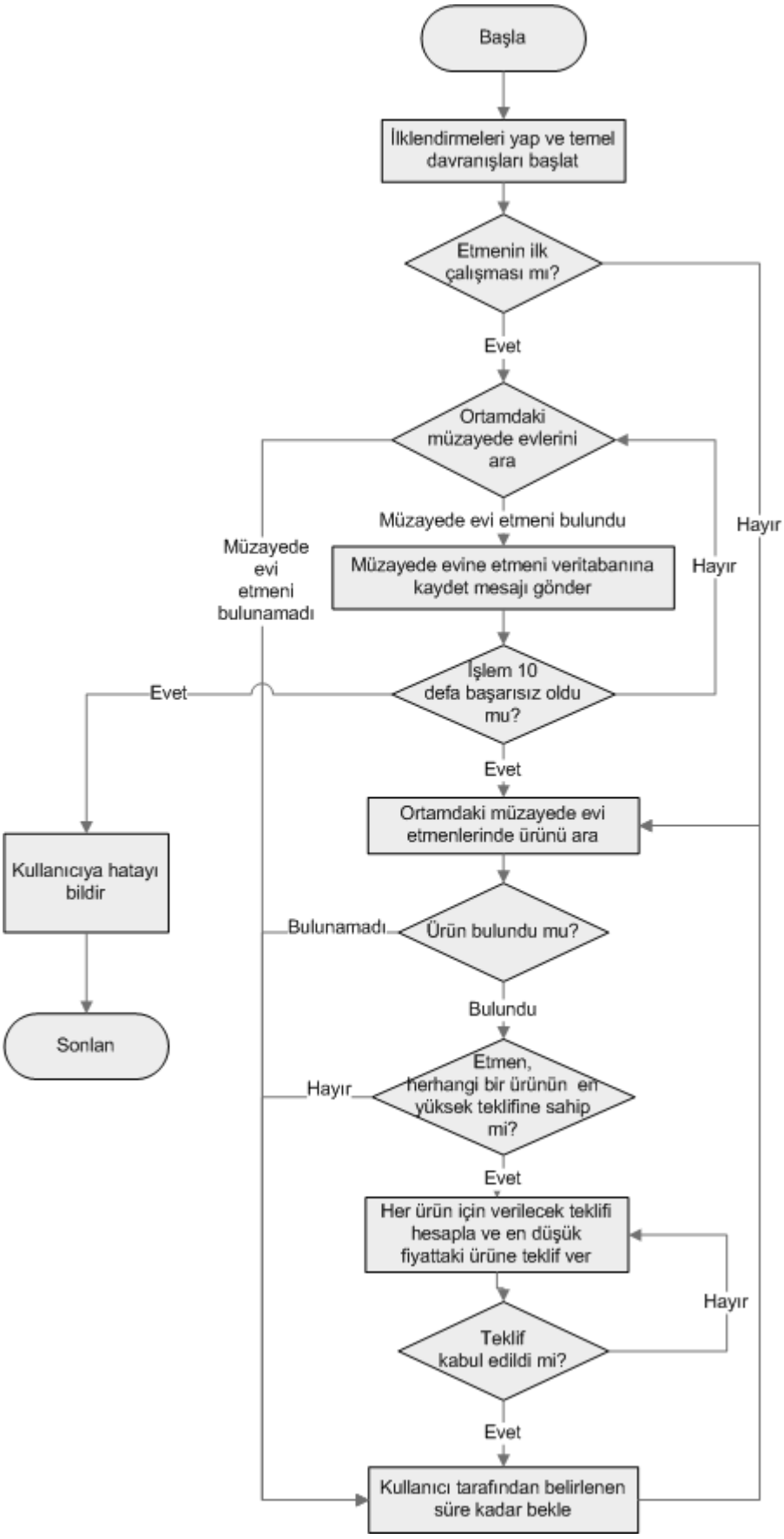


Şekil 5.7 : Hemen Alma Stratejisi.

5.3.7.5 Davranışlar

Alıcı etmenin ürün arama ve müzayedeye katılma işlemlerini yapabilmesi için çeşitli davranışlar mevcuttur.

- SendSearchItemBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Bir ürünün satıldığı müzayedelerin, müzayede evlerinde aranması için kullanılan davranıştır. Sisteme bağlı tüm müzayede evlerine arama mesajı gönderilir ve cevap olarak alınan ürün listeleri etmenin ürün listesine eklenir.
- CalculateNewBidsBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Yeni tekliflerin hesaplandığı davranıştır.
- ProposeNewBidBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Hesaplanan yeni teklifin, müzayede etmenine önerildiği davranıştır. Müzayede etmeni teklifi kabul ederse teklif verme süreci sonlanır, eğer teklif kabul edilmezse yeni teklif hesaplanır ve yapılır.



Şekil 5.8 : Alıcı Etmenin Akış Diyagramı.

- BidderBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. SendSearchItemBehaviour, CalculateNewBidsBehaviour ve ProposeNewBidBehaviour davranışlarını FSMBehaviour sınıfını kullanarak sonlu durumlu makine modeline göre çalıştırır.
- BidderInvokerBehaviour: TickerBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. TickerBehaviour sınıfı ilk çalışmasını bekleme süresini doldurduktan sonra yaptığı için, BidderBehaviour davranışı TickerBehaviour olarak gerçeklenmemiştir. İlk olarak BidderBehaviour davranışı çalıştırılır, ilk çalışmadan sonra kullanıcının belirlediği yeniden kontrol etme süresi her dolduğunda, BidderBehaviour davranışı BidderInvokerBehaviour tarafından çalıştırılır.
- HandleInformBehaviour: CyclicBehavior sınıfının genişletilmesiyle gerçeklenmiştir Etmene gelen INFORM mesajlarını çevrimsel şekilde işleyerek gerekli davranışı çalıştırır.
- HandleInformAuctionResultBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayede sonucunu bildiren InformAuctionResult mesajının alınması durumunda çalıştırılır. Bu davranış, grafik arabirimini kullanarak müşteriye müzayedenin kazanılıp ürünün satın alındığını bildirir ve etmenin çalışmasını sonlandırır.
- SendRegisterBuyerAgentBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Kullanıcı, müzayede tercihlerini girdikten sonra etmen çalışmaya başladığında, alıcı etmen bilgilerini veritabanına kaydettirmek için bu davranışı çalıştırır.

5.3.7.6 Mesajlar

Alıcı etmen; mesaj içeriği için AuctionOntology ve AuthorizationOntology ontolojilerini ve SL dilini kullanır. Gönderilen mesajların conversation-id parametresi, mesajın tipini belirten bir karakter katarına evrensel benzersiz bir karakter katarı eklenerek belirlenir. Alıcı etmenin gönderdiği mesajların ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği listesi aşağıdaki gibi sıralanır:

- SearchItem: Alıcı etmenin, müzayede evlerinde ürün aramak için gönderdiği mesajdır. Mesaja parametre olarak aranan ürünün ismi eklenir. Ürünün

isminin mesaja eklenmesi zorunludur. Bu mesaja cevap olarak mesajın gönderdiği tüm müzayede evi etmenlerinden SearchItemResult mesajı beklenir.

sender : Alıcı etmen

receiver : Sisteme bağlı bütün müzayede etmenleri

conversationId : "search-item-"

performative : REQUEST

language : SLCodec

ontology : AuctionOntology

content : String veri tipinde, aranan ürünün ismi

- ProposeNewBid: Alıcı etmenin, müzayede etmenine yeni teklif önermek için gönderdiği mesajdır. Mesaja parametre olarak teklifi içeren Bid nesnesi ile alıcı etmenin bilgilerini içeren BuyerAgentData nesnesi eklenir. Bu parametrelerin ikisi de zorunludur. Cevap olarak müzayede etmeninden önerinin sonucunu bildiren ProposeNewBidReply mesajı beklenir.

  sender : Alıcı etmen

  receiver : Müzayede etmeni

  conversationId : "propose-new-bid-"

  performative : PROPOSE

  language : SLCodec

  ontology : AuctionOntology

  content : Bid ve BuyerAgentData nesneleri

- RegisterBuyerAgent: Alıcı etmenin, bilgilerini veritabanına kaydettirmesi için müzayede evi etmenine gönderdiği mesajdır. Mesaja parametre olarak alıcı etmenin bilgilerini içeren BuyerAgentData nesnesinin eklenmesi zorunludur. Cevap olarak müzayede evi etmeninden işlemin sonucunu bildiren RegisterBuyerAgentReply mesajı beklenir.

  sender : Alıcı etmen

receiver : Müzayede evi etmeni

conversationId : “register-buyer-agent-”

performative : REQUEST

language : SLCodec

ontology : AuthorizationOntology

content : BuyerAgentData nesnesi

5.4 Müzayede Evi Uygulaması

Müzayedeler, müzayede evi uygulaması tarafından yönetilirler. Müzayede evleri müşterilerin sisteme kayıt olma, sisteme giriş yapma, ürün arama, müzayede düzenleme, müzayedeye katılma ve teklif verme isteklerini yerine getirirler. Sistemde birden fazla müzayede evi uygulaması bulunabilir. Bu uygulamalardan her birisi sisteme kayıtlı bir müzayede evini temsil eder. Müzayede evi uygulamasında bir müzayede evi etmeni, yönetilen her müzayedeye denk düşen birer müzayede etmeni ve gerek duyulduğunda yaratılan ödeme etmenleri bulunur.

5.4.1 Müzayede evi etmeni

Müzayede evi etmeni, müşterilerin ürün aramak ve müzayede açmak için bağlantı kurdukları, müzayede evinin yöneticisi olan etmendir. Her müzayede evi uygulamasında yalnızca bir adet müzayede evi etmeni bulundurur. Görevleri aşağıdaki gibi sıralanır.

- Müşterilerin ve müşteri etmenlerinin sisteme kayıt olma isteklerine cevap vermek.
- Müşterilerin sisteme giriş yapma isteklerine cevap vermek ve eğer verilen cevap olumlu ise müşterinin kayıtlı alıcı ve satıcı etmen bilgilerini göndermek.
- Müşterilerin ürün arama isteklerine cevap vermek.
- Satıcı etmenler müzayede açmak istediklerinde, yeni bir müzayede etmeni yaratarak müzayedeyi başlatmak ve başlatılan müzayede ve müzayede etmeni bilgilerini satıcı etmenlere bildirmek.

5.4.1.1 İlklendirme

Müzayede evinin ilklendirme işlemi *setup* metodu içerisinde yapılır. İlk olarak müzayede evi etmeni, servislerini dizin yöneticisine kayıt ettirir. Böylece müşteri uygulamasındaki alıcı ve satıcı etmenlerin dizin yöneticisine bağlanarak müzayede evi etmenini bulmaları sağlanır. Ardından verilen hizmetleri yerine getirmek için gerekli olan çevrimli davranışlar çalıştırılır.

Müzayede evi ilk açıldığında, veritabanı etmeninden kendisi üzerine kayıtlı olan müzayede listesini ister. Veritabanı etmeninden alınan bu listeye göre, her müzayedenin yöneticisi olan müzayede etmenleri tek tek yaratılır ve uygulamanın en son kapandığı durumda başlaması sağlanır.

5.4.1.2 Çalışma

Müzayede evi etmeni reaktif bir etmendir. Kendisine yöneltilen isteklere ve önerilere cevap verir. Şekil 5.9'da akış diyagramı verilen müzayede evi etmenine kullanıcı kayıt olma isteği gönderdiğinde, bu istek veritabanı etmenine yönlendirilir ve veritabanı etmeninden alınan cevaba göre müşterinin yetkilendirme etmenine olumlu veya olumsuz bir cevap gönderilir. Kullanıcı sisteme giriş isteği gönderdiğinde, bu istek de veritabanı etmenine yönlendirilir ve veritabanı etmeninden alınan cevaba göre yetkilendirme etmenine cevap verilir. Müşterinin alıcı etmeninden ürün arama isteği alındığında, bu istek veritabanı etmenine yönlendirilir ve veritabanı etmeninden alınan ürün listesi kullanıcının alıcı etmenine yönlendirilir. Müşterinin satıcı etmeni yeni müzayede açmak istediğinde ise, bir müzayede etmeni yaratılır ve müzayedenin kontrolü bu etmenin yönetimine verilir. Bir müzayede yaratıldıktan sonra müzayedenin kontrolü tamamen müzayede etmenine geçer. Müzayede etmeni başarılı bir şekilde açıldığını müzayede evi etmenine bildirdikten sonra, müşterinin satıcı etmenine işlemin başarılı olduğunu bildiren bir cevap gönderilir.

5.4.1.3 Davranışlar

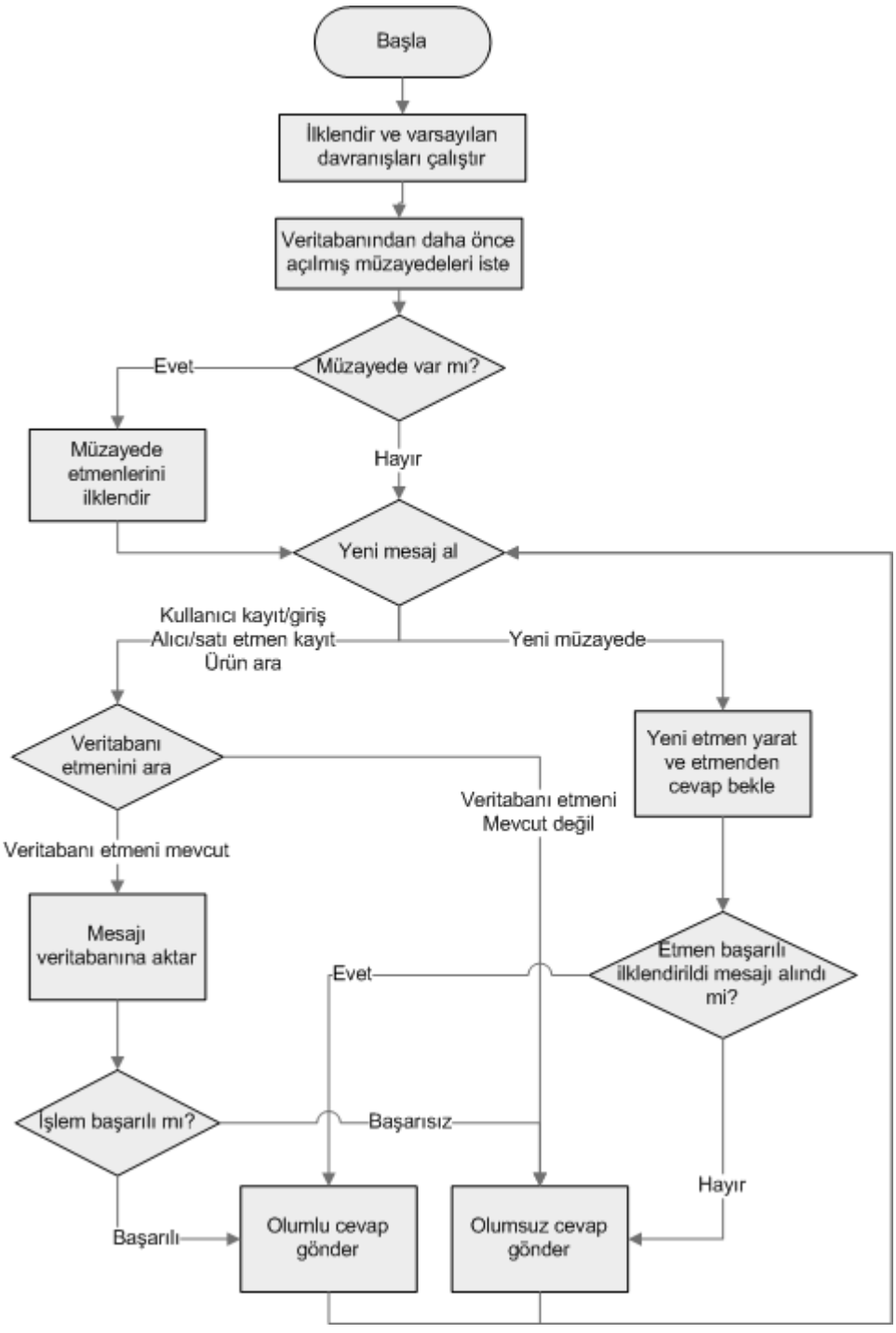
Müzayede evi etmeninin müşteri uygulamasının etmenlerinden gelen mesajları dinlemesi ve işlemesi için çeşitli davranışları mevcuttur.

- HandleRequestBehaviour: CyclicBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayede evi etmenine gelen istek (request) mesajlarını dinleyen davranıştır. Yeni bir mesaj alındığında mesajın içeriği kontrol

edilerek, mesajın tipine göre HandleSearchItemBehaviour, HandleRegisterBuyerAgentBehaviour veya HandleRegisterDealerAgentBehaviour davranışlarından birisi çalıştırılır. Yeni davranış çalıştırıldıktan sonra, istek mesajları dinlenmeye devam edilir.

- HandleProposeBehaviour: CyclicBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayede evi etmenine gelen öneri (propose) mesajlarını dinleyen davranıştır. Yeni bir mesaj alındığında mesajın içeriği kontrol edilir ve mesajın tipine göre HandleCreateNewAuctionBehaviour, HandleSignInUserBehaviour veya HandleSignUpBehaviour davranışlarından birisi çalıştırılır. Yeni davranış çalıştırıldıktan sonra, öneri mesajları dinlenmeye devam edilir.
- HandleSearchItemBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müşterinin alıcı etmeni tarafından ürün arama isteği gönderildiğinde çalıştırılır. Ürün arama isteği veritabanı etmenine yönlendirilir ve alınan cevap müşterinin alıcı etmenine gönderilir.
- HandleRegisterBuyerAgentBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müşteri uygulamasında oluşturulan yeni bir alıcı etmenin bilgilerinin, veritabanına kaydedilmesi sağlar.
- HandleRegisterDealerAgentBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müşteri uygulaması ilk açıldığında oluşturulan satıcı etmeninin bilgilerinin, veritabanına kaydedilmesini sağlar.
- HandleCreateNewAuctionBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müşterinin satıcı etmeni yeni bir müzayede yaratma isteği gönderdiğinde çalıştırılır. Yeni bir müzayede yaratmak için bir müzayede etmeni oluşturulur ve müzayede etmeninden başarılı bir şekilde ilklendirildiği mesajı alınana kadar beklenir. Eğer müzayede etmeninden cevap alınamazsa veya başarısız ilklendirme cevabı alınırsa, satıcı etmene olumsuz sonuç gönderilerek; başarılı ilklendirme cevabı alınırsa, satıcı etmene olumlu sonuç gönderilerek davranış sonlandırılır.
- HandleSignUpUserBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Bir müşterinin yetkilendirme etmeninden kayıt olma mesajı

alındığında çalıştırılır ve veritabanı etmeni ile haberleşerek kullanıcının sisteme kayıt olmasını sağlar.



Şekil 5.9 : Müzayede Evi Etmeninin Akış Diyagramı.

- HandleSignInUserBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Bir müşterinin yetkilendirme etmeninden sisteme giriş yapma isteği alındığında çalıştırılır. Veritabanına bağlanılarak kullanıcının

kayıtlı alıcı ve satıcı etmen bilgileri alınır ve bu etmen bilgileri, etmenlerin tekrar oluşturulması için yetkilendirme etmenine gönderilir.

5.4.1.4 Mesajlar

Müzayede evi etmeni; mesaj içeriği için AuctionOntology, DatabaseOntology, AuhorizationOntology ontolojilerini ve SL dilini kullanır. Müzayede evi etmeni tarafından gönderilen mesajların conversation-id parametresi, mesajın tipini belirten bir karakter katarına evrensel benzersiz bir karakter katarı eklenerek belirlenir. Müzayede evi etmeni tarafından gönderilen mesajlar ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği, aşağıda belirtilmiştir.

- GetAuctions: Müzayede evi etmeninin, veritabanı etmenine devam eden müzayedelerin listesini almak için gönderdiği mesajdır. Müzayede evi etmeni ilk açıldığında bu mesajı gönderir ve cevap olarak aldığı GetAuctionsReply mesajına eklenen her müzayede için birer müzayede etmeni açar.

  sender : Müzayede evi etmeni

  receiver : Veritabanı etmeni

  conversationId : “get-auctions-”

  performative : REQUEST

  language : SLCodec

  ontology : DatabaseOntology

  content : Müzayede etmeninin adı ve müzayede durumu

- SearchAuction: Müzayede evi etmeninin, alıcı etmenden SearchItem mesajı aldığında veritabanı etmenine gönderdiği ürün arama mesajıdır. Cevap olarak alınan SearchAuctionReply mesajının içeriği kullanılarak alıcı etmene SearchItemResult mesajı gönderilir.

  sender : Müzayede evi etmeni

  receiver : Veritabanı etmeni

  conversationId : “search-auction-”

  performative : REQUEST

  language : SLCodec

ontology : DatabaseOntology

content : Aranan ürünün ismi

- SearchItemResult: Alıcı etmenin ürün aramak için gönderdiği SearchItem mesajına cevap olarak gönderilir. Veritabanı etmeninden alınan ürün listesini alıcı etmene gönderir.

  sender : Müzayede evi etmeni

  receiver : Alıcı etmen

  conversationId : Cevap verilen mesajın diyalog belirteci

  performative : CONFIRM ya da REFUSE

  language : SLCodec

  ontology : AuctionOntology

  content : Auction nesnesi listesi

- CreateNewAuctionReply: Müzayede evi etmeni, yeni müzayedeyi başarılı bir şekilde yarattıktan ve yeni müzayede etmeninden, başladığına dair StartAuction mesajını aldıktan sonra; işlemin sonucuna göre satıcı etmene işlem sonucunu bu mesaj ile gönderir.

  sender : Müzayede evi etmeni

  receiver : Satıcı etmen

  conversationId : Cevap verilen mesajın diyalog belirteci

  performative : ACCEPT_PROPOSAL ya da REJECT_PROPOSAL

  language : SLCodec

  ontology : AuctionOntology

  content : Auction nesnesi

- SignInUser: Yetkilendirme etmeninin sisteme girmek için gönderdiği SignInUser mesajı alındığında, bu mesaj doğrudan veritabanı etmenine aktarılır.

  sender : Müzayede evi etmeni

  receiver : Veritabanı etmeni

conversationId : “login-user-”

performative : REQUEST

language : SLCodec

ontology : AuthorizationOntology

content : UserData nesnesi

- SignInUserReply: Veritabanı etmeninden SignInUserReply mesajı alındığında, bu mesaj doğrudan yetkilendirme etmenine aktarılır. Verilen cevaptaki diyalog belirteci, yetkilendirme etmeninden alınan ilk SignInUser mesajının diyalog belirtecidir.

  sender : Müzayede evi etmeni

  receiver : Yetkilendirme etmeni

  conversationId : Cevap verilen mesajın diyalog belirteci

  performative : CONFIRM ya da REFUSE

  language : SLCodec

  ontology : AuthorizationOntology

  content : UserData, DealerAgentData, BuyerAgentData listesi

- SignUpUser: Yetkilendirme etmeninin sisteme kayıt olmak için gönderdiği SignUpUser mesajı alındığında, bu mesaj doğrudan veritabanı etmenine aktarılır.

  sender : Müzayede evi etmeni

  receiver : Veritabanı etmeni

  conversationId : “register-user-”

  performative : REQUEST

  language : SLCodec

  ontology : AuthorizationOntology

  content : UserData, CreditCard ve BankAccount nesneleri

- SignUpUserReply: Veritabanı etmeninden SignUpUserReply mesajı alındığında, bu mesaj doğrudan yetkilendirme etmenine aktarılır. Verilen cevaptaki diyalog belirteci, yetkilendirme etmeninden alınan ilk SignUpUser mesajının diyalog belirtecidir.

  sender : Müzayede evi etmeni

  receiver : Yetkilendirme etmeni

  conversationId : Cevap verilen mesajın diyalog belirteci

  performative : CONFIRM ya da REFUSE

  language : SLCodec

  ontology : AuthorizationOntology

  content : UserData nesnesi

- RegisterBuyerAgent: Müşterinin alıcı etmeninden, sisteme kayıt olmak ve verilerini kaydetmek için gönderdiği RegisterBuyerAgent mesajı alındığında, bu mesaj doğrudan veritabanı etmenine aktarılır.

  sender : Müzayede evi etmeni

  receiver : Veritabanı etmeni

  conversationId : “insert-buyer-agent-”

  performative : REQUEST

  language : SLCodec

  ontology : AuthorizationOntology

  content : BuyerAgentData nesnesi

- RegisterBuyerAgentReply: Veritabanı etmeninden RegisterBuyerAgentReply mesajı alındığında, bu mesaj doğrudan müşterinin alıcı etmenine aktarılır. Verilen cevaptaki diyalog belirteci, alıcı etmenden alınan ilk RegisterBuyerAgent mesajının diyalog belirtecidir.

  sender : Müzayede evi etmeni

  receiver : Alıcı etmen

  conversationId : Cevap verilen mesajın diyalog belirteci

performative : CONFIRM ya da REFUSE

language : SLCodec

ontology : AuthorizationOntology

content : BuyerAgentData nesnesi

- RegisterDealerAgent: Müşterinin satıcı etmeninden, sisteme kayıt olmak ve verilerini kaydetmek için gönderdiği RegisterDealerAgent mesajı alındığında, bu mesaj doğrudan veritabanı etmenine aktarılır.

  sender : Müzayede evi etmeni

  receiver : Veritabanı etmeni

  conversationId : "insert-dealer-agent-"

  performative : REQUEST

  language : SLCodec

  ontology : AuthorizationOntology

  content : DealerAgentData nesnesi

- RegisterDealerAgentReply: RegisterDealerAgent mesajı gönderilen veritabanı etmeninden, RegisterDealerAgentReply mesajı alındığında, bu mesaj doğrudan müşterinin satıcı etmenine aktarılır. Verilen cevaptaki diyalog belirteci, satıcı etmenden alınan ilk RegisterDealerAgent mesajının diyalog belirtecidir.

  sender : Müzayede evi etmeni

  receiver : Satıcı etmen

  conversationId : Cevap verilen mesajın diyalog belirteci

  performative : CONFIRM ya da REFUSE

  language : SLCodec

  ontology : AuthorizationOntology

  content : DealerAgentData nesnesi

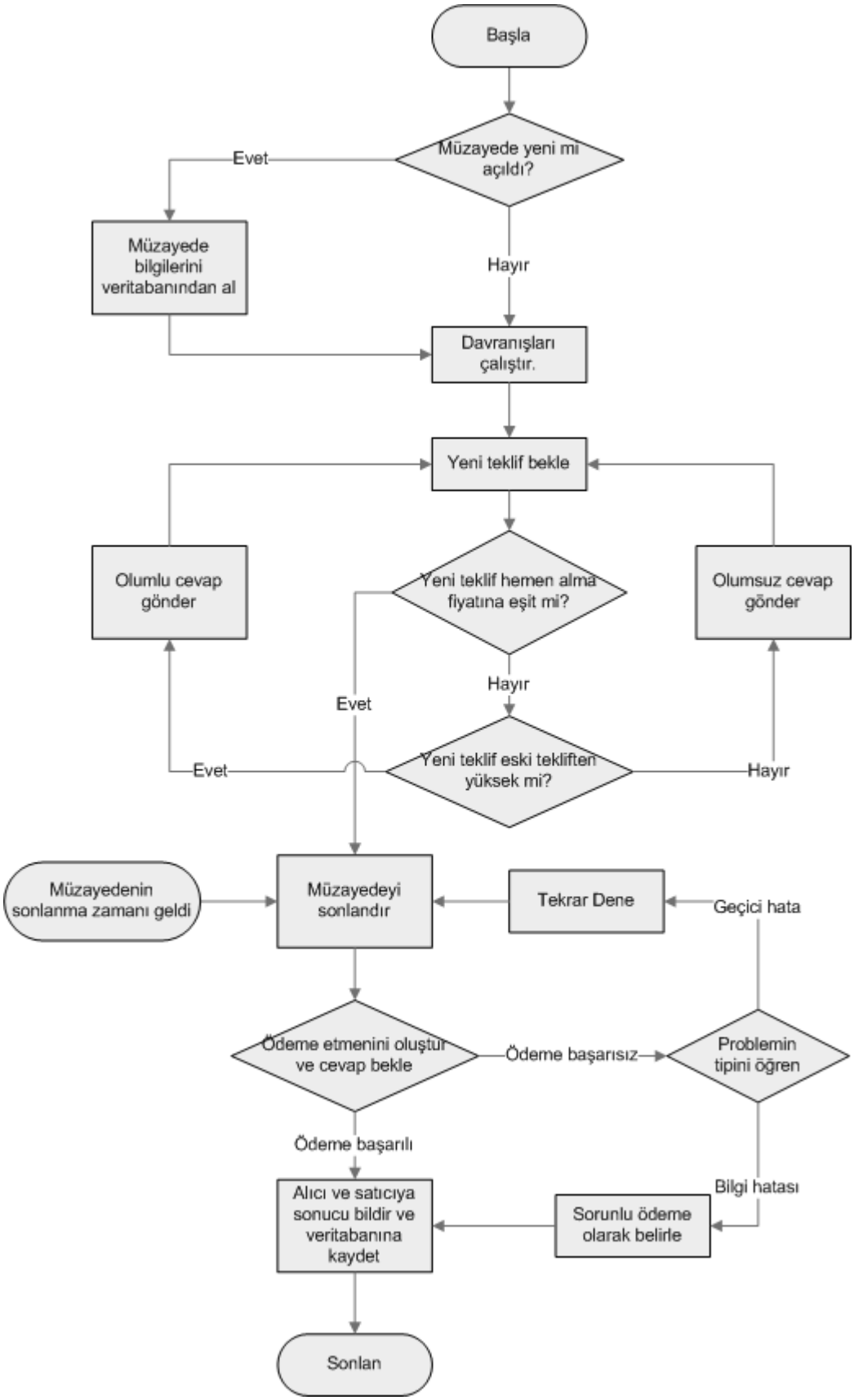
5.4.2 Müzayede etmeni

Müzayede etmeni, bir müzayedenin yönetiminden sorumludur. Bir müzayede evi uygulamasında birden fazla müzayede etmeni bulunabilir. Müzayede etmenleri, müzayede evi etmenleri tarafından müşterinin satıcı etmeninden yeni bir müzayede yaratma isteği alındığında yaratılırlar. Müzayede etmenlerinin görevleri şu şekilde sıralanır:

- Müzayedeye katılan alıcı etmenlerden teklif almak; bu teklif eğer o andaki en yüksek tekliften daha yüksekse teklifi kabul etmek, aksi halde teklifi reddetmek.
- Ürüne yapılan yeni teklif sonucunda, ürüne teklif edilen en yüksek fiyat değiştiğinde bu durumu müşterinin satıcı etmenine bildirmek.
- Ürüne yapılan yeni teklif sonucunda değişen müzayede bilgilerini veritabanına kaydetmesi için veritabanı etmeni ile haberleşmek.
- Müzayedenin süresi tamamlandığında veya bir alıcıdan ürünü hemen alma isteği geldiğinde müzayedeyi kapatmak ve sonucu alıcı ve satıcı etmenlere bildirmek.

5.4.2.1 İlklendirme

Müzayede etmeni çalışmaya başladığında ilk olarak kendisine parametre olarak aktarılan müzayede bilgilerini okur. Bu bilgilere göre içyapılarındaki verileri günceller ve çalışmaya başladığını StartAuction mesajı ile müzayede evine bildirir. Ardından kendisine gelen PROPOSE mesajlarını dinlemek için HandleProposeBehaviour davranışını, REQUEST mesajlarını dinlemek için HandleRequestBehaviour davranışını ve INFORM mesajlarını dinlemek için HandleInformBehaviour davranışını çalıştırır. Müzayedenin bitiş zamanında otomatik olarak uyanıp müzayedeyi sonlandırması için CloseAuctionWakerBehaviour davranışını da çalıştırdıktan sonra, alıcı etmenlerden gelecek teklifleri beklemeye başlar. Müzayede etmeninin akış diyagramı Şekil 5.10'da verilmiştir.



Şekil 5.10 : Müzayede Etmeninin Akış Diyagramı.

5.4.2.2 Çalışma

Herhangi bir müşterinin alıcı etmeninden ProposeNewBid mesajı ile yeni teklif alındığında, yeni teklif o anki en yüksek teklifle karşılaştırılır ve bu teklif en yüksek teklifse kabul edilir. Eğer yeni teklif ilk teklifse müzayedenin başlangıç fiyatından büyükse teklif kabul edilir. Sonuç da ProposeNewBidReply mesajı ile alıcı etmene bildirilir. Satıcı etmene bilgilendirme amaçlı InformAuctionUpdate mesajı gönderilir ve veritabanı etmenine değişikliği kaydetmesi için SaveAuctionInformation mesajı gönderilir. Eğer müzayede için bir hemen alma fiyatı belirlendiyse ve alıcı etmenin verdiği teklif hemen alma fiyatına eşitse müzayede kapatılır ve ürün müşteri alıcı etmenine satılmış olur. Eğer hiç bir etmen hemen alma işlemi yapmazsa, müzayedenin süresi dolduğunda müzayede kapatılır ve o an en yüksek fiyatı vermiş olan etmene ürün satılır.

Müzayede sonlandığında bir ödeme etmeni yaratılır ve satıcı etmenin sağladığı hesap numarası bilgisi ile alıcı etmenin sağladığı kredi kartı bilgisi kullanılarak ödeme işlemi bu etmen üzerinden yapılır. Ödeme etmeni, ödeme işlemi tamamlandıktan sonra müzayede etmenine InformPaymentResult mesajı gönderir ve çalışmasını sonlandırır. Ödeme işlemi tamamlandıktan sonra veritabanı etmenine gönderilen SaveAuctionInformation mesajı ile müzayede bilgileri kaydedilir, satıcı ve alıcı etmenlere InformAuctionResult mesajı ile sonuç bildirilir ve etmen çalışmasını sonlandırır.

5.4.2.3 Davranışlar

Müzayede etmeninin müzayedeyi yönetebilmesi için gerçeklenmiş çeşitli davranışları mevcuttur.

- StartAuctionBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Veritabanı etmenine müzayede bilgilerini kaydetmesi için mesaj gönderir ve aldığı cevaba göre, müzayede evi etmenine başarılı veya başarısız bir şekilde ilklendirildiğini bildirir.
- HandleProposeBehaviour: CyclicBehavior sınıfının genişletilmesiyle gerçeklenmiştir Etmene gelen PROPOSE mesajlarını işleyerek gerekli davranışı çalıştırır.

- HandleProposeNewBidBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Yeni bir teklif geldiğinde bu teklifi işler ve teklif yapan alıcı etmene ret veya onay mesajı gönderir. Yeni teklifi ve teklif veren etmen bilgilerini veritabanında güncellemesi için veritabanı etmeni ile haberleşir.
- CloseAuctionWakerBehaviour: WakerBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayedenin belirlenen sonlanma süresi geldiğinde ödeme etmenini çalıştırır.
- HandleInformBehaviour: CyclicBehavior sınıfının genişletilmesiyle gerçeklenmiştir Etmene gelen INFORM mesajlarını alır. Etmene gelen INFORM mesajı sadece InformPaymentResult olduğu için mesaj bu davranış içerisinde işlenir. Eğer ödeme işlemi başarılı ise CloseAuctionBehaviour davranışını çalıştırır.
- CloseAuctionBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayedenin kapatılmasını sağlar. Bu davranışın görevi, öncelikle SaveAuctionInformationBehaviour davranışını, bu davranış tamamlandıktan sonra ise SendInformAuctionResultBehaviour davranışını çalıştırmaktır.
- SaveAuctionInformationBehaviour: Behaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayede bilgilerinin kaydedilmesi için veritabanı etmeni ile UpdateAuction mesajını göndererek haberleşir.
- SendInformAuctionResultBehaviour: OneShotBehaviour sınıfının genişletilmesiyle gerçeklenmiştir. Müzayede sonlandıktan sonra satıcı ve alıcı etmenlere müzayede sonucunu içern InformAuctionResult mesajını gönderir.

5.4.2.4 Mesajlar

Müzayede etmeni; mesaj içeriği için AuctionOntology, DatabaseOntology ontolojilerini ve SL dilini kullanır. Müzayede etmeni tarafından gönderilen mesajların conversation-id parametresi, mesajın tipini belirten bir karakter katarına evrensel benzersiz bir karakter katarı eklenerek belirlenir. Müzayede etmeni tarafından gönderilen mesajlar ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği, aşağıda belirtilmiştir.

- StartAuction: Etmenin ilklendirme işleminin sonucunda, müzayede evi etmenine gönderdiği mesajdır.

  sender : Müzayede etmeni

  receiver : Müzayede evi etmeni

  conversationId : Etmene parametre olarak aktarılan diyalog belirteci

  performative : INFORM

  language : SLCodec

  ontology : AuctionOntology

  content : Auction nesnesi

- InsertAuction: Veritabanı etmenine müzayede ve müzayede etmeni bilgilerini veritabanına eklemesi için gönderilen mesajdır.

  sender : Müzayede etmeni

  receiver : Veritabanı etmeni

  conversationId : “insert-auction-”

  performative : REQUEST

  language : SLCodec

  ontology : DatabaseOntology

  content : Auction ve AuctionAgentData nesneleri

- UpdateAuction: Veritabanı etmenine müzayede bilgilerini güncellemesi için gönderilen mesajdır.

  sender : Müzayede etmeni

  receiver : Veritabanı etmeni

  conversationId : “update-auction-”

  performative : REQUEST

  language : SLCodec

  ontology : DatabaseOntology

  content : Auction nesnesi

- ProposeNewBidReply: Alıcı tarafından yapılan bir teklifin sonucunu bildirir.

  sender : Müzayede etmeni

  receiver : Alıcı etmen

  conversationId : Cevap verilen mesajın diyalog belirteci

  performative : ACCEPT_PROPOSAL ya da REJECT_PROPOSAL

  language : SLCodec

  ontology : DatabaseOntology

  content : Auction nesnesi

- InformAuctionUpdate: Yeni teklif yapıldığında güncellenen müzayede bilgilerini satıcı etmene bildirmek için gönderilen mesajdır.

  sender : Müzayede etmeni

  receiver : Satıcı etmen

  conversationId : Zorunlu bir alan değildir.

  performative : INFORM

  language : SLCodec

  ontology : AuctionOntology

  content : Auction nesnesi

- InformAuctionResult: Tamamlanan bir müzayedenin bilgilerini satıcı ve alıcı etmenlere bildirmek için gönderilen mesajdır.

  sender : Müzayede etmeni

  receiver : Satıcı etmen ve alıcı etmen

  conversationId : Zorunlu bir alan değildir.

  performative : INFORM

  language : SLCodec

  ontology : AuctionOntology

  content : Auction nesnesi

5.4.3 Ödeme etmeni

Ödeme etmeninin görevi, alıcı etmenin kredi kartı bilgileri ile satıcı etmenin banka hesap bilgilerini kullanarak ödeme işlemini gerçekleştirmektir. Ödeme işleminin gerçeklenmesi sistem tasarımının kapsamı dışında bırakıldığı için etmen herhangi bir işlem yapmadan ödeme yapıldı bilgisini verir.

5.4.3.1 İlklendirme ve çalışma

Ödeme etmenine parametre olarak, müzayede bilgilerini içeren Auction nesnesi verilir. Ödeme etmeni kendisine verilen bu parametreleri okuduktan sonra SendGetPaymentInformationBehaviour davranışını çalıştırarak veritabanı etmeninden alıcı etmenin kredi kartı bilgilerini ve satıcı etmenin banka hesap bilgilerini alır. Bu bilgiler veritabanı etmeninden başarılı bir şekilde alınırsa SendInformPaymentResultBehaviour davranışı olumlu bir sonuç ile başarısız bir şekilde alınırsa olumsuz bir sonuç ile çağrılır. Bu davranış ile InformPaymentResult mesajı müzayede etmenine gönderilir ve etmenin çalışması sonlanır.

5.4.3.2 Davranışlar

Ödeme etmeninin, ödeme işlemini gerçekleştirmek için sahip olduğu davranışlar aşağıdaki gibi sıralanır:

- SendGetPaymentInformationBehaviour: Behaviour davranışının genişletilmesi ile gerçeklenmiştir. Veritabanı etmeninden ödeme bilgileri GetPaymentInformation mesajı ile istenir ve bilgiler alındıktan sonra SendInformPaymentResultBehaviour davranışı çalıştırılır.
- SendInformPaymentResultBehaviour: OneShotBehaviour davranışının genişletilmesi ile gerçeklenmiştir. Müzayede etmenine ödeme sonucunu bildirmek için InformPaymentResult mesajını gönderir.

5.4.3.3 Mesajlar

Ödeme etmeni; mesaj içeriği için DatabaseOntology ve AuctionOntology ontolojilerini ve SL dilini kullanır. Ödeme etmeni tarafından gönderilen mesajların conversation-id parametresi, mesajın tipini belirten bir karakter katarına evrensel benzersiz bir karakter katarı eklenerek belirlenir. Ödeme etmeni tarafından

gönderilen mesajlar ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği, aşağıda sıralanmıştır.

- GetPaymentInformation: Veritabanı etmeninden müzayedenin kazanan alıcısının kredi kartı bilgilerini ve satıcısının banka hesap bilgilerini istemek için gönderilir.

  sender : Ödeme etmeni

  receiver : Veritabanı etmeni

  conversationId : "get-payment-information-"

  performative : REQUEST

  language : SLCodec

  ontology : DatabaseOntology

  content : Alıcı etmenin (String) ve satıcı etmenin adı (String)

- InformPaymentResult: Müzayede etmenine ödeme sonucunu bildirmek için gönderilir.

  sender : Ödeme etmeni

  receiver :Müzayede etmeni

  conversationId : "inform-payment-result-"

  performative : INFORM

  language : SLCodec

  ontology : AuctionOntology

  content : Ödeme sonucu (Int)

5.5 Veritabanı Uygulaması

Sistemde yaratılan etmenleri ve müzayedeleri kayıt altına almak için bir veritabanı uygulaması bulunur. Tüm müzayede sisteminde bir adet veritabanı uygulaması bulunur. Bu uygulama bütün müzayede evi uygulamalarına hizmet verir. Bu uygulamalarda çalışan müzayede evi, müzayede ve ödeme etmenleri veritabanı etmeni ile haberleşebilirler. Veritabanı uygulaması bir adet veritabanı uygulaması,

bir adet veritabanı etmeni ve bir adet MySQL veritabanından oluşur. Veritabanı uygulamasının görevi, tercihler dosyasını okuyarak gerekli parametrelerle veritabanı etmenini yaratmaktır. Veritabanı üzerinde sadece veritabanı etmeni işlem yapar.

5.5.1 Tercihler

Veritabanı uygulamasının çalışması için gerekli bilgiler bir tercihler dosyasında bulunur. Bu bilgilerin bazıları veritabanı uygulaması tarafından veritabanı etmenini başlatmak için, bazıları ise veritabanı etmeni tarafından veritabanına bağlanmak için kullanılır. Tercihler dosyasında bulunan bilgiler aşağıdaki aşağıda sıralanmıştır.

- Ana konak adresi: JADE sisteminin ana merkezinin adresidir. Tüm uygulamalar bu adrese bağlanırlar.
- Etmen ismi: Veritabanı etmeninin, sisteme bağlanırken kullanacağı addır.
- Veritabanı adresi: Veritabanı sisteminin bulunduğu konağın adresidir.
- Veritabanı ismi: Veritabanı etmeninin bağlanacağı veritabanının ismidir.
- Kullanıcı adı: Veritabanı etmeninin veritabanına bağlanırken kullanacağı kullanıcı adıdır.
- Şifre: Veritabanı etmeninin veritabanına bağlanırken kullanacağı şifredir.

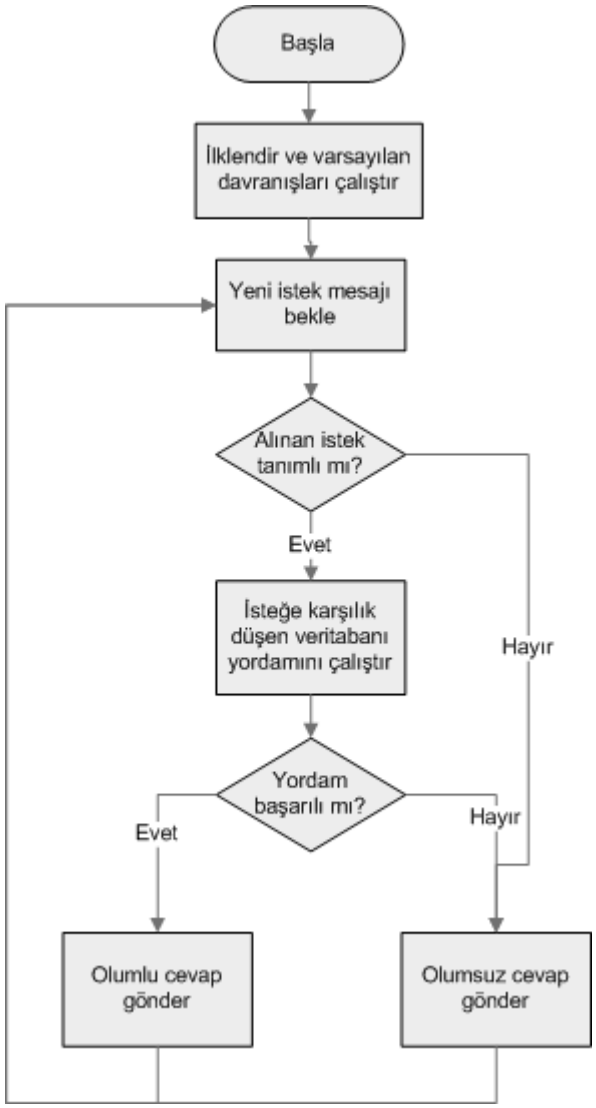
5.5.2 Veritabanı etmeni

Veritabanı etmeni reaktif bir etmendir. Görevi, müzayede evinde çalışan etmenlerden gelen istekleri gerçekleştirmek ve işlem sonucuna göre cevap vermektir. Bu nedenle, başlatıldıktan sonra dizin yöneticisine kayıt olur ve HandleRequestBehaviour davranışını çalıştırarak istek mesajı beklemeye başlar. Veritabanı etmeninin akış diyagramı Şekil 5.11'de verilmiştir.

5.5.2.1 Davranışlar

Veritabanı etmeninin diğer etmenlerden gelen istekleri gerçekleştirmek için tanımladığı birçok davranış vardır.

- HandleRequestBehaviour: CyclicBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Alınan REQUEST mesajının içeriği kontrol edilerek ilgili işlemi gerçekleştirecek olan davranış çalıştırılır.



Şekil 5.11 : Veritabanı Etmeninin Akış Diyagramı.

- HandleInsertAuctionBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müzayede etmeninin yeni müzayede eklemek için gönderdiği InsertAuction mesajını işler ve işlemin sonucunu içeren InsertAuctionReply mesajını gönderir.
- HandleUpdateAuctionBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müzayede etmeninin var olan bir müzayedeyi güncellemek için gönderdiği UpdateAuction mesajını işler ve işlemin sonucunu içeren UpdateAuctionReply mesajını gönderir.
- HandleSignInUserBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müşterinin yetkilendirme etmeninin müzayede evi etmeni aracılığıyla, sisteme giriş yapmak ve henüz işi

sonlanmamış olan etmenlerinin bilgilerini veritabanından almak için gönderdiği SignInUser mesajını işler ve işlemin sonucunu içeren SignInUserReply mesajını gönderir.

- HandleSignUpUserBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müşterinin yetkilendirme etmeninin müzayede evi etmeni aracılığıyla, sisteme kayıt olmak için gönderdiği SignUpUser mesajını işler ve işlemin sonucunu içeren SignUpUserReply mesajını gönderir.
- HandleGetAuctionsBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müzayede evi etmeninin ilk açılışta, henüz kapanmamış müzayedelerini veritabanından almak için gönderdiği GetAuctions mesajını işler ve işlemin sonucunu gönderir.
- HandleSearchAuctionBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müşterinin alıcı etmeninin müzayede evi etmeni aracılığıyla, müzayededе satılan ürünler arasında aramak için gönderdiği SearchAuction mesajını işler ve işlemin sonucunu içeren SearchAuctionReply mesajını gönderir.
- HandleRegisterBuyerAgentBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müşteri uygulamasında yeni yaratılan bir alıcı etmenin, bilgilerini veritabanına kaydetmek için müzayede evi etmeni aracılığıyla gönderdiği RegisterBuyerAgent mesajını işler ve işlemin sonucunu içeren RegisterBuyerAgentReply mesajını gönderir.
- HandleRegisterDealerAgentBehaviour: OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Müşteri uygulamasında yeni yaratılan bir satıcı etmenin bilgilerini veritabanına kaydetmek için müzayede evi etmeni aracılığıyla gönderdiği RegisterDealerAgent mesajını işler ve işlemin sonucunu içeren RegisterDealerAgentReply mesajını gönderir.
- HandleGetPaymentInformationBehaviour. OneShotBehaviour davranışının genişletilmesiyle gerçeklenmiştir. Ödeme etmeninin, alıcı etmenin kredi kartı bilgileriyle satıcı etmenin hesap bumarası bilgilerini almak için gönderdiği GetPaymentInformation mesajını işler ve işlemin sonucunu içeren GetPaymentInformationReply mesajını gönderir.

5.5.2.2 Mesajlar

Müzayede etmeni; mesaj içeriği için DatabaseOntology, AuthorizationOntology ontolojilerini ve SL dilini kullanır. Veritabanı etmeni tarafından gönderilen mesajların conversation-id parametresi, mesajın tipini belirten bir karakter katarına evrensel benzersiz bir karakter katarı eklenerek belirlenir. Veritabanı etmeni tarafından gönderilen mesajlar ve her mesaja karşılık düşen ACLMessage nesnesinin içeriği, aşağıda belirtilmiştir. Tüm mesajlar için ortak olan alanlar şunlardır.

sender :Veritabanı etmeni

conversationId : Cevap verilen mesajın diyalog belirteci

performative : REFUSE ya da CONFIRM

language : SLCodec

- InsertAuctionReply: Yeni müzayede ekleme işleminin sonucunu bildirmek için gönderilir.

  receiver : Müzayede etmeni

  ontology : DatabaseOntology

  content : Auction ve AuctionAgentData nesneleri

- SignInUserReply: Yetkilendirme etmeninin müzayede evi etmeni aracılığıyla yaptığı sisteme giriş isteğinin sonucunu ve henüz işi bitmemiş olan satıcı ve alıcı etmenlerin bilgilerini bildirmek için gönderilir.

  receiver : Müzayede evi etmeni

  ontology : AuthorizationOntology

  content : UserData, DealerAgentData ve BuyerAgentData listesi

- SignUpUserReply: Yetkilendirme etmeninin müzayede evi etmeni aracılığıyla yaptığı sisteme kayıt olma isteğinin sonucunu bildirmek için gönderilir.

  receiver : Müzayede evi etmeni

  ontology : AuthorizationOntology

  content : UserData nesnesi

- GetAuctionsReply: Müzayede evi etmeninin kendisine bağlı müzayedeleri arama işleminin sonucunu bildirmek için gönderilir.

  receiver : Müzayede evi etmeni

  ontology : DatabaseOntology

  content : Auction listesi

- SearchAuctionReply: Müzayede evi etmeninin, müşterisi için ürünün adına göre müzayede arama isteğinin sonucunu bildirmek ve bulunan müzayede listesini göndermek için gönderilir.

  receiver : Müzayede evi etmeni

  ontology : DatabaseOntology

  content : Auction listesi

- RegisterBuyerAgentReply: Alıcı etmenin müzayede evi etmeni aracılığıyla yaptığı verilerini kaydetme isteğinin sonucunu bildirmek için gönderilir.

  receiver : Müzayede evi etmeni

  ontology : AuthorizationOntology

  content : BuyerAgentData nesnesi

- RegisterDealerAgentReply: Satıcı etmenin müzayede evi etmeni aracılığıyla yaptığı verilerini kaydetme isteğinin sonucunu bildirmek için gönderilir.

  receiver : Müzayede evi etmeni

  ontology : AuthorizationOntology

  content : DealerAgentData nesnesi

- GetPaymentInformationReply: Ödeme etmeninin, alıcı ve satıcı etmenlerin ödeme bilgilerini elde etme isteğinin sonucunu bildirmek için gönderilir.

  receiver : Ödeme etmeni

  ontology : DatabaseOntology

  content : BankAccount ve CreditCard nesneleri

6. DENEYLER

Tasarlanan sistemin çalışmasını incelemek için çeşitli deneyler gerçeklenmiştir. Bu deneylerin amacı müzayedelere teklif verme sürecinde, alıcı ve satıcı etmen sayılarının, alıcı etmen stratejilerinin ve kullanıcı tercihlerinin müzayedeye verilen teklif miktarlarını ve müzayedenin hemen alma fiyatına erişme süresini nasıl etkilediğini incelemektir. Bu amaçla 6 adet deney gerçeklenmiştir. Bu deneylerin 3 tanesinde aynı strateji altında farklı kullanıcı tercihlerine sahip olan alıcı etmenlerin müzayedeye katıldığı durum, 1 tanesinde farklı stratejilere sahip olan alıcı etmenlerin müzayedeye katıldığı durum, 2 tanesinde etmenlerin birden fazla müzayedeye katıldıkları durum incelenmiştir.

Deneylerde her alıcı etmen farklı bir müşteri tarafından yaratılmıştır. Açılan her müzayede, farklı bir müzayede evinden açık arttırmaya sunulmuştur. Sistemde 1 adet veritabanı uygulaması bulunmaktadır. Tüm deneylerde müzayedeler belirli bir fiyata erişildiğinde kapatılmaktadır ve alıcı etmenlerin en fazla verebileceği teklif müzayedelerin hemen alma fiyatından fazla tutulmuştur.

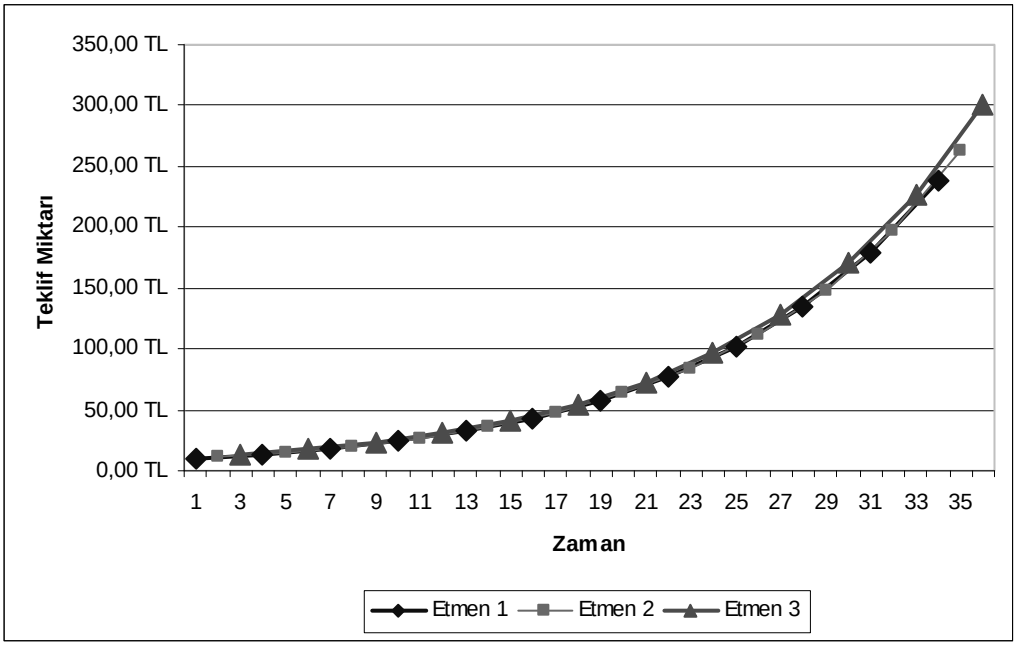
6.1 Deney 1

Deneyin amacı, verilen en yüksek teklife göre teklif verme stratejisinin karakteristik özelliklerini ve aynı strateji altında, kullanıcının teklif arttırma oranı tercihinin etmenin davranışına ve teklif miktarlarına etkisini incelemektir. Deneyde 3 adet alıcı etmen ve 1 adet müzayede etmeni bulunmaktadır. Müzayedede satılan ürünün başlangıç satış fiyatı 10 TL, hemen alma fiyatı 300 TL'dir. Müşterilerin alıcı etmenler için girdikleri tercihler Çizelge 6.1'de sıralanmıştır.

Çizelge 6.1 : Deney 1'deki Müşteri Tercihleri.

	Etmen 1	Etmen 2	Etmen 3
Strateji	En Yüksek Teklif	En Yüksek Teklif	En Yüksek Teklif
Teklif Arttırma Oranı	5	10	20
Yeniden Kontrol Etme Süresi	1	1	1

Şekil 6.1'de deney 1 için müzayedenin teklif miktarının zamana göre değişmesi görülmektedir. Müzayedenin başlangıcında en yüksek teklif miktarı düşük olduğu için alıcı etmenler tarafından verilen teklifler de düşük düzeyde olmuştur. Müzayede ilerledikçe en yüksek teklif miktarı artmış, böylece etmenlerin verdikleri teklifler de bu oranda artmıştır. Sonuç olarak, tüm etmenler en yüksek teklife göre teklif verme stratejisi uyguladıklarında, verilen tekliflerin artan bir hızla artttığı gözlemlenmiştir. En yüksek teklif arttırma oranına sahip olan etmen, 3 numaralı etmendir. Şekil 6.1'de de görüleceği gibi, en yüksek teklif artışlarını 3 numaralı etmen yapmıştır ve son olarak da büyük bir teklif artışı yaparak ürünü hemen alma fiyatı ile satın almıştır.



Şekil 6.1 : Deney 1 Sonuçları.

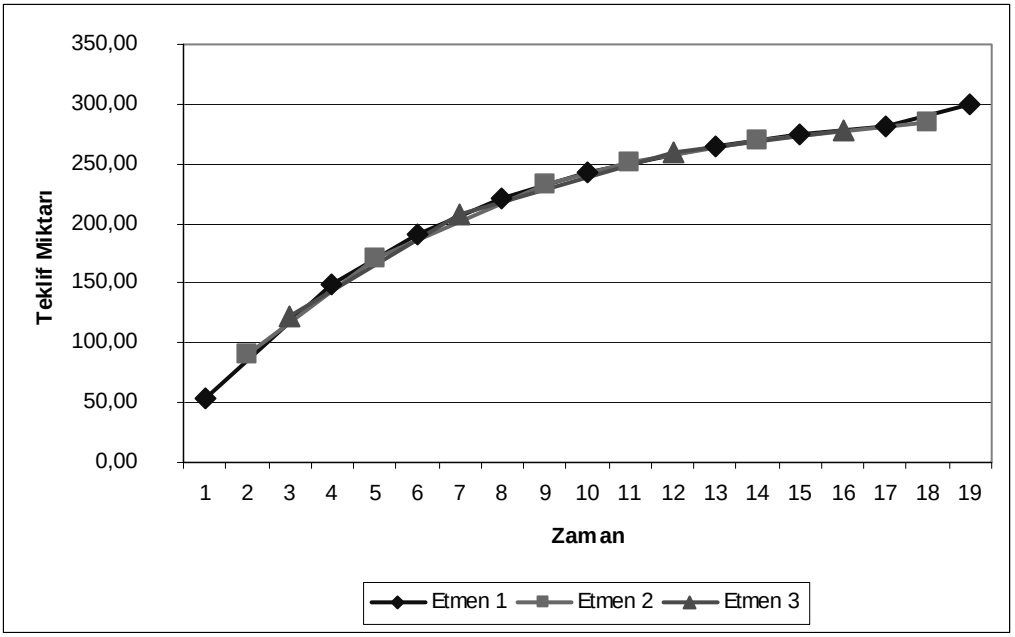
6.2 Deney 2

Deneyin amacı, hemen alma fiyatına göre teklif belirleme stratejisinin karakteristik özelliklerini ve aynı strateji altında, kullanıcının müzayedeyi yeniden kontrol etme süresi tercihinin, etmenin davranışına ve teklif miktarlarına olan etkisini incelemektir. Deneyde 3 adet alıcı etmen ve 1 adet müzayede etmeni bulunmaktadır. Müzayedede satılan ürünün başlangıç satış fiyatı 10 TL, hemen alma fiyatı 300 TL'dir. Müşterilerin alıcı etmenler için girdikleri tercihler Çizelge 6.2'de sıralanmıştır.

Çizelge 6.2 : Deney 2'deki Müşteri Tercihleri.

	Etmen 1	Etmen 2	Etmen 3
Strateji	Hemen Alma Fiyatı	Hemen Alma Fiyatı	Hemen Alma Fiyatı
Teklif Arttırma Oranı	15	15	15
Yeniden Kontrol Etme Süresi	1	2	3

Şekil 6.2'de deney 2 için müzayedenin teklif miktarının zamana göre değişmesi görülmektedir. Müzayedenin başlangıcında, hemen alma fiyatı ile en yüksek teklif arasındaki fark yüksek olduğu için etmenler teklif vermeye daha yüksek fiyattan başlamışlardır. Müzayede ilerledikçe hemen alma fiyatı ile en yüksek teklif arasındaki fark azaldığı için tekliflerin artma oranı da azalmıştır. Sonuç olarak tüm etmenlerin hemen alma fiyatına göre teklif verme stratejisi uyguladıkları durumda, müzayedeye verilen tekliflerin azalan bir şekilde arttığı gözlemlenmiştir. Deney sırasında 1 numaralı etmen 9 teklif, 2 numaralı etmen 6 teklif, 3 numaralı etmen 4 teklif yapmıştır. 1 numaralı etmen müzayedenin durumunu daha sık gözlemlemiş ve daha sık teklif vererek müzayedeyi kazanma şansını arttırmıştır; sonuç olarak da müzayedeyi kazanmıştır.



Şekil 6.2 : Deney 2 Sonuçları.

6.3 Deney 3

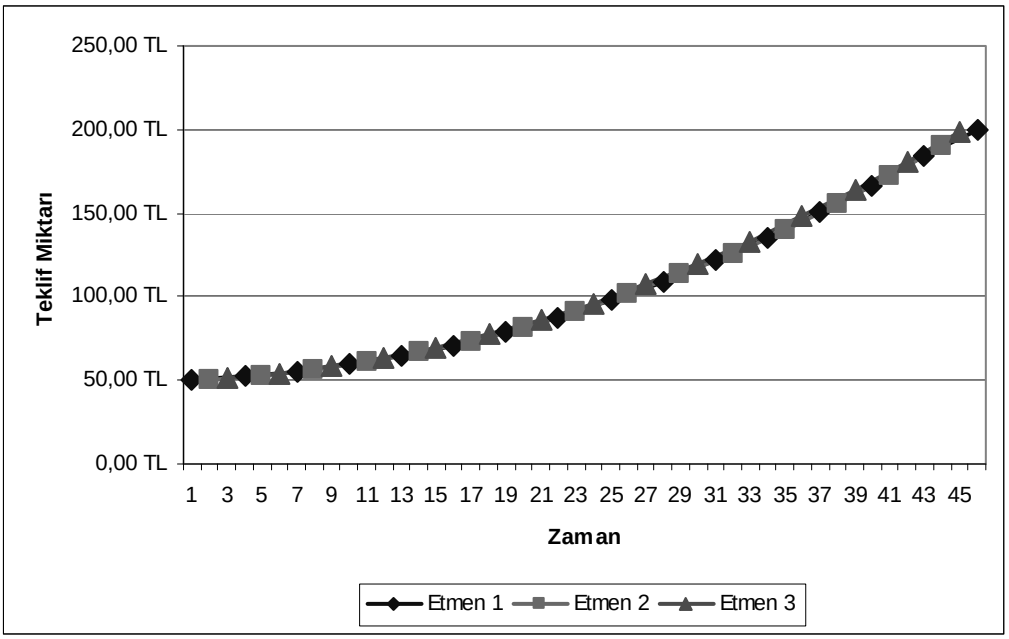
Deneyin amacı, müzayedeye verilen diğer tekliflerin ortalamasına göre teklif verme stratejisinin özelliklerini ve teklif arttırma oranının bu stratejiyi uygulayan etmenin teklifleri üzerindeki etkisini incelemektir. Deneyde 3 adet alıcı etmen ve 1 adet müzayede etmeni bulunmaktadır. Müzayedede satılan ürünün başlangıç satış fiyatı 50 TL, hemen alma fiyatı 200 TL'dir. Bu stratejide, teklif arttırma oranı ürün için belirlenmiş bir fiyat yerine arttırma miktarına uygulandığı için diğer stratejilere göre daha fazla bir arttırma miktarı seçilmesi gerekir. Bu durum göz önünde bulundurularak müşterinin seçtiği tercihler Çizelge 6.3'te sıralanmıştır.

Çizelge 6.3 : Deney 3'teki Müşteri Tercihleri.

	Etmen 1	Etmen 2	Etmen 3
Strateji	Diğer Tekliflerin Ortalaması	Diğer Tekliflerin Ortalaması	Diğer Tekliflerin Ortalaması
Teklif Arttırma Oranı	33	66	100
Yeniden Kontrol Etme Süresi	1	1	1

Şekil 6.3'te deney 3 için müzayedenin teklif miktarının zamana göre değişmesi görülmektedir. Müzayedenin başlangıcında, ilk teklif yapan etmen başlangıç fiyatının küçük bir oranı kadar arttırma yapacağı için, bu arttırmayı göz önünde bulundurarak fiyat arttıracak ikinci etmen de az miktar fiyat arttıracaktır. Bir etmenin iki teklifi arasında, iki etmenin teklifleri olacağı için verilen teklifler zamanla artacaktır. Fakat bu artma miktarı en fazla tekliflerin artma miktarının ortalaması kadar olabileceği için Deney 1'deki en yüksek teklife göre teklif belirleme stratejisiyle karşılaştırıldığında çok daha az keskin bir artma olacaktır.

Deneyde düzenlenen müzayedeyi en az teklif arttırma oranına sahip olan etmen kazanmıştır. Sonuç olarak, diğer tekliflerin ortalamasına göre teklif verme stratejisinde, tüm etmenlerin aynı stratejiyi uyguladığı durumda teklif arttırma oranının en yüksek teklife göre teklif belirleme stratejisiyle karşılaştırıldığında daha önemsiz olduğu gözlemlenmiştir. Bunun nedeni tekliflerdeki artışın çok keskin olmamasıdır. Deney 2, teklifin artış hızı azalan bir müzayedede yeniden kontrol etme süresinin önemini ortaya koymuştur. Bu nedenle, artış hızı az miktarda yükselen diğer tekliflerin ortalamasına göre teklif verme stratejisinde, yeniden kontrol etme süresininin sıklaştırılması, müzayedenin kazanılmasında daha fazla etkilidir.



Şekil 6.3 : Deney 3 Sonuçları.

6.4 Deney 4

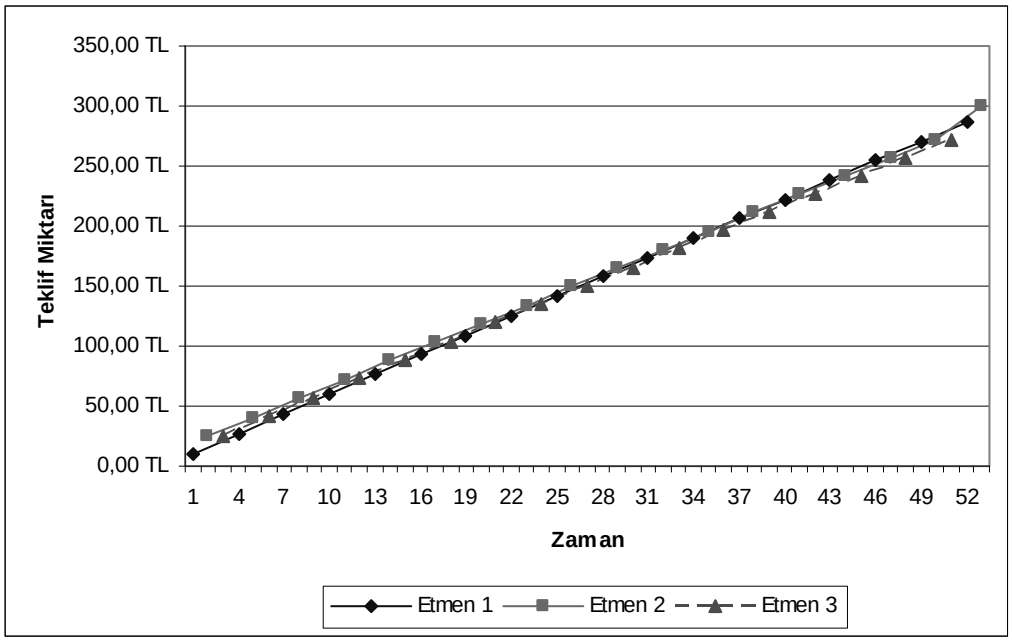
Deneyin amacı, bir müzayedeye katılan etmenlerin farklı stratejiler izlediği durumun müzayede ve etmenlerin verdiği tekifler üzerindeki etkilerini incelemektir. Deneyde 3 adet alıcı etmen ve 1 adet müzayede etmeni bulunmaktadır. Müzayededе satılan ürünün başlangıç satış fiyatı 10 TL, hemen alma fiyatı 300 TL'dir. Müşterilerin alıcı etmenler için girdikleri tercihler Çizelge 6.4'te sıralanmıştır.

Çizelge 6.4 : Deney 4'teki Müşteri Tercihleri.

	Etmen 1	Etmen 2	Etmen 3
Strateji	En Yüksek Teklif	Hemen Alma Fiyatı	Diğer Tekliflerin Ortalaması
Teklif Arttırma Oranı	5	5	5
Yeniden Kontrol Etme Süresi	1	1	1

Şekil 6.4'te deney 4 için teklif miktarının zamana göre değişmesi görülmektedir. Farklı stratejiler birlikte uygulandığında, müzayedenin teklif miktarı doğrusal bir artış göstermiştir. Grafik incelediğinde, hemen alma fiyatına göre teklif yapan 2 numaralı etmenin müzayedenin başlangıcında yüksek arttırma yaptığı, fakat müzayede ilerledikçe arttırma miktarını azalttığı gözlenmiştir. En yüksek teklife göre teklif yapan 1 numaralı etmen ise 2 numaralı etmenin tersi bir şekilde davranış göstererek, müzayedenin başlangıcında arttırma miktarını düşük tutmuş, müzayede

ilerledikçe tekliflerinin arttırma miktarını yükseltmiştir. Grafikte de görüleceği gibi, diğer tekliflerin ortalamasını göz önüne alarak teklif yapan 3 numaralı etmen, arttırma oranını 5 olarak belirlediği için arttırma miktarını sürekli düşük tutmuştur. Zamana göre bu teklif miktarları birbirlerini dengelemiş ve ürünün değeri doğrusal bir artış göstermiştir.



Şekil 6.4 : Deney 4 Sonuçları.

6.5 Deney 5

Deneyin amacı, etmenlerin birden fazla müzayedeye katıldıkları durumun müzayede ve etmenlerin verdikleri teklifler üzerindeki etkilerini incelemektir. Bu amaçla aynı ürünün satıldığı iki müzayede düzenlenmiş ve bu müzayedelere üç etmen katılmıştır. Deneyde alıcı etmenlerin tercihlerinin müzayede ve teklifler üzerindeki etkilerini en aza indirmek için üç alıcı etmenin de tercihleri birbirinin aynısı olarak belirlenmiştir. Ayrıca her zaman diliminde bir etmenin teklif vereceği şekilde etmenlerin müzayedelere katılmaya başlama süreleri ayarlanmıştır. Satıcı etmenlerin, düzenledikleri müzayedeler ile ilgili tercihleri Çizelge 6.4'te; alıcı etmenlerin, müzayedelere katılmak için belirledikleri tercihler de Çizelge 6.5'te sıralanmıştır.

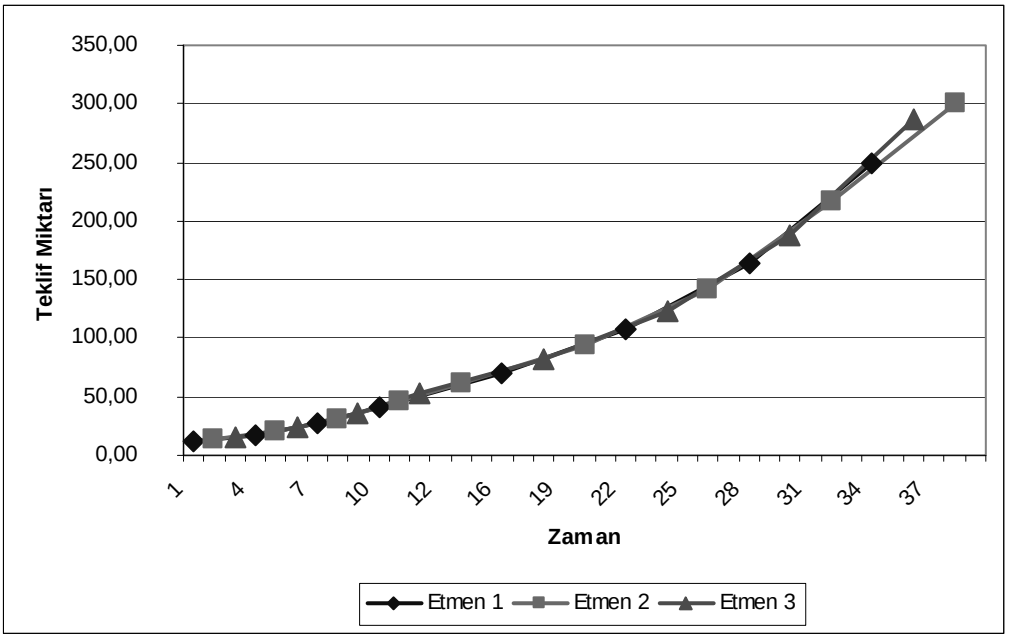
Çizelge 6.5 : Deney 5'teki Satıcı Tercihleri.

	Müzayede 1	Müzayede 2
Başlangıç Fiyatı	10 TL	50 TL
Hemen Alma Fiyatı	300 TL	300 TL

Çizelge 6.6 : Deney 5'teki Müşteri Tercihleri.

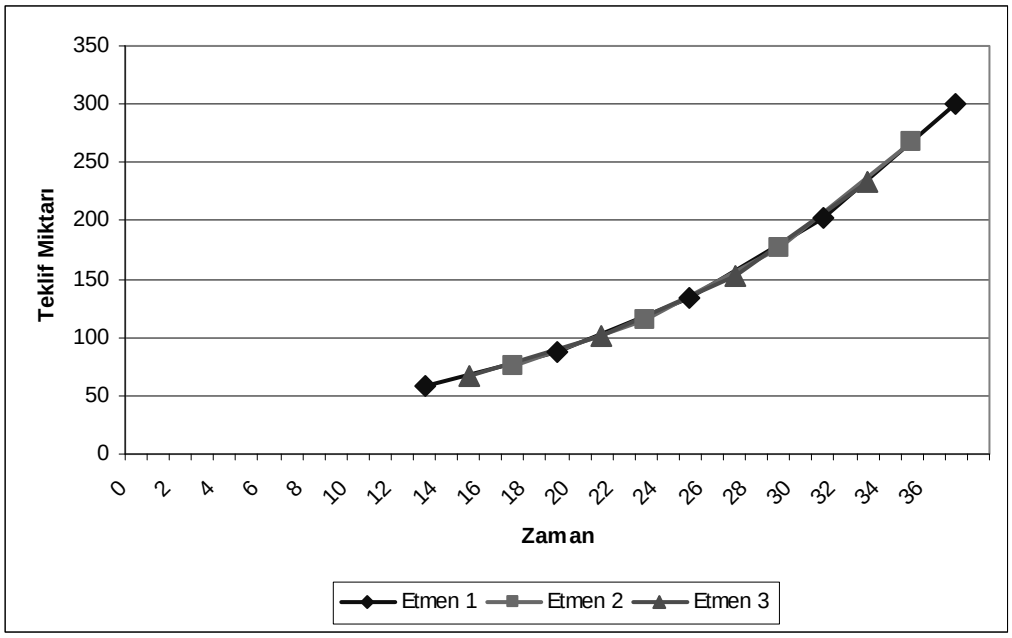
	Etmen 1	Etmen 2	Etmen 3
Strateji	En Yüksek Teklif	En Yüksek Teklif	En Yüksek Teklif
Teklif Arttırma Oranı	5	5	5
Yeniden Kontrol Etme Süresi	1	1	1

Şekil 6.5 ve Şekil 6.6'da görüldüğü gibi, etmenler 2 numaralı müzayedenin başlangıç fiyatının aşıldığı 12. zaman dilimine kadar sadece 1 numaralı müzayedeye teklif vermişlerdir. 13. zaman diliminden itibaren, 2 numaralı müzayedenin başlangıç fiyatı olan 50 TL aşılmış ve teklifler iki müzayedeye de yapılmıştır. İki müzayede de 300 TL'de kapanmış ve 1 numaralı müzayedeyi 2 numaralı etmen, 2 numaralı müzayedeyi ise 1 numaralı etmen kazanmıştır.



Şekil 6.5 : Deney 5 Sonuçları (Müzayede 1).

Müzayedelerin ve teklif veren etmenlerin özellikleri birbirlerine benzer oldukları için, 2 numaralı müzayedeye de teklif verilmeye başlanılan 13 numaralı zaman diliminden itibaren, her iki zaman diliminde bir kere bir müzayedeye teklif yapılmıştır. Bu durumun nedeni, en son teklif verilen müzayedenin en yüksek teklif miktarının diğer müzayededen yüksek olması ve bu nedenle en yüksek teklife göre teklifi hesaplayan etmen tarafından hesaplanan yeni teklif miktarının da diğer müzayededen yüksek olmasıdır. Etmen, daha düşük teklif verebileceği müzayedeye katılacağı için, en yüksek teklifi düşük olan müzayedeye teklif verecektir.



Şekil 6.6 : Deney 5 Sonuçları (Müzayede 2).

6.6 Deney 6

Deneyin amacı, çok sayıda ve farklı stratejilere sahip alıcı etmenlerin çok sayıda müzayede düzenlenmiş bir ortamdaki davranışlarını ve bu davranışların teklifler üzerindeki etkilerini gözlemlemektir. Bu amaçla aynı ürünün satıldığı 3 adet müzayede açılmış ve bu müzayedelere farklı strateji ve tercihlere sahip 9 adet alıcı etmenin katılması sağlanmıştır. Bu amaçla satıcı etmenlerin düzenledikleri müzayedeler ile ilgili tercihleri Çizelge 6.7'de; alıcı etmenlerin müzayedelere katılmak için belirledikleri tercihler de Çizelge 6.8'de sıralanmıştır.

Çizelge 6.7 : Deney 6'daki Satıcı Tercihleri.

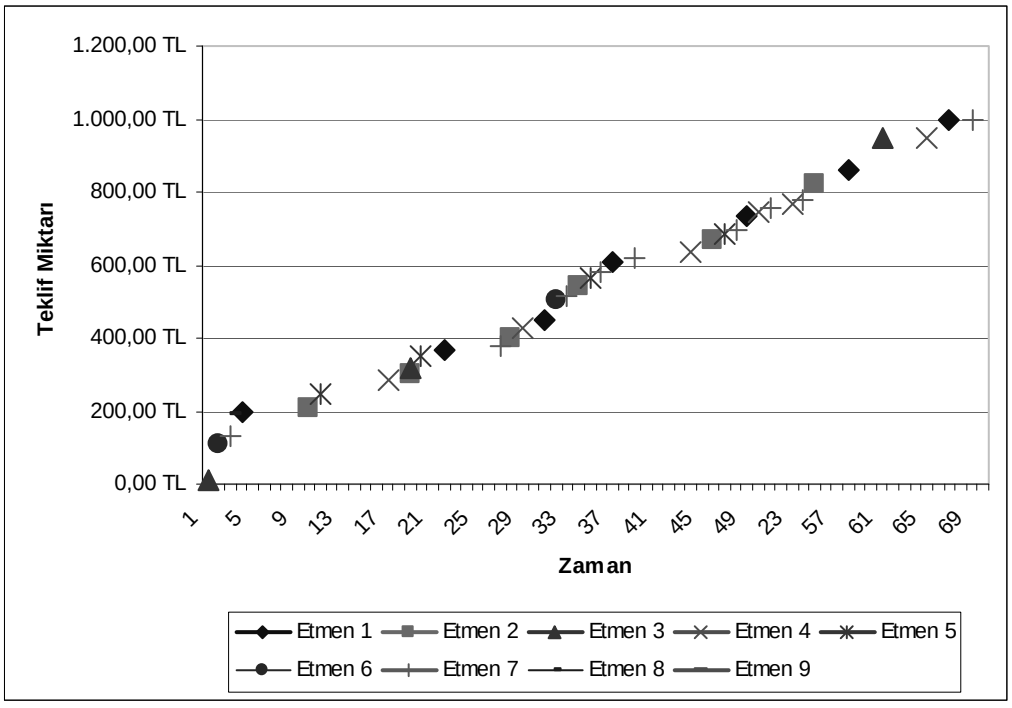
	Müzayede 1	Müzayede 2	Müzayede 3
Başlangıç Fiyatı	10 TL	100 TL	200 TL
Hemen Alma Fiyatı	1000 TL	1000 TL	1000 TL

Şekil 6.7'de 1 numaralı müzayededeki, Şekil 6.8'de 2 numaralı müzayededeki ve Şekil 6.9'da 3 numaralı müzayededeki teklif miktarının zamana göre değişimi verilmiştir. İlk olarak 1 numaralı müzayedeye teklif verilmiş, fakat daha sonra hemen alma fiyatına göre teklif veren etmenlerin yüksek teklifler vermesi sonucunda kısa sürede diğer müzayedelere de teklif verilmeye başlanmıştır.

Çizelge 6.8 : Deney 6'daki Müşteri Tercihleri.

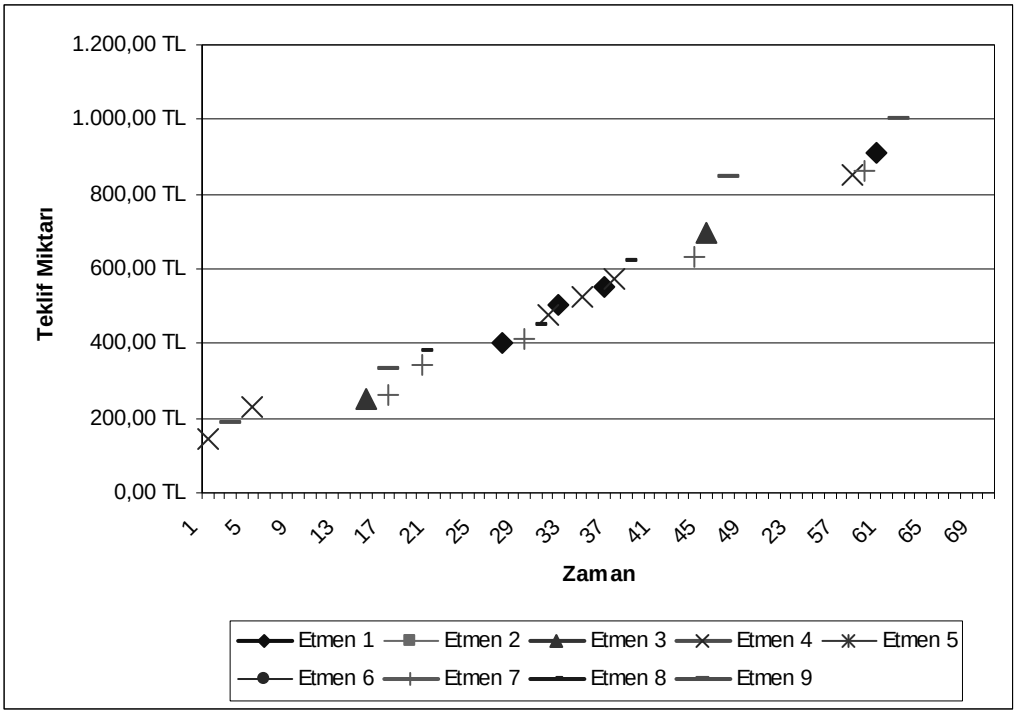
	Strateji	Teklif Arttırma Oranı	Yeniden Kontrol Süresi
Etmen 1	En Yüksek Teklif	5	1
Etmen 2	En Yüksek Teklif	5	3
Etmen 3	En Yüksek Teklif	10	5
Etmen 4	Hemen Alma Fiyatı	5	1
Etmen 5	Hemen Alma Fiyatı	5	3
Etmen 6	Hemen Alma Fiyatı	10	5
Etmen 7	Diğer Tekliflerin Ortalaması	20	1
Etmen 8	Diğer Tekliflerin Ortalaması	50	3
Etmen 9	Diğer Tekliflerin Ortalaması	100	5

1 numaralı müzayedeyi 7 numaralı etmen kazanmıştır. Bu etmenin, teklif arttırma oranı fazla olmamasına rağmen müzayedeyi kazanmasında en önemli etken müzayedeleri yeniden kontrol etme süresinin sıklığıdır. Bu müzayedede ikinci ve üçüncü gelen etmenler 1 ve 4 numaralı etmenlerdir. Bu etmenlerin ikisi de müzayedeleri en sık kontrol eden etmenlerdir. Fakat hemen alma fiyatına göre strateji belirleyen 4 numaralı etmen çok fazla arttırma yapamadığı için avantajını kaybetmiştir. Müzayede 1'in sonucuna göre farklı strateji ve tercihlerde alıcıların bulunduğu bir müzayedede, hemen alma fiyatına göre veya diğer tekliflerin ortalamasına göre teklif verilmesi ve müzayedelerin durumunun sık sık kontrol edilmesi avantajlıdır.



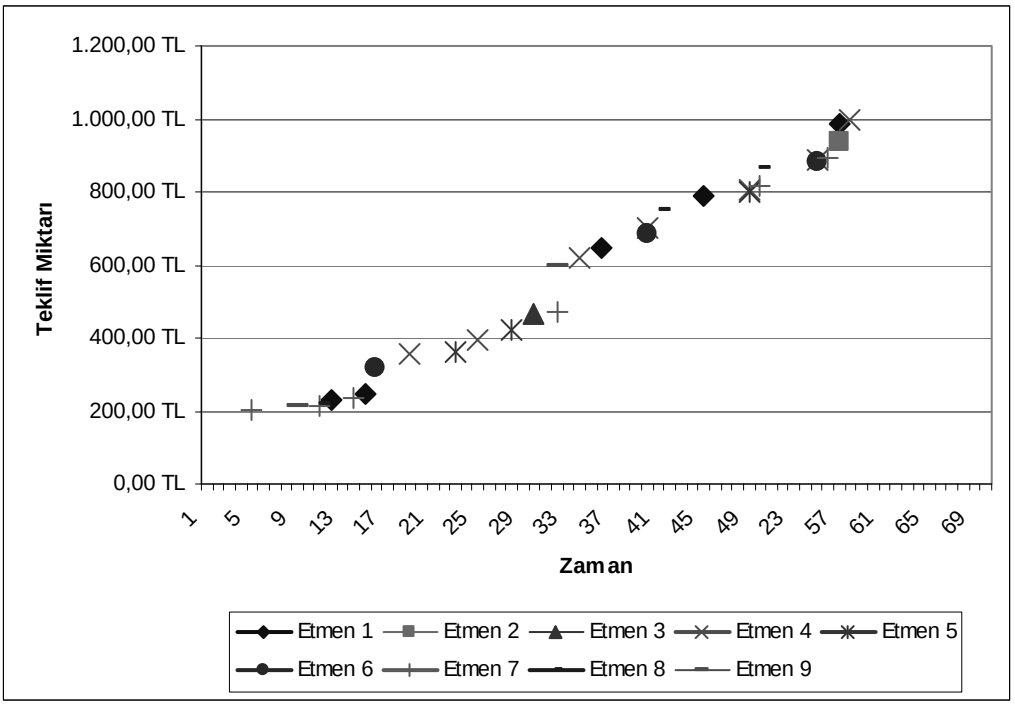
Şekil 6.7 : Deney 6 Sonuçları (Müzayede 1).

2 numaralı müzayedeyi 9 numaralı etmen kazanmıştır. Bu etmenin kazanmasını sağlayan etken, çok yüksek bir teklif arttırma oranı belirlemiş olmasıdır. Bu nedenle müzayedenin kapanma fiyatına çok çabuk erişebilmiştir. Bu müzayedede ikinci ve üçüncü gelen etmenler; 1 numaralı müzayedede de ikinci olan 1 numaralı etmen ve 1 numaralı müzayedeyi daha sonraki bir zaman diliminde kazanan 7 numaralı etmendir. Müzayede 2'nin sonucuna göre, çok yüksek arttırma oranına sahip olan ve diğer tekliflerin ortalamasına göre tekliflerini belirleyen alıcılar, farklı stratejiler uygulayan etmenlerin bulundukları bir ortamda müzayedeleri sık sık kontrol etmeseler de kazanma şansları yüksektir.



Şekil 6.8 : Deney 6 Sonuçları (Müzayede 2).

3 numaralı müzayedeyi 4 numaralı etmen kazanmıştır. 4 numaralı etmenin kazanmasına neden olarak müzayedeyi en sık kontrol eden etmenler arasında olması ve izlediği hemen alma fiyatına göre teklif belirleme stratejisinde, belirlenen teklif hemen alma fiyatının %95'inden yüksek ise hemen alma fiyatını teklif etmesi gösterilebilir. Bu müzayedeyi kazanan 4 numaralı etmenin, 1 numaralı müzayedeyi kazanamamasının sebebi, kritik olan son teklifinde belirlediği teklifin hemen alma fiyatından %95'inden çok az miktarda da olsa küçük olmasıdır. Müzayede 3'ün sonucunda, diğer müzayedelerde de ortaya çıkan müzayedeyi çok sık kontrol etme stratejisinin önemi tekrar ortaya çıkmıştır.



Şekil 6.9 : Deney 6 Sonuçları (Müzayede 3).

7. SONUÇLAR VE TARTIŞMA

Bu çalışma dağıtık çoklu etmen sistemi kullanılarak geliştirilmiş bir müzayede sisteminin ayrıntılarını içermektedir. Gerçek hayatta karşılaşılan müzayedeleri takip etme ve teklif belirme sorununa bir çözümdür. Yazılımların insanlar yerine karar verebilmesi ve onlar adına hareket edebilmesine örnek olarak verilebilecek bir sistemdir.

Sistemin gerçeklenmesi sırasında JADE ortamının özelliklerinden azami bir şekilde yararlanılmış ve FIPA standartlarına uyan bir dağıtık çoklu etmen sistemi meydana getirilmiştir. Uygulamada her farklı rol için birer etmen türü oluşturularak dağıtık bir sistem üzerinde kendi başına esnek kararlar verebilen akıllı etmenlerin birlikte çalışmaları sağlanmıştır. Etmenlerin, yerine karar verdikleri kullanıcıların tercihlerini bir grafik arabirim sayesinde alması ve yaptıkları hareketleri bu arabirim aracılığıyla kullanıcılara geri bildirmeleri sağlanarak hareketlerinin takip edilmesi sağlanmıştır. Ayrıca etmenlerin, kullanıcılar adına verecekleri kararların belirlenmesi için çeşitli algoritmalar üretilmiştir.

Sistem bir müzayede sistemi olarak tasarlanmış olmasına rağmen bir satış ve gezgin satın alma sistemi olarak da kullanılabilir. Ürün satıcılarının ürünlerini müzayede düzenlemeden tek bir fiyat belirleyerek satabilmeleri, alıcıların da bu tarz satışlara açık arttırma yapmadan katılabilmeleri sağlanmıştır. Bu şekilde müzayede sisteminin bir satın alma sistemi olarak genişlemesi sağlanmıştır. Bu genişlemedeki tek sınırlama, müzayedelerde tek ürün satışa çıkarıldığı için satış olarak açılan müzayedelerde de tek ürünün satışa çıkarılmasıdır. Satışa çıkarılacak ürünlerin her adedi için farklı bir müzayede açmak gereklidir. Sistem bu açıdan müzayede evleri ve satıcı etmenler için stok yönetimi içerecek ve bir müzayedede birden fazla ürün satılacak şekilde geliştirilebilir.

Sistem üzerinde yapılan testler sonucunda, bazı tercihlerin ve stratejilerin bazı durumlarda diğer stratejilere üstün olduğu belirlenmiştir. Bu nedenle etmenin müzayedede izlediği stratejiye ek olarak diğer etmenlerin de müzayedelerde

izledikleri stratejilerin önemli olduğu ortaya çıkmıştır. Sistem bu açıdan, bir alıcı etmenin diğer alıcı etmenlerin müzayedede izlediği stratejileri gözlem yapıp tahmin ederek kendi stratejisini değiştirmesi sağlanarak geliştirilebilir. Tasarlanan sistemdeki strajilerden olan diğer etmenlerin tekliflerine göre teklif belirleme yöntemi, bu öneriye yakın bir stratejidir.

Sistemdeki eksik noktalar, müzayede sonucunda ödeme yapılması ve etmen haberleşmesinin güvenliği konularıdır. Bu özellikler, tasarlanan sistemin amacı ve konusu olan müzayede problemine yönelik olmadıkları için dikkate alınmamışlardır. Gelecekteki ödeme sistemleri ve güvenlik üzerine yoğunlaşmış çalışmalar ile bu eksik noktalar kapatılabilir.

KAYNAKLAR

[1] **Wooldridge, M.,** 2002. An Introduction to Multiagent Systems, John Wiley & Sons Ltd, London

[2] **Weiss, G., Wooldridge, M.,** 1999. Multiagent Systems A Modern Approach to Distributed Artificial Intelligence, The MIT Press, Cambridge, Massachusetts

[3] **Franklin, S., Graesser, A.,** 1996. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996, 6.

[4] **Nwana, H.,** 1996. Software Agents: An Overview, *Knowledge Engineering Review*, Vol. **11**, No 3, September 1996, 1-40.

[5] **Lange, D., Oshima, M.,** 1999. Seven good reasons for mobile agents, *Communications of the ACM*, Vol. **42**, Issue 3, March 1999, 88-89.

[6] **Jennings ve diğ.,** 1998. A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems*, **1**, 1998, 275-306.

[7] **Burmeister ve diğ.,** 1997. Application of multi-agent systems in traffic and transportation, *Software Engineering. IEE Proceedings*, Vol. **44**, Issue 1, February 1997, 51-60.

[8] **Nishibe ve diğ.,** 1993. Distributed Channel Allocation in ATM Networks, *Global Telecommunications Conference*, **1**, 1993, 417-423.

[9] **Schoonderwoerd ve diğ.,** 1997. Ant-like agents for load balancing in telecommunications networks, *Proceedings of the first international conference on Autonomous agents*, Marina del Rey, California, United States, 1997, 209-216.

[10] **JADE resmi sayfası,** <*http://jade.tilab.com/*>, alındığı tarih 04.04.2009.

[11] **Repast Simphony resmi sayfası**, <*http://repast.sourceforge.net/*>, alındığı tarih 04.04.2009.

[12] **Swarm resmi sayfası**, <*http://swarm.org/index.php/Swarm_main_page*>, alındığı tarih 04.04.2009.

[13] **NetLogo resmi sayfası**, <*http://ccl.northwestern.edu/netlogo/*>, alındığı tarih 04.04.2009.

[14] **MASON resmi sayfası**, <*http://cs.gmu.edu/~eclab/projects/mason/*>, alındığı tarih 04.04.2009.

[15] **Bellifemine ve diğ.**, 2003, JADE A White Paper, *Telecom Italia EXP magazine*, Vol. **3**, No.3, 2003, 6-19.

[16] **Bellifemine ve diğ.,** 2007. Developing Multi-Agent Systems with JADE, John Wiley & Sons Ltd, London.

[17] **Bellifemine ve diğ.,** 2007. JADE Programmers Guide, *Telecom Italia Lab*, 2007.

[18] **Caire, G.,** 2007. JADE Tutorial – JADE Programming for Beginners, *Telecom Italia Lab*, 2007.

[19] **SC0001L**, 2002. FIPA Abstract Architecture Specification, Geneva, Switzerland.

[20] **SC00061G**, 2002. FIPA ACL Message Structure Specification, Geneva, Switzerland.

[21] **SC00037J**, 2002, FIPA Communicative Act Library Specification Geneva, Switzerland.

[22] **Cancedda, P., Caire, G.,** 2008. Creating Ontologies by Means of the BeanOntology Class, *Telecom Italia S.p.A*, October 2008.

[23] **Yokoo, M.**, 2006. Theory of Internet Auctions *IEEE International Conference on Systems, Man and Cybernetics*, Taipei, Taiwan, Vol. **2**, 8-11 October 2006, 1244-1249.

[24] **Anthony, P., Jennings, N.,** 2003. Developing a bidding agent for multiple heterogeneous auctions, *ACM Transactions on Internet Technology*, Vol. **3**, Issue 3, August 2003, 185-217.

[25] **Sarıkaya, M.,** 2002. Asimetrik Bilgi Çerçevesinde Müzayedeler, *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, Cilt **3**, Sayı 2, 2002.

[26] **Takayuki ve diğ.,** 2000. BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions, *Fourth International Conference on Multi-Agent Systems (ICMAS 2000)*, 10-12 July 2000, Boston, MA, USA, 2000, 399.

[27] **Cuni et. al.,** 2004. MASFIT: Multiagent system for fish trading. *Proceedings of the 16th European Conference on Artificial Intelligence*, August 22-27, Valencia, Spain, 2004, 710-714.

[28] **Morris, J., Maes, P.,** 2000. Sardine: An Agent-facilitated Airline Ticket Bidding System, *the 4PthP International Conference on Autonomous Agents*, 2000, 96-103.

ÖZGEÇMİŞ

Ad Soyad: Ali Üllenoğlu

Doğum Yeri ve Tarihi: Adana, 30 Ekim 1983

Adres: Mecidiyeköy, İstanbul

Lisans Üniversite: İstanbul Teknik Üniversitesi