İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SANAL FİRMALARA YÖNELİK ETMEN VE ROL TABANLI WEB SERVİS PAZARI

DOKTORA TEZİ

Ali DURMUŞ

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

HAZİRAN 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SANAL FİRMALARA YÖNELİK ETMEN VE ROL TABANLI WEB SERVİS PAZARI

DOKTORA TEZİ

Ali DURMUŞ
(504022211)

Tezin Enstitüye Verildiği Tarih : 29 Eylül 2010
Tezin Savunulduğu Tarih : 28 Haziran 2011

Tez Danışmanı : Prof. Dr. Nadia ERDOĞAN(İTÜ)

Diğer Jüri Üyeleri : Prof. Dr. Bülent ÖRENCİK (İTÜ)
Prof. Dr. Levent AKIN (BÜ)
Prof. Dr. Oya KALIPSIZ(YTÜ)
Doç. Dr. Şima ETANER UYAR(İTÜ)

HAZİRAN 2011

Eşime ve aileme,

ÖNSÖZ

Tez çalışmam süresince beni her koşulda destekleyen ve daima bana yol gösteren sayın hocam Prof. Dr. Nadia Erdoğan'a, manevi desteğini hiçbir şart altında esirgemeyen sevgili eşim Müge Durmuş'a ve beni her zaman destekleyen aileme en içten teşekkürlerimi sunarım.

Eylül 2010 Ali Durmuş

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **v**
İÇİNDEKİLER ..... **vii**
KISALTMALAR ..... **xi**
ÇİZELGE LİSTESİ ..... **xiii**
ŞEKİL LİSTESİ ..... **xv**
ÖZET ..... **xvii**
SUMMARY ..... **xxiii**
1. GİRİŞ **1**
1.1 Problemin İncelenmesi 1
1.2 Çözüm Önerisi 3
1.3 Tezin Amacı Ve Özgün Katkıları 5
1.3.1 Sanal firma özellikleri ve çözümleri 6
1.3.2 Özgün katkılar 8
1.4 Tezin Bölümleri 10
2. YAZILIM ETMENLERİ, SERVİSLER, ROLLER **13**
2.1 Yazılım Etmenleri Ve Sistemleri 13
2.1.1 Etmen nedir? 13
2.1.2 Çoklu etmen sistemi 14
2.1.3 Hareketli etmenler 15
2.2 Servis Odaklı Hesaplama 17
2.2.1 Temel servis odaklı mimari 18
2.2.2 Web servis mimarisi 20
2.2.2.1 XML ve özellikleri 21
2.2.2.2 SOAP (Basit Nesne Erişimi Protokolü) 22
2.2.2.3 WSDL (Web Servis Tanımlama Dili) 23
PortType elemanı 24
Message elemanı 24
Type elemanı 25
Binding elemanı 25
Servis gerçekleme tanımı 25
2.2.2.4 UDDI 25
UDDI yüksek seviyeli mimarisi 26
UDDI yayın API'si 26
UDDI kimlik denetleme modeli 26
UDDI data model 26
UDDI çağırma modeli 27
UDDI sorgulama API'si 28
2.2.3 Servis odaklı mimari ve etmenler 28
2.3 Roller 30
2.3.1 Rol nedir? 30
2.3.1.1 Rol tabanlı erişim kontrolü (RTEK) 30
2.3.1.2 Nesneye yönelik programlamada rol 31

2.4 JAVA Dili ve Katkıları 32
2.4.1 Yansıma 32
2.4.2 Etiketleme 34
2.5 JADE Etmen Çatısı 35
2.5.1 JADE mimarisi 36
2.5.2 JADE paketleri 38
2.5.3 JADE araçları 39
2.5.4 JADE diğer özellikleri 42
3. ETMEN TABANLI WEB SERVIS PAZARI (EWSP) 45
3.1 Etmen Web Servisi Sistemleri 45
3.1.1 Servis tabanlı organizasyonlar arası iş akışı 45
3.1.1.1 Değerlendirme 46
3.1.2 Sanal web servisleri- web servislerinin kişiselleştirilmesi 47
3.1.2.1 Sanal web servisi için yazılım etmeni ve mimari 48
3.1.2.2 Değerlendirme 50
3.1.3 Hisse senedi ticareti simülasyon sistemi 51
3.1.3.1 Akıllı etmen sistemleri ile web servis entegrasyonu 51
3.1.3.2 STSS'nin mimari tasarımı 51
3.1.3.3 Sistem gerçeklenmesi 53
3.1.3.4 Değerlendirme 53
3.1.4 JACK etmen çatısının web servisi için genişletilmesi 53
3.1.4.1 Değerlendirme 54
3.1.5 Web servisleri için etmen yönelimli yöntem 54
3.1.5.1 Değerlendirme 56
3.1.6 Web servisleri ile hareketli etmen sistemlerini tümleştirmek 56
3.1.6.1 MA2WS 57
3.1.6.2 WS2MA 58
3.1.6.3 Değerlendirme 58
3.1.7 Web servisi sağlamlığını artırmak için etmen kullanımı 58
3.1.7.1 Değerlendirme 59
3.2 Etmen Web Servisi Sistemlerinin Değerlendirilmesi 60
3.3 Rol Tabanlı Etmen Sistemleri 60
3.3.1 Gaia 61
3.3.2 AALAADDIN 62
3.3.3 ROPE 64
3.3.4 TRUCE 65
3.3.5 BRAIN 67
3.3.6 RoMAS 69
3.4 Rol Tabanlı Etmen Sistemlerinin Değerlendirilmesi 70
4. ETMEN TABANLI WEB SERVIS PAZARI (EWSP) 73
4.1 EWSP Sistem Yapısı 74
4.2 Sanal Firma 75
4.2.1 Sanal firma başvuru süreci 76
4.3 EWSP'de Roller 78
4.4 EWSP'de Servisler 79
4.5 Örnek Uygulama Servis-Rol-Etmen İlişkisi 81
5. ETMEN TABANLI WEB SERVIS PAZARI (EWSP) MİMARİSİ 85
5.1 EWSP Mimarisi Bileşenleri 86
5.1.1 EWSP web 86
5.1.2 EWSP etmeni 89

5.1.3 EWSP yönetici etmeni 94
5.1.4 EWSP bütünleşik servis yöneticisi etmenleri 97
5.1.5 Sanal firma ve sanal çalışan etmen 97
5.1.5.1 Servis tanımı 98
5.1.6 İstemci (Müşteri) 100
5.2 Servis Kaydı 100
5.3 Role Servis XSD (EWSPX) 101
5.4 EWSP'de İhale Yönetimi 104
5.4.1 Servis istek kriterleri 105
5.4.2 İhale süreci 107
5.4.2.1 Servis sağlayıcı adayların belirlenmesi 107
Tekil Servis Sağlayıcı Adayların Belirlenmesi 108
Bütünleşik Servis Sağlayıcı Adayların Belirlenmesi 108
5.4.2.2 Servis sağlayıcı adayların tarihsel performansının hesaplanması 109
5.4.2.3 Teklif değerlendirme aşaması 109
5.4.2.4 İhalenin sonuçlandırılma aşaması 110
5.4.2.5 İhale sonrası takip 111
6. BAŞARIM DEĞERLENDİRMESİ 113
6.1 Windows Test Ortamı 114
6.1.1 Test senaryo 1 115
6.1.1.1 Servis bulma adımlarının değerlendirilmesi 117
6.1.2 Test senaryo 2 118
6.1.3 Test senaryo 3 119
6.1.4 Test senaryo 4 121
6.2 Test Ortamı Linux 123
6.2.1 Linux ile test senaryo 1 124
6.3 Windows Ve Ubuntu Çalışmaları Karşılaştırılması 126
6.4 Değerlendirme ve Sonuçlar 126
7. SONUÇLAR VE ÖNERİLER 129
KAYNAKLAR 131
EKLER 137
ÖZGEÇMİŞ 149

KISALTMALAR

XML	: Extensible Markup Language
XSD	: XML Schema Document
WSDL	: Web Service Definition Language
SOAP	: Simple Object Access Protocol
JADE	: Java Agent DEvelopment Framework
DF	: Directory Facilator
HTTP	: Hypertext Transfer Protocol
JVM	: Java Virtual Machine
UDDI	: Universal Description, Discovery, and Integration
URI	: Uniform Resource Identifier
W3C	: World Wide Web Consortium
API	: Application Programming Interface
FIPA	: Foundation for Intelligent Physical Agents
EWSP	: Etmen Tabanlı Web Servis Pazarı
EWSPX	: EWSP Role Servis XSD

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Rol tabanlı etmen sistemlerinin değerlendirilmesi. 71
Çizelge 5.1 : EWSP'deki annotation sınıfları ve tanımları..... 99
Çizelge 6.1 : İhale süreci temel adım süreleri..... 118
Çizelge 6.2 : Servis çağrısı temel adımlar ve süreler..... 121

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Sanal çalışan, sanal firma, rol ve servis ilişkisi. 7
Şekil 2.1 : Ortam içindeki etmen. 14
Şekil 2.2 : Çoklu etmen sistemi genel yapısı. 15
Şekil 2.3 : Çoklu etmen sistemi genel yapısı. 16
Şekil 2.4 : İstemci-sunucu modeli ile hareketli etmen modeli karşılaştırılması. 16
Şekil 2.5 : Temel servis odaklı mimari. 19
Şekil 2.6 : XML şemaları ile XML belgeleri arasındaki ilişki. 22
Şekil 2.7 : SOAP iletisinin temel yapısı. 23
Şekil 2.8 : WSDL elemanları. 24
Şekil 2.9 : portType elemanı. 24
Şekil 2.10 : Message elemanı. 25
Şekil 2.11 : WSDL service ve port elemanı. 25
Şekil 2.12 : UDDI yapıları arasındaki ilişki. 27
Şekil 2.13 : Sınıf hiyerarşisi. 31
Şekil 2.14 : JADE mimarisi. 36
Şekil 2.15 : JADE mimari UML diyagramı. 37
Şekil 2.16 : JADE RMA. 39
Şekil 2.17 : JADE DA. 40
Şekil 2.18 : JADE sniffer etmeni. 40
Şekil 2.19 : JADE introspector etmeni. 41
Şekil 2.20 : JADE DF etmeni. 41
Şekil 2.21 : Farklı taşıyıcılara dağıtılmış JADE platformu 42
Şekil 3.1 : SCW ortamına entegre etmenler. 46
Şekil 3.2 : Web servisleri, etmenler, sanal web servisleri ve kullanıcıları. 48
Şekil 3.3 : STSS mimarisi. 52
Şekil 3.4 : JACK etmen çatısının genişletilmiş hali 53
Şekil 3.5 : Etmen web servis çağrısı örneği. 56
Şekil 3.6 : Hareketli etmen web servisi tümleştirmesi 57
Şekil 3.7 : N adet etmenin web servisi vermesi. 59
Şekil 3.8 : Gaia metodolojisinin modelleri arasındaki ilişkiler. 61
Şekil 3.9 : Temel model. 62
Şekil 3.10 : ROPE bileşenleri. 64
Şekil 3.11 : TRUCE ilkel basamakları. 66
Şekil 3.12 : BRAIN çatısı. 68
Şekil 4.1 : EWSP sistemi genel yapı. 75
Şekil 4.2 : Sanal çalışan-sanal firma başvuru. 77
Şekil 4.3 : Sanal çalışan rol servis ilişkisi. 77
Şekil 4.4 : Sanal firma rol servis ilişkisi. 79
Şekil 4.5 : Rol tanım XSD. 80
Şekil 4.6 : Örnek sanal firma rol servis. 83
Şekil 5.1 : EWSP genel mimari. 85

Şekil 5.2 : EWSP mimarisi. 87
Şekil 5.3 : Servis XSD. 90
Şekil 5.4 : Servis tanımı WSDL dönüşümü. 94
Şekil 5.5 : EWSP servis yaratma. 101
Şekil 5.6 : Rol tanım XSD. 103
Şekil 5.7 : İhale süreci. 105
Şekil 6.1 : Tek aday ile servis bulma. 116
Şekil 6.2 : Tek aday ile bütünleşik servis bulma. 117
Şekil 6.3 : Birden fazla aday ile servis bulma. 119
Şekil 6.4 : Birden fazla EWSP yöneticisi ile servis bulma ortalama süre. 120
Şekil 6.5 : Normal web servis - etmen servis çağrısı karşılaştırması. 122
Şekil 6.6 : Tekil etmen servis – bütünleşik etmen servis çağrı karşılaştırması. 123
Şekil 6.7 : Tek aday ile servis bulma. 125
Şekil 6.8 : Windows-ubuntu karşılaştırması tek aday ile servis bulma. 126
Şekil A.1 : Uygulama ana sayfa. 139
Şekil A.2 : Uygulama yönetici ekranı 139
Şekil A.3 : Servis listesi ekranı 140
Şekil A.4 : İhale talepleri ekranı 140
Şekil A.5 : İhale adayları değerlendirme ekranı 141
Şekil A.6 : İhale kazananların liste ekranı 141
Şekil A.7 : Servis çağrıları liste ekranı 142
Şekil A.8 : Sanal firma rol listesi 142
Şekil A.9 : Sanal firma servis listesi 143
Şekil A.10 : Sanal firma başvuru listesi 143
Şekil A.11 : Sanal firma çalışan listesi 144
Şekil A.12 : Sanal çalışan firma listesi 144
Şekil A.13 : Sanal çalışan sanal firma başvuru ekranı 145
Şekil A.14 : Sanal çalışan rol listesi 145
Şekil A.15 : Sanal firma ihale mesajları 146
Şekil A.16 : Sanal firma kazanılan ihale listesi 146
Şekil A.17 : Sanal firma çağrı listesi 147

SANAL FİRMALARA YÖNELİK ETMEN VE ROL TABANLI WEB SERVİS PAZARI

ÖZET

Gelişen internet teknolojisi firmalara yeni iş fırsatları ve iş olanakları yaratmaktadır. Teknolojinin sağladığı olanaklar, firmalara, internet aracılığı ile hizmetlerini (servislerini) satabilme ve aynı şekilde ihtiyaç duydukları hizmetleri (servisleri) satın alabilme fırsatı sunmaktadır. Alım ve satım işlemlerinde coğrafi sınırlar bir engel olmaktan çıkmış, firmalar çok farklı yerlere servis satabilir duruma gelmiştir. Hatta bazen, firmaların satın aldığı bir servis, sattıkları bir başka serviste kullanılabilmektedir.

Pazarda oluşan anlık taleplere firmalar hızlı bir şekilde yanıt vermek durumundadır. Yine aynı şekilde, pazarda oluşan değişimlere karşı firmalar da dinamik olarak yeniden organize olmak durumundadır. Yeni olanaklar ve pazar dinamizmi firmaları yapısal olarak dinamik olmaya zorlamaktadır. Firmalar içsel organizasyonlarını ve çalışan yapısını pazar ihtiyaçlarını ve dinamizmini karşılayabilmek için hızlı bir şekilde değiştirmek durumunda kalabilmektedir. Bu doğrultuda, firma içindeki alt birimlerin yapılarının değiştirilmesi (içsel organizasyon değişikliği), yeni eleman alımı veya gerektiğinde eleman çıkarma işlemleri yapılabilmektedir. Teknolojik gelişmeler firmalara sadece alış-satışlar için fırsat yaratmamış, kendi içsel yapıları içinde yeni dinamizm olanakları yaratmıştır. Firmalar kendi işleri için dış kaynak kullanabilmekte, hızlı bir şekilde ihtiyaç duydukları elemanları işe almakta ya da bazen kiralayabilmektedir. Yine firmalar teknolojinin getirdiği olanakları kullanarak çalışanlarının işe gelmeden firma faaliyetlerini yürütmesine izin vermektedir.

Firmaların dinamik piyasa koşullarına ayak uydurarak servislerini pazarda satabilmeleri için aşağıdaki temel sorunların çözümlenmesi gerekmektedir.

- Firma yapılarının dinamik olarak oluşturulmasına imkan sağlamak gerekmektedir.

- Firmaların eleman ihtiyaçlarının hızla giderilmesi için eleman alımı ve bu elemanların servislerle ilişkilendirme işlemlerinin hızlıca yapılabilmesi gerekmektedir.
- Firmaların elemanlarını ofis ortamından bağımsız olarak çalıştırabilmeleri gerekmektedir.
- Firmaların servislerini satabilecekleri bir ortamın var olması gerekmektedir.
- Firmaların servislerinin farklı ortamlardan kullanılabilmesi gerekmektedir.
- Müşterilerin servis satın alabilecekleri bir ortam var olması gerekmektedir.
- Müşteriye ihtiyacına göre birden fazla firmanın servislerinin birleştirilmesi (bütünleşik servis) olanağının sağlanması gerekmektedir.

Birbirlerinden farklı ortamlara hizmet verebilme ve geniş bir pazara ulaşma söz konusu olduğunda, web servisi teknolojisi bu gereksinimleri karşılayan en uygun yapıdır. Web servisleri heterojen ortamlarda çalışma yeteneği ve servis tabanlı yapısı ile diğer sistemlere göre öne çıkmaktadır. Servis tabanlı mimari atomik servislere yoğunlaşmıştır ve bütün işler birer servis olarak kurgulamıştır. Web servislerinin, temel özellikleri olan ortamdan bağımsızlık, gevşek bağlılık ve modülerlik ile firmaların dinamik piyasa koşullarında karşılaşacakları problemlerin çözümlerinde kullanımı uygun olmaktadır.

Olumlu yanlarına rağmen web servislerinin firmaların gereksinimlerini karşılayamadığı pek çok nokta bulunmaktadır. Dinamik firma yapısının oluşturulması, firma içindeki yetkilerin dağıtılması gibi yetenekleri web servislerinin sağlaması mümkün değildir. Web servisleri ve klasik programlama teknikleri bu tip problemlere çözüm getirmemektedir. Web servisleri, yapısını değiştiremeyen, işbirliği ve koordinasyon sağlayamayan statik yapılardır. Dinamizm, işbirliği ve koordinasyon söz konusu ise, bir ortamda bulunan ve tasarım amacına ulaşmak için ortamda özerk hareket etme yeteneğine (otonom) sahip bilgisayar yazılımları olan etmenler akla gelmektedir. Etmen insan veya diğer sistemlerin müdahalesi olmadan kendi başına karar verme yeteneğine sahiptir, kendi iç durumu ve hareketleri üzerinde kontrolü vardır. Etmenler ve web servisleri özellikleri itibari ile birbirlerini tamamlayıcı durumundadır.

Web servislerine dayalı bir çözüm farklı ortamlardan çağrılma ve kullanılma problemini çözmekte ve büyük problemleri küçük problemlere ayırma yaklaşımı getirmekte, ancak etmenlerin doğal bir özelliği olan dinamizm, etkileşim, koordinasyon ve işbirliği konularında bir çözüm getirememekte, ya da yetersiz kalmaktadır. İki yaklaşımın kendine özgü yetenekleri birbirini tamamlar niteliktedir ve bu çalışmada beraber kullanılmıştır.

Yukarıda belirtilen gereksinimleri karşılamak ve problemleri çözmek için, tez çalışması kapsamında, **sanal firmalara yönelik bir etmen ve rol tabanlı web servis pazarı** yaklaşımı önerilmiştir. Bu tezde ortaya konulan konulardan biri olan **sanal firma** sanal çalışanlardan ve sanal yöneticilerden oluşan bir çoklu etmen sistemidir. Bu tezin konusu olan sanal firma ile gerçek hayattaki firma yapılarının sanal ortama taşınması ve sanal ortamda servislerinin dış dünyaya açılması ve pazarlanması sağlanmış ve bunun için bütünleşik bir çatı ortaya konulmuştur. Yöneticilerden ve çalışanlardan oluşan sanal firma içinde yetenekler, mevkiler ve yetkiler roller aracılığı ile belirlenmiştir. Roller belirli servislerle ilişkilendirilmiştir. Rolleri edinen çalışan, sanal firmanın rolle ilişkilendirilmiş olan firma servislerini sunma yeteneği kazanmaktadır. Sanal firma servisleri etmen tabanlı web servis pazarında (EWSP) satılabilecektir. Sanal firma rolleri EWSP'de web servislerine dönüştürülmekte, web servisleri ise EWSP'de pazarlanabilmektedir. EWSP sistemi üzerinden web servis talebinde bulunan istemci/müşterilerin talepleri doğrultusunda sistemde ihale yapılıp ve talebi karşılamak için servis sağlayıcı etmenler belirlenmektedir. EWSP üzerinde bir istemci talebinin karşılanması için birden fazla servis bir araya getirilerek bütünleşik servisler oluşturulabilmektedir. İhale sonrası sanal firma taahhütleri (ihaleyi kazanan servis sağlayıcının) EWSP sistemi tarafından takip edilebilmektedir.

Sanal firma modelinin tez çalışması bağlamında geliştirilen temel özellikleri şunlardır

- Sanal firma sanal yönetici/kurucu ve sanal çalışanlardan oluşur.
- Rol kavramı kullanılarak sanal çalışanlar belirli servisler konusunda yetkilendirilir ve sorumluluk verilir.
- Sanal firma içindeki roller dinamik olarak çalışanlara dağıtılır veya geri alınır. Rollerin çalışanlara verilmesi veya alınması çalışma zamanında olmaktadır.

- Sanal firma içindeki bir rol bir veya daha fazla servis ile ilişkilendirilebilir. Rol kavramının tasarımla bütünleşmiş olması sisteme esneklik ve dinamizm sağlamıştır. Sanal firma fonksiyonlarının rollere dağıtılması ve roller aracılığı ile yönetilmesi sanal firmalara günümüz koşullarında ihtiyaç duydukları dinamizmi kazandırmıştır.
- Rol tanımları dinamik olarak bir xml dosyası ile yapılır. Böylece, tanımlamalar çalışma zamanında gerçeklenmiş olmaktadır. Sanal firma tanımlamalarının dinamik şekilde yapılabilmesi için, çalışma kapsamında, bir rol tanımlama dili olan EWSPX geliştirilmiştir. EWSPX kullanılarak rol tanımları rahat ve hızlı bir şekilde yapılabilmekte, geliştiriciler kendi sanal firmalarını kolay bir şekilde sisteme entegre edebilmektedirler.
- Bir rol, atandığı çalışana, rol içerisinde yer alan servisleri sunma hakkı/yetkisi verir.
- Sanal firmaya sanal çalışanların katılımının sağlanması için bir çalışma ortamı oluşturulmuştur. Adaylar yürütme sırasında bir başvuru gerçekleştirirler ve sanal yönetici tarafından yapılan değerlendirme sonucunda başvuruları sonuçlandırılır. Bu şekilde sanal firmalara yapısal dinamizm de kazandırılmıştır.
- Sanal firma ölçeklenebilirdir. Sanal firma içindeki yönetici ve çalışanlar aynı bilgisayar sisteminde olmak durumunda değildir. Farklı sistemler üzerinde çalışabilirler.
- Sanal firma yapısı esnektir. Dinamik ve çalışma zamanlı rol dağıtımı ile sanal firmanın kendi içsel hiyerarşisi esnek ve hızlı bir şekilde değişebilir. Aynı şekilde çalışanlarının yetenek ve yetkileri de değişebilir.
- Sanal firma rol yapıları esnek ve dinamiktir. Roller xml dosyası ile sistem açılışında tanımlandığı için esnek ve dinamik olarak tanımlanabilir. Rollerin içerdikleri servisler (rol servis ilişkisi) dinamik bir şekilde tanımlanabilmektedir.

EWSP sistemi özellikleri şunlardır:

- Sanal firmaların servisleri web servislerine dönüştürülür ve sisteme kayıt edilir.

- İstemciler (müşteriler) tarafından iletilen servis talepleri karşısında uygun servisin belirlenmesi amacıyla bir ihale hizmeti gerçeklenir.
- İstemcilerin taleplerini yerine getirmek için birden fazla servis bir araya getirilerek bütünleşik servis oluşturulabilir.
- Bütünleşik servis, bütünleşik servis yürütücüsü tarafından yürütülür. Bütünleşik servis yürütücüsü kendisine gelen talepleri ve çağrıları bütünleşik servisi oluşturan alt servislere ileterek sonucu oluşturur.
- İstemciler servis talepleri için sadece servis kriterlerini kullanmaz, aynı zamanda kendi kriterlerinin ağırlıklarını da vererek uygun bir servis bulabilirler. Böylece istemcinin ihtiyaçlarına uygun web servisi elde etme imkanı artmaktadır.
- İstemcinin talebi olan servisi bulmak için yapılan ihale sonrasında ihaleyi kazanan etmen taahhüdü takip edilir. Taahhüt takibi ile müşterinin hakları korunmakla birlikte ihalelere katılan rakip etmenlerin de hakları korunmaktadır.
- Kazanan etmenin taahhütlerini yerine getirme düzeyi diğer ihaleler için girdi olarak kullanılmaktadır. Yani taahhütlerini yerine getiren etmen yeni ihalelerde daha avantajlı duruma gelirken taahhütlerini yerine getirmeyen etmen yeni ihalelerde dezavantajlı duruma gelmektedir.
- EWSP ölçeklenebilir yapıda kurgulanmıştır. Servis talebinde ortaya çıkan duruma göre, ihtiyaç dahilinde birden fazla EWSP Yönetici etmeni yaratılıp, yük paylaştırılabilir. Sisteme katılan sanal firmalar farklı makinelerde çalışabilmektedir.
- EWSP sistemi açıktır. EWSP sistemi web üzerinden çalıştığı için çok farklı işletim sistemleri üzerinde çalışan sistemler ile etkileşim halinde olabilir. Farklı işletim sistemi üzerinde çalışan sanal firmalar bu sisteme dahil olabilir.

Bu çalışma içinde, yukarıda kısaca değinilen özellikleri taşıyan bütünleşik bir çatı ve yürütme ortamı tasarlanmış ve gerçeklenmiştir. Sistemin başarımını ölçmeye yönelik deneysel sonuçlar öngörülen hedefler ile uyumludur.

A ROLE AND AGENT BASED WEB SERVICE MARKET FOR VIRTUAL COMPANIES

SUMMARY

Developments in internet technologies provide companies with new business potentials and opportunities. Companies can sell services that they produce, and at the same time, they can buy services that they need. Geographical boundaries are no longer an obstacle, as companies can carry out their activities and exchange services regardless of their location. They may even buy services in order to be able to provide their own services that they sell.

Companies have to respond quickly to instant demands that occur in the market. Likewise, companies need to reorganize against the changes in the market. They are forced to be structurally dynamic in order to respond to new opportunities and market dynamics. Companies' internal organizational structure and employees may change quickly to meet market needs and dynamics. In this respect, changing the structure of sub-units within companies (internal organizational changes), hiring or dismissing of employees may be required. Technological developments enable not only the marketing of services, but also allow companies to reorganize their internal structure dynamically. Companies can use outsourcing for their business, they recruit needed staff or sometimes hire in a quick way. Again, using the opportunities of technology, companies may allow employees to conduct company's activities without being present at office.

The following main issues have to be resolved for companies to adapt to dynamic market conditions and sell their services.

- the creation of the dynamics of company structure
- hiring and association of employees with companies' services
- conducting company's activities without being present at office
- the presence of a virtual environment where companies can market their services

- enriching services with features that allow them to be used in different environments
- the presence of a virtual environment where customers can search for and buy services
- providing the ability to merge services of different companies to meet customer demands: creation of composite services dynamically

When providing services to different environments and reaching a broad market is the main concern, the web service technology is the most appropriate technology to meet these requirements. Web services with their ability to work in heterogeneous environments and their service based architecture stand out when compared to other systems. Service-based architecture focuses on an atomic service and each requirement is met through a service. Therefore, web services with their environment/technology independence, loosely coupling and modularity characteristics are appropriate to resolve the problems companies encounter in dynamic market conditions.

Despite their positive aspects, still, certain needs that companies face cannot be met by web services. It is not possible to provide some capabilities such as creating a dynamic company structure or distribution of privileges within the company through the use of web services. Web services and conventional programming techniques do not provide solutions to these problems. Web services are static entities which cannot change their structure dynamically, and, they neither have cooperation and coordination abilities.

When dynamism, cooperation and coordination are main issues, agent systems come to mind. An agent is computing system, which stays in an environment, and has an ability to act autonomously in that environment to achieve its design goals. An agent has the ability to make a decision without the intervention of humans or other systems and it has a control over its own internal state and movements.

The features of agents and web services complement each other. A web services based method can solve issues related to invocation and usage in different environments and also with the approach of dividing big problems into smaller problems, can produce better solutions. However, it is inadequate to solve problems concerning dynamism, interaction, coordination and cooperation, at which agents

excel. Consequently, both approaches have distinctive capabilities and they complement one another when used together.

To meet the above requirements and to solve the related problems, this thesis proposes **a role and agent based web services market for virtual companies**. One of the contributions introduced in this thesis is the concept of a **virtual company**, which is a multi-agent system that represents a virtual organization consisting of a virtual manager and a number of virtual employees. The concept of a virtual company carries the structure of an actual company into virtual environments and allows it to operate in a virtual market by opening its services to the outside world. An integrated framework that provides the means for these capabilities has been implemented during the course of thesis work. The positions, qualifications and privileges of the members of the virtual company, the manager and the employees, are determined through roles that are assigned to them. Each role is associated with a set of services. An employee who is assigned a certain role gains the ability/responsibility to provide the services that are associated with that role. A virtual company's services can be marketed in a virtual market environment, **an agent based web service market (AWSM)** which is implemented in the context of this theses. Virtual company's roles are transformed into web services and can be marketed in AWSM. When clients or customers request web services, the AWSM framework announces a tender to determine service provider agents that are willing to meet client's request. If the request cannot be met by a single service, a number of them can be merged into a composite service. After the bid finalizes, the virtual company's commitments (service provider that won the tender) are followed and saved in a history file by AWSM system.

The virtual company model that is proposed and implemented in context of this thesis possesses the following basic features:

- A virtual company consists of a virtual manager/founder and virtual employees.
- Via the role concept, virtual employees are authenticated and given responsibility for certain services.
- Roles within virtual companies are distributed to employees or taken back. Role assignment or cancellation takes place dynamically at runtime.

- A role within a virtual company can be associated with a single or several services. The integration of role concept with agents has extended the flexibility and dynamism of the system. The distribution and management of the functionality of a virtual company via roles enhances the dynamism a virtual company requires in today's conditions.
- Roles are defined with an XML file, enabling definition of roles to be implemented at runtime. **AWSMX**, a role specification language, has been developed to describe dynamically a virtual company. As role definitions can be made easily and quickly, developers can easily integrate their own virtual company into the system by using EWSPX.
- The role concept provides the ability to assign an employee the services associated with that role.
- A virtual execution environment that facilitates employee participation into virtual companies has been created. Candidates apply virtual company at runtime and the applications are finalized after the evaluation of a virtual manager. This manner of execution introduces structural dynamism into a virtual company.
- A virtual company with its executives and employees does not have to be located on the same computer system; they can be running on different systems.
- Virtual company is flexible. Its internal hierarchy is flexible and can change quickly at runtime via a new role distribution. Likewise, employees' skills and privileges may also change.
- Virtual company's role structure is flexible and dynamic. Roles are defined by XML files at the startup of the system in a flexible manner. Services included in role (role-service association) can also be defined in a dynamic way.

The agent based web service market fulfills the following features:

- Virtual companies' services are converted into web services and are registered in system.
- A tender is announced to determine the appropriate service/services to meet clients (customers) service request.

- To fulfill client requests, one or more services may be merged to form a composite service.
- Composite services are executed by the **composite service executer**. Composite service executer divides service request into sub-requests that match sub-services of a composite service, delivers them service providers, and comprises the overall response according to sub-responses.
- Clients not only use the service criteria for service requests but also can specify their own weighted criteria to find the proper service. This opportunity increases the possibility of obtaining a web service that most meets the client's need.
- Tenders are put after clients declare their requests for services. The commitment of service provider agent that won the tender is tracked and monitored. By monitoring commitment, customer rights are preserved and also the rights of other competitor agents participating the tender are protected.
- The winning agent's fulfillment level of a commitment is used as an input for new tenders. Hence, an agent fulfilling its commitments becomes more advantageous in new tenders.
- AWSM is a scalable. According to service demand, if needed, more than one AWSM Manager Agent can be created and the load can be shared. Companies participating in the system can run on different virtual machines.
- AWSM system is open. It can interact with different systems since it is designed to run on the web. Virtual companies can participate AWSM regardless of the location and the type of the system on which they are deployed.

In this thesis, an integrated framework and a runtime environment with the features briefly described above are designed and implemented. Experimental results that aim to measure the performance of the system has been found to be compatible with the envisaged target.

1. GİRİŞ

Bu bölümde öncelikle tez çalışmasının konusu olan, günümüzde firmaların kendisini sanal ortamda var etme ve bu ortamda faaliyet göstererek tamamen sanallaşmasına ilişkin gereksinimler ve karşılaşılan problemler incelenecektir. Daha sonra, söz konusu gereksinimleri karşılamak üzere, tez çalışması kapsamında geliştirilmiş olan sanal firma kavramına dayalı çözüm kısaca tanıtılacak ve tezin özgün katkıları özetlenecektir.

1.1 Problemin İncelenmesi

Gelişen internet teknolojisi firmalara yeni iş fırsatları ve iş olanakları yaratmaktadır. Teknolojinin sağladığı olanaklar, firmalara, internet aracılığı ile hizmetlerini (servislerini) satabilme ve aynı şekilde ihtiyaç duydukları hizmetleri (servisleri) satın alabilme fırsatı sunmaktadır. Alım ve satım işlemlerinde coğrafi sınırlar bir engel olmaktan çıkmış, firmalar çok farklı yerlere servis satabilir duruma gelmiştir. Hatta bazen, firmaların satın aldığı bir servis, sattıkları bir başka serviste kullanılabilmektedir.

Pazarda oluşan anlık taleplere firmalar hızlı bir şekilde yanıt vermek zorundadır. Yine aynı şekilde, pazarda oluşan değişimlere karşı firmalar da dinamik olarak yeniden organize olmak durumundadır. Yeni olanaklar ve pazar dinamizmi firmaları yapısal olarak dinamik olmaya zorlamaktadır. Firmalar içsel organizasyonlarını ve çalışan yapısını, pazar ihtiyaçlarını ve dinamizmini karşılayabilmek için hızlı bir şekilde değiştirmek zorunda kalabilmektedir. Bu doğrultuda, firma içindeki alt birimlerin yapılarının değiştirilmesi (içsel organizasyon değişikliği), yeni eleman alımı veya gerektiğinde eleman çıkarma işlemleri yapılabilmektedir. Teknolojik gelişmeler firmalara sadece alış-satışlar için fırsat yaratmamış, kendi içsel yapıları içinde yeni dinamizm olanakları yaratmıştır. Firmalar kendi işleri için dış kaynak kullanabilmekte, hızlı bir şekilde ihtiyaç duydukları elemanları işe almakta ya da zaman zaman kiralayabilmektedir. Yine firmalar teknolojinin getirdiği olanakları

kullanarak çalışanlarının işe gelmeden firma faaliyetlerini yürütmesine izin vermektedir.

Firmaların dinamik piyasa koşullarına ayak uydurarak servislerini pazarda satabilmeleri için aşağıdaki temel sorunların çözümlenmesi gerekmektedir.

- Firma yapılarının dinamik olarak oluşturulmasına imkan sağlanması gerekmektedir.
    - Firmaların değişen pazar koşullarına hızlı bir şekilde uyum gösterebilmesi için firma yapısının hızlı, değişebilir, dinamik bir yapıda olması gerekmektedir.
- Firmaların eleman ihtiyaçlarının hızla giderilmesi için eleman alımı ve bu elemanların servislerle ilişkilendirme işlemlerinin hızlıca yapılabilmesi gerekmektedir.
    - Günümüz şartlarında firmalar hızlı bir şekilde eleman almakta ve işe alınan elemanın belirli rollere sahip olması ve belirli servisleri verebilmesi gerekmektedir.
- Firmaların elemanlarını ofis ortamından bağımsız olarak çalıştırabilmeleri gerekmektedir.
    - Firma çalışanlarının esnek çalışma koşullarına sahip olması gerekmektedir.
- Firmaların servislerini satabilecekleri bir ortamın var olması gerekmektedir.
    - Firmaların servislerini satabilecekleri ortamın yaratılması ve varlığını sürdürmesi için koşulların sağlanması gerekmektedir.
- Firmaların servislerinin farklı ortamlardan kullanılabilmesi gerekmektedir.
- Müşterilerin servis satın alabilecekleri bir ortam var olması gerekmektedir.
    - Müşteriler için;
        - Fiyat olarak en uygun,
        - Performans olarak en uygun servisin sorgulanıp bulunabilmesi mümkün olmalıdır
    - Satış sonrası taahhütlerin takip edilmesi gerekmektedir.

- Müşteriye ihtiyacına göre birden fazla firmanın servislerinin birleştirilmesi (bütünleşik servis) olanağının sağlanması gerekmektedir.

1.2 Çözüm Önerisi

Birbirlerinden farklı ortamlara hizmet verebilme ve geniş bir pazara ulaşma söz konusu olduğunda, web servisi teknolojisi bu gereksinimleri karşılayan en uygun yapıdır. Web servisleri heterojen ortamlarda çalışma yeteneği ve servis tabanlı yapısı ile diğer sistemlere göre öne çıkmaktadır. Servis tabanlı mimari atomik servislere yoğunlaşmıştır ve bütün işler birer servis olarak kurgulanmıştır. Web servislerinin, aşağıda yer alan özellikleri nedeniyle, firmaların dinamik piyasa koşullarında karşılaşacakları problemlerin çözümlerinde kullanımları uygun olmaktadır.

Ortam bağımsız: Web servisler farklı ortamlardan çağrılabilmektedir.

Gevşek bağlı: Servis odaklı mimari gereği kullanılan ve kullanan sistem arasındaki bağ gevşektir. Birisi diğerini etkilemez.

Modülerlik: Belirli bir işi yapmak amacıyla tasarlandıkları için web servisleri modüler bir yapıdadırlar ve tekrar tekrar kullanılabilirler. Bu yaklaşımda büyük problemler küçük parçalar haline getirilir ve çözülür.

Ancak, web servislerinin firmaların gereksinimlerini karşılayamadığı pek çok nokta bulunmaktadır. Dinamik firma yapısının oluşturulması, firma içindeki yetkilerin dağıtılması gibi yetenekleri sağlaması mümkün değildir. Web servisleri ve klasik programlama teknikleri bu tip problemlere çözüm getirmemektedir. Web servisleri, yapısını değiştiremeyen, işbirliği ve koordinasyon sağlayamayan statik yapılardır. Dinamizm, işbirliği ve koordinasyon söz konusu ise, bir ortamda bulunan ve tasarım amacına ulaşmak için ortamda özerk hareket etme yeteneğine (otonom) sahip bilgisayar sistemleri olan etmenler akla gelmektedir. Etmen, insan veya diğer sistemlerin müdahalesi olmadan kendi başına karar verme yeteneğine sahiptir, kendi iç durumu ve hareketleri üzerinde kontrolü vardır.

Yapılan çalışmalar göstermiştir ki etmenler web servislerinin yeteneklerini aşağıda belirtilen noktalarda arttırmaktadır [1];

- Web servisi sadece kendinden haberdar iken, etmenler hem kendilerinden, hem de etkileşim halinde oldukları diğer etmenler ve onların yeteneklerinden haberdardırlar.
- Etmenler yapıları/tanımları gereği etkileşim halinde olabilmektedir; oysa web servisi kendisi çağrılana kadar pasif durumdadır.
- Web servisleri otonom değildir. Otonomluk etmenlerin doğal bir özelliğidir.
- Etmenler işbirliğine açıktır. Bir problemin çözümü için birlikte hareket edebilirler. Web servislerinde ortak hareket etme özelliği bulunmamaktadır.

Web servislerine dayalı bir çözüm farklı ortamlardan çağrılma ve kullanılma problemini çözmekte ve büyük problemleri küçük problemlere ayırma yaklaşımı getirmekte, ancak etmenlerin doğal bir özelliği olan dinamizm, etkileşim, koordinasyon ve işbirliği konularında bir çözüm getirememekte, ya da yetersiz kalmaktadır. İki yaklaşımın kendine özgü yetenekleri birbirini tamamlar niteliktedir ve beraber kullanılabilir.

Çözüm; yukarda belirtilen gereksinimleri karşılamak ve problemleri çözmek için, tez çalışması kapsamında, **sanal firmalara yönelik bir etmen ve rol tabanlı web servis pazarı** yaklaşımı önerilmiştir. Literatürde sanal yapıların oluşturulmasına yönelik sanal işletme/sanal organizasyon gibi çalışmalar vardır. Grefen, Eshuis, Mehandjiev, Kouvas, Weichhart[53] çalışmalarında dinamik tedarik zincirleri oluşturmak ve tedarik işlemlerindeki karmaşık süreçleri yönetmek için etmen tabanlı anlık sanal organizasyon yapısı önermişlerdir. Sun, Huan[54] çalışmalarında çoklu etmen sistemleri kullanarak sanal işletmeleri modellemiş ve sanal işletmelerin risk yönetimini yapmışlardır. Rodriguez, Perez-Lancho, De Paz, Bajo, Corchado[55] çalışmalarında sanal organizasyon kavramlarını kullanarak açık çoklu etmen sistemi tasarlamışlardır. Açık çoklu etmen sistemi yeni bileşenlerin kolayca eklendiği, mevcut bileşenlerin sistemi kolayca terk ettiği ve çalışma ortamlarının dinamik olarak değiştiği sistemlerdir[55]. Sanal işletme partnerler arası belli bir amaçla geçici işbirliği şeklinde, sanal ortamda oluşturulan yapıdır [48]. Sanal işletmeler daha çok belli işleri yerine getirmek için dinamik oluşturulan ve daha sonra dağılan geçici yapılardır. Yine sanal organizasyonlar belli bir amaca ulaşmak için kaynakları paylaşan esnek, bağımsız dağıtık bireyler veya kurumların oluşturduğu ağdır [49,50]. Kaynaklar bilgiler, bilgisayar sistemleri, veri depoları ve ağ bağlantıları gibi

elemanlar olabilir. Sanal organizasyonların gerçeklenmesi, farklı problemlerinin (kaynak kullanımı, yönetilmesi) çözülmesi için farklı teknolojiler kullanılmıştır. Bu tanımlamalardan da yola çıkılarak oluşturulan ve bu tezde ortaya konulan konulardan biri olan **sanal firma**, sanal çalışanlardan ve sanal yöneticilerden oluşan bir çoklu etmen sistemidir. Bu tezin konusu olan sanal firma ile gerçek hayattaki firma yapılarının sanal ortama taşınması ve sanal ortamda servislerinin dış dünyaya açılması ve pazarlanması sağlanacak ve bunun için bütünleşik bir çatı ortaya konulacaktır. Yöneticilerden ve çalışanlardan oluşan sanal firma içinde yetenekler, mevkiler ve yetkiler roller aracılığı ile belirlenmektedir. Roller ise belirli servislerle ilişkilendirilmiştir. Rolleri edinen çalışan, sanal firmanın rolle ilişkilendirilmiş olan firma servislerini sunma yeteneği kazanacaktır. Sanal firma servisleri etmen tabanlı web servis pazarında (EWSP) satılabilecektir. Sanal firmadaki roller EWSP'de web servislerine dönüştürülmekte ve web servisleri pazarlanabilecektir. EWSP sistemi üzerinden web servis talebinde bulunan istemci/müşterilerin talepleri doğrultusunda sistemde ihale yapılacak ve talebi karşılamak için servis sağlayıcı etmenler belirlenecektir. EWSP üzerinde bir istemci talebinin karşılanması için birden fazla servis bir araya getirilerek bütünleşik servisler oluşturulabilecektir. İhale sonrası sanal firma taahhütleri (ihaleyi kazanan servis sağlayıcının) EWSP sistemi tarafından takip edilecektir.

1.3 Tezin Amacı Ve Özgün Katkıları

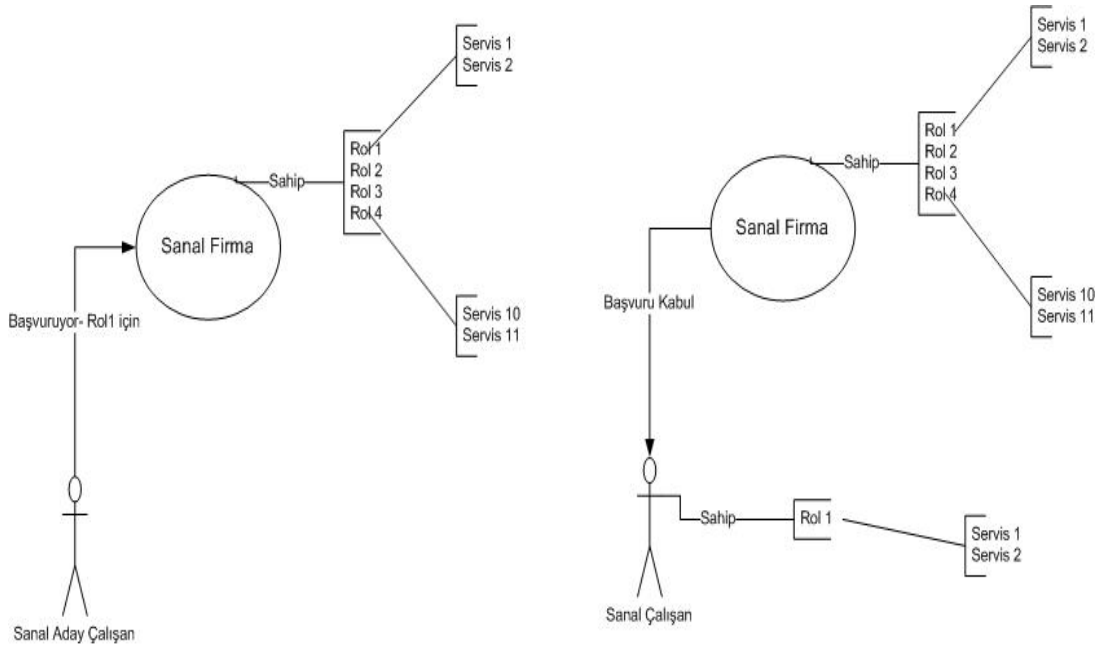
Tezin amacı etmen tabanlı web servis pazarını sanal firmalar ile birlikte gerçeklemektir. Gerçekleştirilen çatıya EWSP (Etmen Tabanlı Web Servis Pazarı) ismi verilmiştir. Tezin amaçları:

- Sanal firmalar için web servisi ve etmen teknolojisini kullanan bir çatı sunmak ve gerçeklemek.
- Sanal firmaların servislerini roller aracılığı ile sanal firma çalışanları arasında dağıtabilmek.
- Sanal firmaların servislerini satabilmeleri için etmen web servisi pazarı çatısını sunmak ve gerçeklemek

- Etmen tabanlı web servis pazarından ihtiyaç halinde birden fazla etmen servisinin bir araya gelerek bir bütünleşik servis oluşturmasına izin verecek bir yapı sunmak.
- Etmen tabanlı web servis pazarından bütünleşik servislerin çalışabilmesine olanak vermek.

1.3.1 Sanal firma özellikleri ve çözümleri

Çoklu etmen sistemleri sanal firmalar oluşturmak için kullanılabilecek bir yöntemdir. Etmen grupları sanal firmanın bileşenlerini (firma yöneticisi ve çalışanlar) oluşturacaktır. Her sanal firma bir yönetici (kurucu) etmen tarafından yaratılacaktır. Sanal firmada görev alacak sanal çalışanlar da birer etmen olarak tanımlanacaktır. Sanal çalışanlar sanal firmada işe girecek, istedikleri zaman ayrılacaktır. Sanal firma dinamik yapısını oluşturmak için sanal firmalar roller üzerine kurgulanacaktır. Firmalar genelde hiyerarşik bir yapıya sahiptirler. Bu yapı tanımlanırken yapı içersindeki birimlerin veya çalışanların sorumlulukları ve yetkileri roller üzerinden tanımlanmaktadır Bir rol belirli yetki ve sorumlulukları temsil eder. Firma farklı servisleri dinamik bir şekilde sunabilmelidir. Bu dinamizmi sağlamak üzere roller belirli servislerle ilişkilendirilebilirler. Firma içindeki bütün roller ve bu rollerin sahip oldukları servisler firma kurucusu tarafından tanımlanır. Bir çalışanın firma içindeki konumu sahip olduğu roller ile belirlenir. Bir çalışan bir veya daha fazla role sahip olabilir. Çalışan üstlendiği rollere bağlı olarak, o rolün ilişkilendirildiği servisleri sunma hakkına/yetkisine de sahip olacaktır. Rol tabanlı etmen sistemi sanal firma yapısını oluşturmak için kullanılabilir bir yöntemdir. Şekil 1.1'de yukarıda ifade edilen yapıya uygun sanal firma, sanal çalışan, rol ve servis ilişkisi gösterilmektedir.



Şekil 1.1 : Sanal çalışan, sanal firma, rol ve servis ilişkisi.

Firmaların en temel amaçlarından birisi servislerini, teknolojinin verdiği olanakları kullanarak en geniş pazara ulaştırabilmektir. En geniş alana erişim, farklı sistemlerde, ortam bağımsız bir şekilde çalışabilen web servisleri ile mümkün olabilir. Dolaysıyla, sanal firma servislerinin web servisleri olarak pazara sunulması uygun olacaktır. Firma servislerinin web servislerine dönüştürülmesini ve farklı müşteriler tarafından kullanılabilmesini sağlayacak altyapının kurulması bu çalışmanın hedeflerinden biridir.

Müşterilerin ihtiyaç duydukları servisleri bulabilmeleri ve kendi kriterlerine göre en uygun servis sağlayıcı sanal firmayı seçebilmeleri gerekmektedir. Müşteri için uygun servisin belirlenmesi sırasında fiyat ve performans gibi farklı kriterler kullanılabilir.

Bazı durumlarda, müşterilerin isterlerinin tümünün sadece bir tek sanal firma web servisi ile karşılanması mümkün olmayabilir. Bu gibi durumlarda, farklı sanal firmalarca sunulan web servislerinin bir araya getirilmesi ile oluşturulan bir bütünleşik servis önerilen altyapının sunduğu hizmet anlayışına önemli bir katkı yapacaktır. Farklı kriterler kullanılarak sanal pazarda bütünleşik servislerin oluşturulması sağlanacaktır.

Ayrıca, sanal firmalar ile müşterileri buluşturmak için bir sanal servis pazarı yaratılacaktır. Servis satın alımları sonrasında, satıcı sanal firmaların verdikleri

taahhütler takip edilecek ve edinilen bilgiler ışığında firma hakkında, ilerde başvurulmak üzere, taahhüt takip veritabanı oluşturulacaktır.

Sonuç olarak, bu çalışma, sanal firmalar için dinamik bir model ve pazar imkanı yaratmak üzere bir etmen tabanlı web servis pazarı altyapısına ait tasarım kararlarının alınması ve gerçeklenmesine ilişkin ayrıntıları içermektedir. .

1.3.2 Özgün katkılar

Bu tezin literatüre özgün katkıları üç noktada toplanabilir. Birincisi firmaların kendilerini sanal ortamda var edebilmeleri için önerilen yeni bir sanal firma modelidir. Diğeri, sanal firma içinde, rol ve servis kavramlarını bağdaştırarak, servislerin roller aracılığı ile sunulmasını olanaklı kılan özgün yapıdır. Üçüncüsü ise, sanal firmaların faaliyetlerini sürdürebilecekleri bir yürütme ortamı olan etmen tabanlı web servis pazarıdır.

Sanal firma modelinin tez çalışması bağlamında geliştirilen temel özellikleri şunlardır:

- Sanal firma sanal yönetici/kurucu ve sanal çalışanlardan oluşur.
- . Rol kavramı kullanılarak sanal çalışanlar belirli servisler konusunda yetkilendirilmekte ve sorumluluk verilmektedir.
- Sanal firma içersindeki roller dinamik olarak çalışanlara dağıtılır veya geri alınır. Rollerin çalışanlara verilmesi veya alınması çalışma zamanında olmaktadır.
- Sanal firma içindeki bir rol bir veya daha fazla servis ile ilişkilendirilebilir Rol kavramının tasarıma entegre edilmesi sisteme esneklik ve dinamizm sağlamıştır. Sanal firma fonksiyonlarının rollere dağıtılması ve roller aracılığı ile yönetilmesi sanal firmalara günümüz koşullarında ihtiyaç duydukları dinamizmi kazandırmıştır.
- Rol tanımları dinamik olarak bir xml dosyası ile yapılır. Böylece, tanımlamalar çalışma zamanında gerçeklenmiş olmaktadır. Sanal firma tanımlamalarının dinamik şekilde yapılabilmesi için, çalışma kapsamında, bir rol tanımlama dili olan EWSPX geliştirilmiştir. EWSPX kullanılarak rol tanımları rahat ve hızlı bir şekilde yapılabilmekte, geliştiriciler kendi sanal firmalarını kolay bir şekilde sisteme entegre edebilmektedirler.

- Bir rol, atandığı çalışana, rol içinde yer alan servisleri sunma hakkı/yetkisi verir.
- Sanal firmaya sanal çalışanların katılımının sağlanması için bir çalışma ortamı oluşturulmuştur. Adaylar yürütme sırasında bir başvuru gerçekleştirirler ve sanal yönetici tarafından yapılan değerlendirme sonucunda başvuruları sonuçlandırılır. Bu şekilde sanal firmalara yapısal dinamizm de kazandırılmıştır.
- Sanal firma ölçeklenebilirdir. Sanal firma içindeki yönetici ve çalışanlar aynı bilgisayar sisteminde olmak durumunda değildir. Farklı sistemler üzerinde çalışabilirler.
- Sanal firma yapısı esnektir. Dinamik ve çalışma zamanlı rol dağıtımı ile sanal firmanın kendi içsel hiyerarşisi esnek ve hızlı bir şekilde değişebilir. Aynı şekilde çalışanlarının yetenek ve yetkileri de değişebilir.
- Sanal firma rol yapıları esnek ve dinamiktir. Roller xml dosyası ile sistem açılışında tanımlandığı için esnek ve dinamik olarak tanımlanabilir. Rollerin içerdikleri servisler (rol servis ilişkisi) dinamik bir şekilde tanımlanabilmektedir.

EWSP sistemi özellikleri;

- Sanal firmaların servisleri web servislerine dönüştürülür ve sisteme kayıt edilir.
- İstemciler (müşteriler) tarafından iletilen servis talepleri karşısında uygun servisin belirlenmesi amacıyla bir ihale hizmeti gerçeklenir.
- İstemcilerin taleplerini yerine getirmek için birden fazla servis bir araya getirilerek bütünleşik servis oluşturulabilir.
- Bütünleşik servis, bütünleşik servis yürütücüsü tarafından yürütülür. Bütünleşik servis yürütücüsü kendisine gelen talepleri ve çağrıları bütünleşik servisi oluşturan alt servislere ileterek sonucu oluşturur.
- İstemciler servis talepleri için sadece servis kriterlerini kullanmaz, aynı zamanda kendi kriterlerinin ağırlıklarını da vererek uygun bir servis arayabilirler. Böylece istemcinin ihtiyaçlarına uygun web servisi elde etme imkanı artmaktadır.

- İstemcinin talebi olan servisi bulmak için yapılan ihale sonrasında ihaleyi kazanan etmen taahhüdü takip edilir. Taahhüt takibi ile müşterinin hakları korunmakla birlikte ihalelere katılan rakip etmenlerin de hakları korunmaktadır.
- Kazanan etmenin taahhütlerini yerine getirme düzeyi diğer ihaleler için girdi olarak kullanılmaktadır. Yani taahhütlerini yerine getiren etmen yeni ihalelerde daha avantajlı duruma gelmektedir.
- EWSP ölçeklenebilir yapıda kurgulanmıştır. Servis talebinde ortaya çıkan duruma göre, ihtiyaç dahilinde birden fazla EWSP Yönetici etmeni yaratılıp, yük paylaştırılabilir. Sisteme katılan sanal firmalar farklı makinelerde çalışabilmektedir.
- EWSP sistemi açıktır. EWSP sistemi web üzerinden çalıştığı için çok farklı işletim sistemleri üzerinde çalışan sistemler ile etkileşim halinde olabilir. Farklı sistemlerde çalışan sanal firmalar bu sisteme dahil olabilir.

1.4 Tezin Bölümleri

Tezin kitabının bundan sonraki bölümleri şunlardır:

İkinci bölümde etmenler, servis odaklı mimari, web servisleri, roller ve rol tabanlı mimari anlatılmıştır. Bu tez önerilen çözümün temelini oluşturan konular bu bölümde ele alınmıştır. Etmenler ile ilgili tanımlamalar yapılmıştır. Servis odaklı mimari ve web servisleri detaylı olarak incelenmiştir. Ayrıca benzer çalışma örnekleri hakkında bilgiler verilmiştir.

Üçüncü bölümde tezin gerçeklenmesinde kullanılan JADE etmen çatısı ve java dilinin tezin gerçeklenmesine katkısı olan özellikleri aktarılmıştır.

Dördüncü bölümde Etmen Tabanlı Web Servis Pazarı (EWSP) hakkında bilgi verilmiş ve sistemin temel kavramları olan servisler, roller, sanal firma, sanal çalışan ve bunların birbiriyle olan ilişkileri anlatılmıştır.

Beşinci bölümde EWSP mimarisi, mimari içersindeki elemanların ayrıntıları, mimari içersinde gerçekleşen servis kaydı, rol yapısı gibi kavramlar açıklanmıştır. Bu bölümde ayrıca EWSP içindeki ihale sisteminin detayları aktarılmıştır.

Altıncı bölümde EWSP sisteminin başarım değerlendirilmesi yapılmıştır. Değerlendirme yaparken farklı senaryolar kullanılmış ve bunların sonuçları değerlendirilmiştir.

Yedinci bölümde sonuçlar ve değerlendirmeler yer almaktadır.

2. YAZILIM ETMENLERİ, SERVİSLER, ROLLER

Bu bölümde tezde kullanılan temel kavramlar olan yazılım etmenleri, servisler ve roller hakkında detaylı bilgi aktarılacaktır. Ayrıca sistemin gerçeklenmesinde kullanılan teknolojiler olan java dilinin teze katkısı ve etmen çatısı olan JADE anlatılacaktır.

2.1 Yazılım Etmenleri Ve Sistemleri

Bu bölümde yazılım etmenleri hakkında detaylı bilgi verilecektir. Yazılım etmen yapıları ve yazılım etmen sistemleri aktarılacaktır.

2.1.1 Etmen nedir?

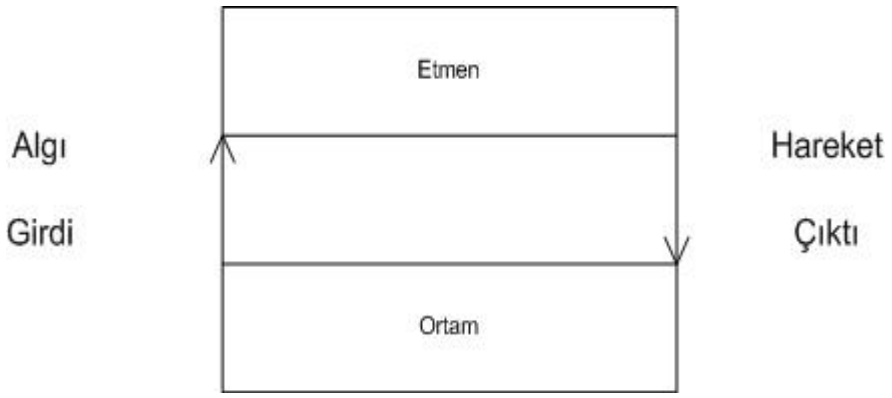
Evrensel olarak kabul edilmiş bir etmen tanımı bulunmamaktadır; bu konu üzerindeki tartışmalar devam etmektedir. Özerklik kavramının etmen tanımının merkezinde olduğu genel olarak kabul görmüştür fakat bunun ötesinde çok az bir uzlaşı vardır. Tek bir tanım üzerinde görüş birliğine ulaşmakta karşılaşılan temel sorun, etmen ile ilişkilendirilmiş özelliklerin farklı alanlarda farklı önemlere sahip olmalarıdır. Örneğin, bazı uygulamalar için etmenin deneyimlerinden öğrenme yeteneği çok önemli iken, başka uygulamalar için bu pek de önemli olmamaktadır.

Etmen bir ortamda bulunan ve tasarım amacına ulaşmak için ortamda özerk hareket etme yeteneğine (otonom) sahip bilgisayar sistemidir [2]. Özerk hareket etme konusunu biraz daha açar isek, etmen insan veya diğer sistemlerin müdahalesi olmadan kendi başına karar verme yeteneğine sahiptir, kendi iç durumu ve hareketleri üzerinde kontrolü vardır. Bu tanıma uygun en basit ve genel etmen tipi olarak aşağıdaki Şekil 2.1'i alabiliriz. Etmen ortamı etkilemek için, ortamdan elde ettiği algı girdileri kullanarak bir hareket çıktısı üretmektedir.

Bu özellikler dışında etmenlerde genelde aşağıdaki özellikler de bulunabilir [3]:

- Reaktif: Ortamda olan değişimleri algılayan ve yanıt verebilen

- Amaç yönelimli: Basit bir şekilde ortama yanıt veren değil amaç doğrultusunda hareket eden.
- İletişim halinde: Diğer etmenler ile ve belki de insan ile iletişim kurabilen.
- Öğrenen: Daha önceki deneyimlerinden yola çıkarak davranışlarını değiştirebilen.
- Hareketli: Kendisini bir makineden başka makineye taşıyabilen.
- Esnek: Hareketleri katı ve kesin değil, değişebilen.
- Akıllı: Karar verme yeteneğine sahip



Şekil 2.1 : Ortam içindeki etmen, [2] den uyarlanmıştır.

2.1.2 Çoklu etmen sistemi

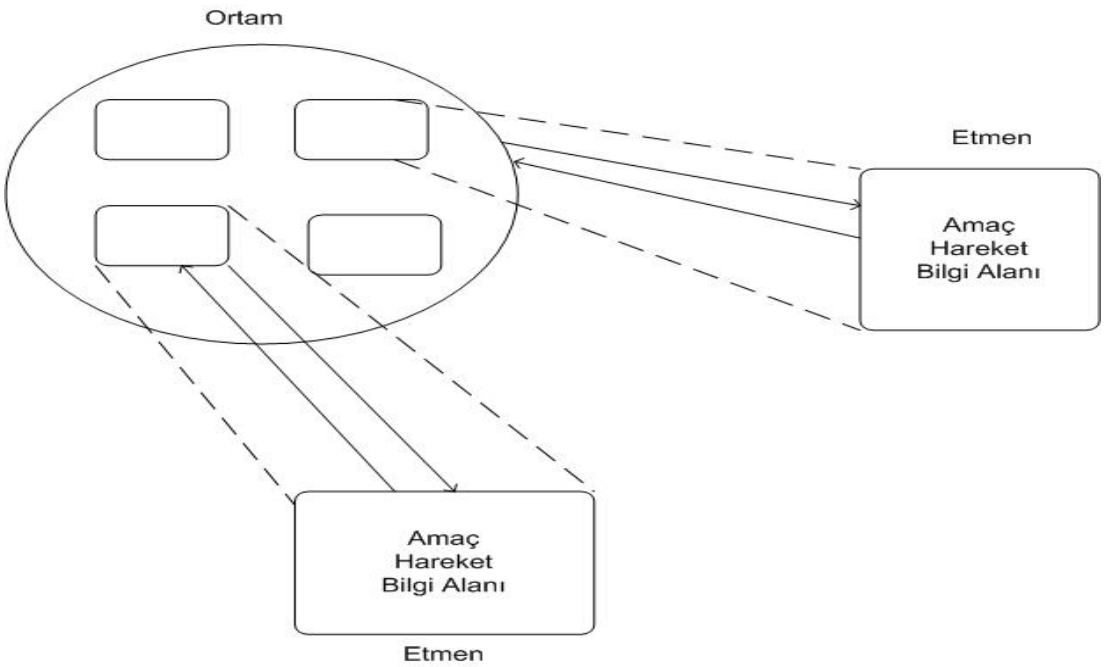
Sahip olunan bilginin sınırlılığı düşünüldüğünde tek bir etmenin her türlü problemi çözmesi beklenemez. Bazı problemlerinin çözümü için çok fazla bilgi gerekmektedir. Ayrıca bazı problemlerin çözümü için farklı uzmanlıkların ortak çalışarak sonuca ulaşması gerekmektedir. Bu ise uzmanlık alanı sınırlı bir etmen ile başarılamaz. Bunlara ek olarak günlük hayatta karşılaşılan çoğu problem doğası gereği dağıtık olmaktadır. Bu problemleri çözmek için açıktır ki çoklu etmen içeren sistemler kurmamız gerekmektedir.

Çoklu Etmen Sistemleri (ÇES) bir problemi çözmek için etkileşen gevşek bağlı problem çözücüler ağıdır ki ortaya çıkan sistem problem çözücülerin bireysel yetenekleri ya da her birinin bilgileri toplamından daha fazladır[2]. Şekil 2.2 ve Şekil 2.3'de de görüldüğü gibi bu problem çözücülerin her biri birer etmendir ve çoklu etmen sistemleri etmenlerin bir araya gelmesi ile oluşmaktadır.

Çoklu etmen sistemi birbiriyle etkileşim halinde ve kolektif olarak amaçlarına ulaşmaya çalışan etmenler grubudur [4]

ÇES birimsel bir yapı sunmaktadır. Problemin büyüklüğü ve gerek duyulan etmen sayısı tahmin edilemeden tasarlanan sistemler yeni gereksinimler doğrultusunda yeni etmenler eklenerek rahat bir şekilde genişletilebilir. Gerek duyulması halinde, ÇES'e belirli uzmanlığa sahip yeni etmenler de eklenebilir. Sistem, kendi uzmanlığı ve bilgisi olan etmenler topluluğundan oluşur.

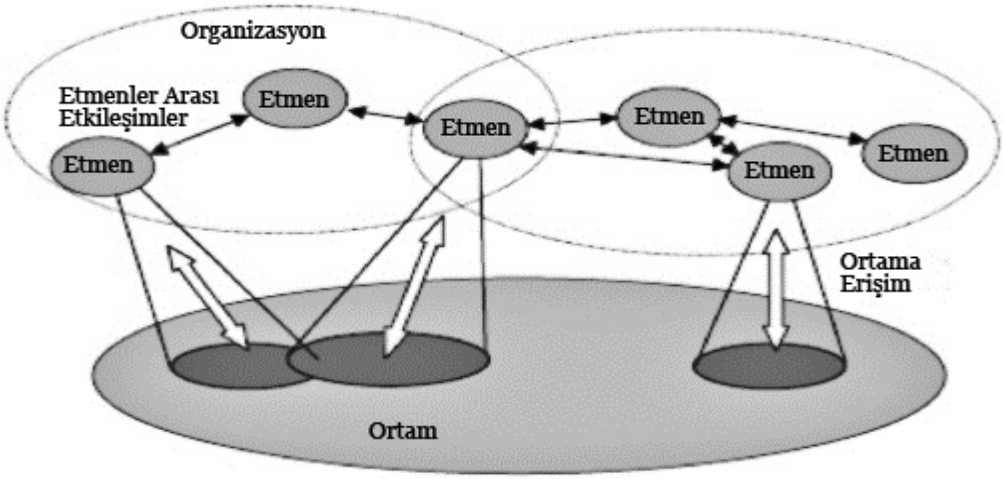
ÇES ile problem çözme işlemi Dağıtık Problem Çözme olarak adlandırılır. ÇES'in problemleri çözebilmesi için etmenlerin kendi aralarında iletişim kurabilmeleri, kendi aktivitelerini yürütürken eşgüdümlü ve pazarlık yapabiliyor olmaları gerekir. Bu üç kavram çoklu etmen sistemleri için çok önemlidir.



Şekil 2.2 : Çoklu etmen sistemi genel yapısı, [2] den uyarlanmıştır.

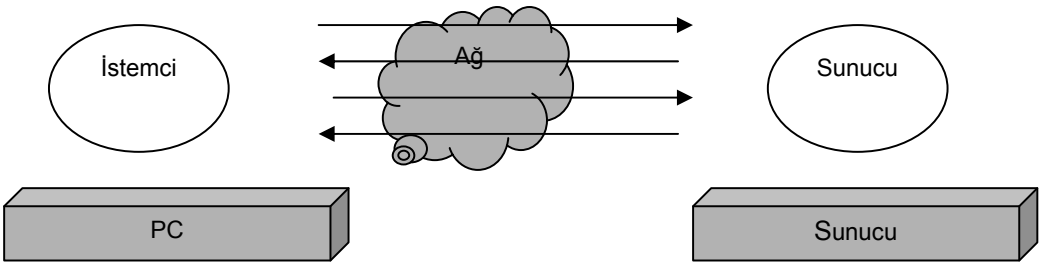
2.1.3 Hareketli etmenler

Hareketli etmen, bir makineden diğer makineye göç edebilen programdır. Program ne zaman ve nereye göç edeceğini kendisi belirler. Etmen çalışmasına belirli bir noktada ara verebilir ve kendisini başka bir makineye taşır ve çalışmasına orada devam eder.

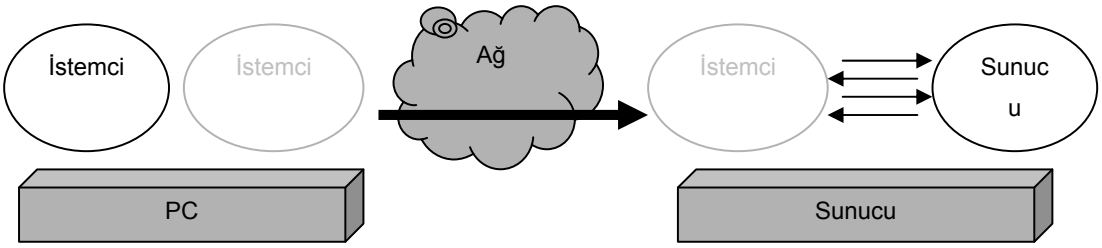


Şekil 2.3 : Çoklu etmen sistemi genel yapısı.

Şekil 2.4'ün ilk bölümünde klasik istemci sunucu modeli, ikinci bölümünde ise hareketli etmen modeli gösterilmektedir. İstemci sunumcu modelinde istemci sunucu ile ağ üzerinden haberleşir. Hareketli etmen modelinde ise istemci kendisini sunucu tarafına göç ettirir ve işlemlerini o tarafta yapar.



Klasik istemci-sunucu modeli



Yeni hareketli etmen hesaplama mantığı

Şekil 2.4 : İstemci-sunucu modeli ile hareketli etmen modeli karşılaştırılması.

Yukarıda verilen tanımı biraz daha genişleterek, hareketli etmenleri şöyle tanımlayabiliriz: hareketli etmen yazılım ortamında yer alan bir yazılım varlığıdır.

Hareketli etmenin içinde bulunduğu ortamı da tanımlamamız gerekir. Bu ortam hareketli etmen ortamı olarak adlandırılabilir. Hareketli etmen ortamı heterojen bilgisayarlar ağları üzerinde dağıtılmış bir yazılımdır ve hareketli etmenlerin çalışabileceği bir çalışma ortamı yaratır.

2.2 Servis Odaklı Hesaplama

Servis Odaklı Hesaplama (SOH) (Service Oriented Computing), uygulama geliştirme için servisleri temel alan bir hesaplama paradigmasıdır [5]. Servis, yerine göre basit bir istek, yerine göre karmaşık bir iş süreci olabilecek fonksiyonları yerine getirir. Servis, kurumlara kendi yeteneklerini standart diller (XML tabanlı) ve protokoller kullanarak internet (veya intranet) üzerinden program şeklinde açmalarını sağlar. Servis açık standartlara dayalı, kendini tanımlayan arayüzler aracılığı ile gerçeklenir.

Servisler, farklı türden hesaplama araçlarına (bilgisayarlar, PDA'lar, cep telefonları) ve yazılım platformlarına (UNIX, Windows) tek şekilde ve her yerden erişebilme imkanı sunduğu için dağıtık hesaplamada önemli bir adımdır diyebiliriz. Servisler servis sağlayıcıları tarafından önerilmektedir. Servisleri gerçekleyen kurumlar servisin tanımlamasını, gerekli teknik ve iş desteğini sağlar. Servisler farklı kurumlar tarafından önerilebilindiği ve internet üzerinden iletişime geçilebildiği için, hem intranet, hem kurumlar arası uygulama entegrasyonuna, hem de işbirliğine uygun dağıtık bir hesaplama altyapısı sağlarlar. Servislerin müşterileri aynı kurum içinden olabileceği gibi, başka kurumlardan diğer çözümler veya uygulamalar da olabilir; dolayısıyla servisler aşağıdaki koşullar sağlamalıdırlar;

- **Teknolojik olarak bağımsız:** servislerin en genel şekliyle standartlaşmış ve bütün BT (Bilgi Teknolojileri) ortamlarında bulunan teknolojiler kullanılarak çağrılıyor olması gerekir. Bu çağırma mekanizmasının (protokoller, tanımlamalar ve bulma mekanizmaları) genel kabul gören standartlarla uyumlu olmasını gerektirir.
- **Gevşek bağlı:** İstemci ya da sunucu tarafının içyapılarını ya da bağlamını bilmeye gerek olmamalıdır.

- **Yer şeffaflığının desteklenmesi:** Servislerin tanımlamaları ve yer bilgileri bir depoda (UDDI (Universal Description, Discovery, and Integration) [6] gibi) depolanmalı ve yerinden bağımsız olarak onları çağıracak olan istemciler bu bilgilere erişebilmeli.

Servisler basit veya bileşik olabilir. Basit servislerin toplamından bileşik servisler oluşur. Örneğin bir kurum, satma ve sevk servislerini bir arada bileşik servis olarak sunabilir.

Servisler iş ile ilgili fonksiyonları kapsarken, servisler arası etkileşimi ve iletişimi sağlamak için altyapıya ihtiyaç vardır. Servisler bir makinede veya dağıtık olarak birden fazla makine üzerinde gerçekleştirilmiş olabilecekleri için farklı altyapı şekillerinde olabilir. URI (Uniform Resource Identifier) ile tanımlanan ve aşağıdaki özellikleri gösteren web servisi servisin özel bir durumudur;

- Özelliklerini standart internet dilleri ve protokolleri kullanarak internet üzerinden açar.
- Açık internet standartlarına (XML arayüzleri gibi) dayalı olarak kendini tanımlayan arayüzler aracılığı ile gerçeklenir.

Etkileşimler XML veri içerikleri taşıyan SOAP (Simple Object Access Protocol) [7] çağrıları şeklinde gerçekleşir. Servis tanımlamaları WSDL (Web Service Definition Language) [8] kullanılarak gerçekleştirilir. WSDL web servisi ile ilgili **ports, port types, bindings** bilgilerini yayınlamak için kullanılır. UDDI standardı servis yayınlarının, web servisi istemcilerinin aday servisleri bulma ve detay bilgilere erişimin yöntemini ifade eder.

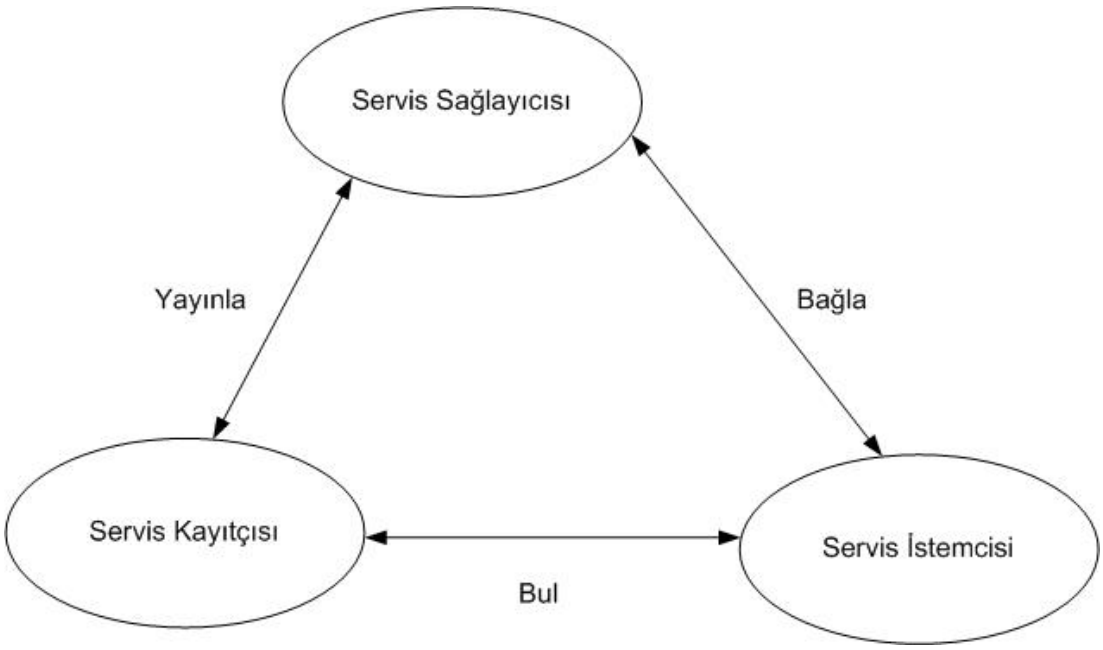
2.2.1 Temel servis odaklı mimari

Entegre olmaya hazır servis modelli uygulamalar oluşturmak servis odaklı mimariye (SOM) dayanır. SOM pek çok servisin bir araya getirilmesi, onların toplamıdır [9].

Etkileşimler XML veri içerikleri taşıyan SOAP çağrıları şeklinde gerçekleşir. Servis tanımlamaları WSDL kullanılarak gerçekleştirilir. WSDL, web servisi ile ilgili bilgileri yayınlamak için kullanılır. UDDI standardı servis yayınlarını içeren ve web servisi istemcilerinin aday servisleri bulmasını ve detay bilgilere erişmesini sağlayan bir dizin servisidir. Bu mimari yaklaşımı özellikle iletişim halinde olması gereken çoklu uygulamaların farklı teknolojiler ve platformlarda çalıştığı durumlara

uygundur. Bu yolla kurumlar en az programlama çabası ile ve servisleri karışık olarak kullanarak işlemlerini yapabilirler.

SOM, son kullanıcı uygulamalarına veya diğer dağıtık servislere bulunabilir ve yayınlanmış arayüzler aracılığı ile servis sağlamak için yazılım tasarlama yöntemidir. Temel SOM servis istemcisi ile sistem sağlayıcısı arasındaki ilişkiyi tanımlar. İstemciler servis almak isteyen yazılımlardır. Sağlayıcılar ise servisi sağlayan yazılımlardır. Sağlayıcılar sağladıkları servislerin tanımlarını yayınlamakla yükümlüdürler. İstemciler istedikleri servislerin tanımlarını bulabilmeli ve bağlanabilmelidirler.



Şekil 2.5 : Temel servis odaklı mimari, [5] ten uyarlanmıştır.

Temel SOM sadece servisler ile ilgili bir mimari değil, aynı zamanda üç tip katılımcı arasındaki ilişkidir: servis sağlayıcı, servis bulan aracı ve servis istemcisi. Şekil 2.5' de görüldüğü gibi etkileşim yayınlama, bulma ve bağlanma işlemleridir. Etkileşimler XML veri içerikleri taşıyan SOAP çağrıları şeklinde gerçekleşir. Tipik bir servis temelli senaryoda, servis sağlayıcısı ağdan erişilebilir yazılım modülüne sahiptir. Servis sağlayıcısı servisin tanımını yapar ve istemciye ya da servis bulucu aracıya yayınlar. Bu işlem servisi bulunabilir kılar. Servis istekçisi arama işlemini kullanarak servis tanımını elde eder. Servis tanımını kullanarak servis sağlayıcısına bağlanır ve servisi çalıştırır.

Servis genelde bir yazılım aracılığı ile gerçeklenen, formel olarak belgelenen bir arayüz ile sarmalmış bir iş fonksiyonudur. Servisler yeniden kullanılabilir; sadece tekil programlarda değil kurumsal uygulamalarda da kullanılabilir. Servisler istemci kanalının ihtiyacına göre bir araya toplanıp daha büyük yeni konfigürasyonlar halini alan anlamlı iş fonksiyonlarını temsil etmek için tasarlanmıştır.

Arayüz basit olarak servislerin diğer uygulamalar ve diğer servisler ile iletişim kurması için bir mekanizma sağlar. Teknik olarak servis arayüzü, istemcinin çağırabileceği servis işlemlerine ait imzaların tanımlamasıdır. Servis açıklaması içinde istemcinin beklediği bütün arayüzler açık olarak tanımlanmalı, ayrıca servisin oluşturulduğu/bir araya getirildiği ortamın sağladığı servis arayüzleri de tanımlanmalı. Servis arayüzü başka servisleri de içerebilir. Yani birleşik bir servis oluşturulabilir.

Servis tanımlamaları servisin yeteneklerini, arayüzünü, davranışlarını ve kalitesini duyurmak için kullanılır. Servis kaydında yer alan servisler hakkında bu bilgilerin yayınlanması ile servisin bulunması, seçilmesi, bağlanması veya yeni bileşik servislerin oluşturulması için gerekli araçları sağlanması mümkün olur. Servis arayüz tanımı servisin imzasının yanı sıra, servisin yeteneklerinin tanımı, kavramsal amacı ve servisten beklenen sonucu da yayınlar. Son olarak, servis kalitesi tanımı, önemli fonksiyonel olan veya olmayan servis kalitesi özellikleri (maliyet, güvenirlik, ölçeklenebilirlik gibi) de yayınlar.

Servisin yerine getirilmesi sırasında fonksiyonları gerçekleştirmek için birden fazla program etkin olabilir. Uygulama bileşimi ve entegrasyonu servisi gerçekleştirmek için kullanılır. Geliştirme aşamasında, servisin nasıl gerçekleneceği, metotların ne olacağı ve gerçeklemenin içyapısı ile ilgilenilir.

Temel SOM'de servis farklı servis istemcileri tarafından çağrılacak şekilde tasarlanır ve servis çağırıcılarına bağlı değildir (gevşek bağlı). Servisler tekrar tekrar kullanılabilir ve istemciler servisin içsel çalışma şeklinden bağımsızdırlar.

2.2.2 Web servis mimarisi

Web servis mimarisi Servis Odaklı Mimarinin gerçeklenmesidir. Web servisleri internet ortamında yer alır ve mesaj değişimleri için internet protokolleri kullanılır. SOM içinde yer alan varlıklar web servisi mimarisi içinde gerçeklenmiştir. Web servisleri arasındaki standart mesajlaşma işlemi Simple Object Access Protocol

(SOAP) [7] tarafından sağlanır. Web servisleri servisin nasıl çağrılacağını gösteren tanım dokümanlarını (Web Service Definition Language (WSDL) [8]) kullanırlar. Servis kayıtçısı ise Universal Description, Discovery ve Integration (UDDI) [6] tarafından gerçeklenir. Web servis kayıtçısı web servis sağlayıcıları için yayın servisi, web servis istemcileri için ise bulma servisi sağlar.

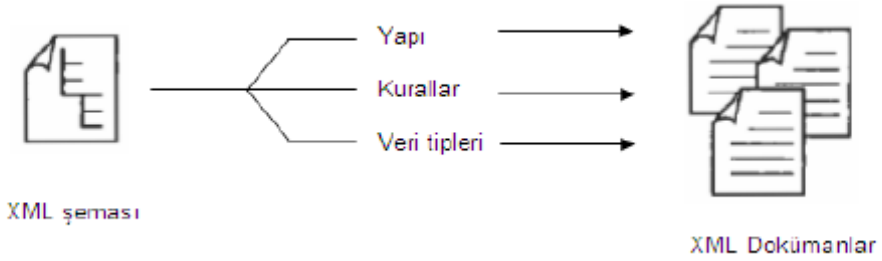
World Wide Web Consortium (W3C) web servislerini şu şekilde tanımlamıştır; "Ağ üzerinde makineden-makineye etkileşimi birlikte çalışır şekilde destekleyen yazılım sistemleridir. Makine tarafından işlenebilen formatta(WSDL) tanımlanmış arayüze sahiptir. Diğer sistemler web servisi ile tanımında yer alan SOAP mesaj formatını kullanarak ile etkileşime geçerler."[10].

Farklı platformlarda web servislerinin birlikte çalışabilirliği internet üzerinden kullandıkları standart iletişim protokolleri (SOAP ve HTTP), standart web servisi tanımlama dili (WSDL) ve standart servis kayıtçısı (UDDI) sayesinde artmaktadır. Web servisleri sayesinde başlangıçta insan ile yazılım arasındaki etkileşim için tanımlanmış olan web, yazılımla yazılım arasındaki etkileşimlerde bilgi taşır hale gelmektedir [11].

2.2.2.1 XML ve özellikleri

XML'in açılımı Genişletilebilir Bağlantılı Metin Dili'dir. XML, bir belgeyi bölümlerine ayıran ve belgenin farklı bölümlerini tanımlayan semantik etiketleri tanımlamada kullanılan bir kurallar dizisidir. Diğer alana özgü bağlantılı metin dillerinin yazılabileceği bir sözdizimini tanımlayan bir meta bağlantılı metin dilidir. Etiketlere gerek duydukça bu etiketlerin oluşturulabildiği bir dildir. Bu etiketler bazı genel ilkelere göre düzenlenmelidir; ancak anlam olarak oldukça esnektirler. Örneğin, soyağacı üzerinde çalışılıyorsa ve soyadların, adların, tarihlerin, doğum tarihlerinin, evlat edinmelerin, ölümlerin, mezarlıkların, ailelerin, evliliklerin, boşanmaların tanımlanması gerekiyorsa, bunların her biri için etiketler oluşturulabilir. Verileri paragraflara, liste öğelerine, tablo hücrelerine ve bazı çok genel kategorilere uyması için zorlamanıza gerek yoktur [12].

XML şemaları XML belgelerinin yapısını tanımlar. Şekil 2.6'da gösterildiği gibi, yapı, doğrulama kuralları, tip kısıtlamaları ve öğeler arası ilişkiler sağlayarak XML belgesinin bütünlüğünü korur. Diğer bir deyişle, XML şemaları XML verileriyle nelerin yapılabileceğini ve nelerin yapılamayacağını belirtir [13].



Şekil 2.6 : XML şemaları ile XML belgeleri arasındaki ilişki.

XML şeması:

- XML belgesinin içeriğini tanımlar ve sınırlandırır
- Özel tiplerin tanımlanmasını sağlar
- Basit tiplerin ve karmaşık tiplerin tanımlanmasını sağlar

XML şeması, programlama dillerinde ve veritabanı uygulamalarında yaygın olarak bulunan çoğu tipi gösteren yaklaşık 46 yerleşik basit tipi tanımlar. XML yazılım alanında yaygın olarak kabul görmekte ve değişik uygulamalarda kullanılmakla birlikte yoğun olarak aşağıdaki alanlarda kullanılmaktadır;

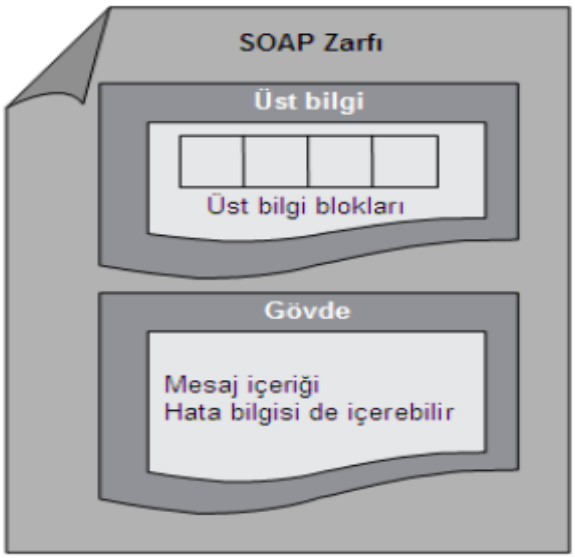
- Veri alış verişi
- Yazılım konfigürasyonu
- Protokol tanımları

2.2.2.2 SOAP (Basit Nesne Erişimi Protokolü)

SOAP yazılım varlıkları arasında internet kullanılarak veri taşımak için kullanılan XML temelli bir protokoldür. SOAP temel olarak yordam çağrımlarındaki istek ve yanıt mesajlarından oluşan XML verisinin taşınması için standart formatı tanımlamaktadır. SOAP mesaj taşıma sistemi standart internet protokollerine dayanmaktadır.

Zarf, üstbilgi ve gövde: Her SOAP mesajı zarf olarak bilinen bir barındırıcı içinde paketlenmiştir. Zarf, mesajın tüm parçalarını barındırmaktan sorumludur.

Her mesaj, meta bilgilerin yerleştirilebileceği bir üstbilgi alanı içerebilir. Çoğu servis odaklı çözümde, bu üstbilgi bölümü genel mimarinin çok önemli bir parçasını oluşturur ve isteğe bağlı bir alan olmasına rağmen genelde kullanılır[14].



Şekil 2.7 : SOAP iletisinin temel yapısı.

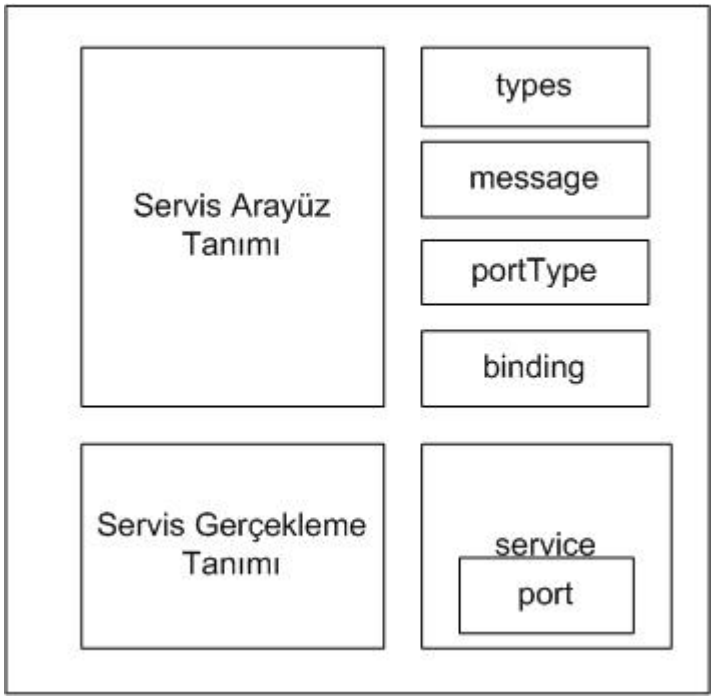
Gerçek mesaj içeriği, genellikle XML biçimlendirilmiş verilerinden oluşan mesaj gövdesi tarafından barındırılır. Mesaj gövdesinin içeriğine genellikle mesajın içeriği adı verilir. Şekil 2.7'de SOAP iletisinin temel yapısı gösterilmiştir.

SOAP mesajları uygulamalar arasında aşağıdaki tipte işlemleri tanımlayabilir;

- Tek yönlü işlemler (çağrılar): Sunucunun sadece SOAP mesajını aldığı ve mesajı işlemeden geri dönüş yaptığı işlemler
- Uyarı işlemleri: Sunucunun mesajı bir veya daha fazla istemciye mesaj talebi olmadan gönderdiği işlemler
- İstek-Yanıt işlemleri: İstemcinin sunucuya istek mesajı gönderdiği ve sunucunun mesajı işledikten sonra yanıt döndüğü işlemler

2.2.2.3 WSDL (Web Servis Tanımlama Dili)

Web servis tanımlama dili (WSDL) web servislerin nasıl çağrılacağını, arayüzü ve gerçekleme detaylarını tanımlayan XML tabanlı bir dildir [8]. Arayüz, web servisinin alacağı ve vereceği mesajı ve parametrelerini tanımlar. Ayrıca gerçekleme, mesajın nasıl işleneceği ve web servise nereden erişileceğini tanımlar. Web servisi belgesi kullanılarak web servis istemcisi web servisini çağırmak için gerekli SOAP mesajını oluşturabilir. WSDL elemanları Şekil 2.8'de gösterilmiştir.



Şekil 2.8 : WSDL elemanları.

PortType elemanı

portType web servisinin sağladığı işlemi tanımlar. Her bir işlem kendi adı, parametrelerinin adı, sırası ve giriş çıkış mesajları tarafından tanımlanır. Örnek mesaj Şekil 2.9'da gösterilmiştir.

```
<wsdl:portType name="unicornManager@AWSMTestPlatform_StudyAbroadRegistration">
        <wsdl:operation name="registerUniversity">
                        <wsdl:input name="registerUniversity-input" message="impl:registerUniversity-input">
                        </wsdl:input>
                        <wsdl:output name="registerUniversity-output" message="impl:registerUniversity-output">
                        </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="findUniversity">
                        <wsdl:input name="findUniversity-input" message="impl:findUniversity-input">
                        </wsdl:input>
                        <wsdl:output name="findUniversity-output" message="impl:findUniversity-output">
                        </wsdl:output>
        </wsdl:operation>
</wsdl:portType>
```

Şekil 2.9 : portType elemanı.

Message elemanı

Mesaj elamanı mesaj formatını tanımlar. Alınan veya gönderilen her mesaj; mesaj adı, parametrelerinin adı ve tipleri ile tanımlanmak durumundadır. Örnek mesaj Şekil 2.10'da gösterilmiştir.

```
<wsdl:message name="registerUniversity-input">
                <wsdl:part name="studyId" type="xsd:string">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="registerUniversity-output">
                <wsdl:part name="studyId" type="xsd:string">
  </wsdl:part>
                <wsdl:part name="countryName" type="xsd:string">
  </wsdl:part>
                <wsdl:part name="message" type="xsd:string">
  </wsdl:part>
                <wsdl:part name="reservationId" type="xsd:string">
  </wsdl:part>
                <wsdl:part name="majorName" type="xsd:string">
  </wsdl:part>
                <wsdl:part name="returnCode" type="xsd:int">
  </wsdl:part>
</wsdl:message>
```

Şekil 2.10 : Message elemanı.

Type elemanı

Type elemanı mesaj içersinde kullanılan karmaşık veri tiplerini tanımlar.

Binding elemanı

Binding elemanı belli bir *portType*'ın özelliklerini tanımlar.

Servis gerçekleme tanımı

Servis elemanı *port* elemanlarını içermektedir. Her bir port elemanı WSDL belgelerinin bulunacağı Web servis bağlama adreslerini içermektedir. Örnek mesaj Şekil 2.11'de gösterilmiştir.

```
<wsdl:service name="unicornManager@AWSMTestPlatform_StudyAbroadRegistration">
        <wsdl:port name="publish" binding="impl:publishSoapBinding">
                <wsdlsoap:address location="http://localhost:85/awsm/ws"/>
        </wsdl:port>
</wsdl:service>
```

Şekil 2.11 : WSDL service ve port elemanı.

2.2.2.4 UDDI

Universal Description, Discovery and Integration (UDDI) servis sağlayıcıları ve onların sağladıkları servisleri depolayan, kategorize eden platform bağımsız bir kayıt sistemidir [6]. Universal Description, Discovery and Integration (UDDI) tanımlaması ve protokolü;

- Mesajların,

- Uygulama programlama arayüzlerinin(API),
- Dağıtık web servisi kayıtçıları için veri yapısının ve bu servisler ile ilgili iş ve teknik bilgiyi depolamasının

tanımlarını belirtir. UDDI, web servis standartlarını ortaya koyan organizasyon olan OASIS [15] standardıdır.

UDDI yüksek seviyeli mimarisi

UDDI http, xml ve SOAP gibi web servislerini ve protokollerini kullanır. UDDI, istemcinin web servisini ve servisin ne yaptığını anlaması için web servisinin tanımlarını bulmasını sağlayan web servisi bulma protokolünü tanımlar. UDDI kayıtçısı, servis için web servisi tanımlama dili (WSDL) dokümanı içine gömülmüş yardımcı veri içerir. UDDI kayıtçısının sağladığı bilgiler ile 3 tip arama yapılabilir;

1. Beyaz sayfa araması: şirket ve servisi hakkında adres, iletişim bilgileri gibi bilgiler döner.
2. Sarı sayfa konu araması: endüstri kategorisine ve taksonomilere göre bilgiler döner.
3. Yeşil sayfa araması: servisin nasıl çalıştırılacağını belirten bilgiler gibi web servisi hakkında teknik bilgiler döner.

UDDI yayın API'si

UDDI yayın API'si yaratma, okuma, güncelleme ve silme işlemlerini içerir. Çağırıcı bu işlemleri iş ve servislerini kayıt etmek veya değiştirmek için kullanır. UDDI kayıtçısı tanımlamalarda belirtilen yayın API'leri çağrısını desteklemeli.

UDDI kimlik denetleme modeli

UDDI yayın API'sinin anahtar prensibi sadece yetkili olanların UDDI iş kayıtçısındaki bilgileri yayınlamasına ve değiştirmesine izin vermesidir. UDDI iş kayıtçıları yetkili tarafların listesini ve kimlerin *businessEntity* ve *tModel'leri* yarattığını tutar. Daha sonra değişiklik ve silme isteklerine ancak bilgiyi yaratan tarafından yapılmışsa izin verirler.

UDDI data model

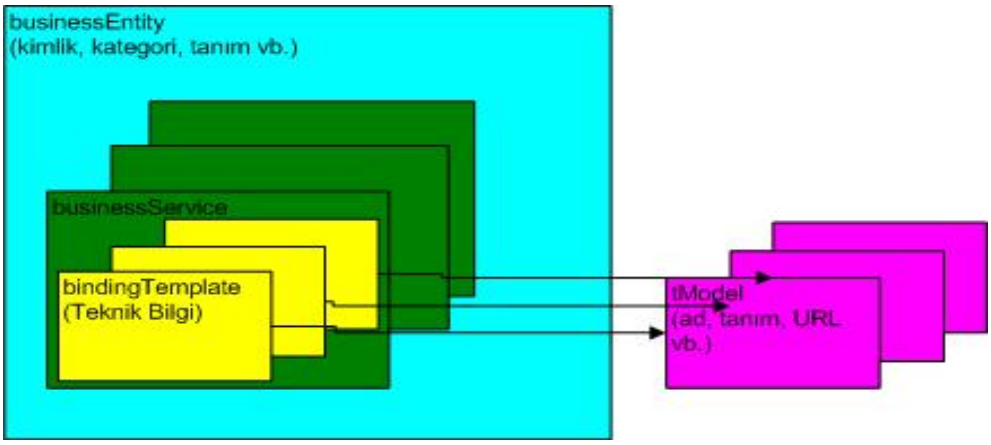
Kayıtçıların kendisi web servisi olarak gerçeklenir. Tipik bir UDDI kayıtçısı aşağıdaki bileşenler çerçevesinde bir etkileşimi destekler;

- Kayıtçı istemcileri: kayıtlı iş ve servisleri aramak ve göz atmak için kayıtçı ile etkileşim halinde olan programlar veya servislerdir.
- Kayıtçı girişleri: kayıtçı içersinde yer alan iş veya servis hakkındaki içeriktir.
- Giriş yardımcı veri: kayıtçı girişini ve kayıtçı girişinin nasıl bulunması gerektiğini tanımlayan içeriktir.

Kayıtçı girişleri ve yardımcı veri aşağıdaki yapıda ifşa edilir;

- **businessEntity**: bu yapı bir veya daha fazla iş servisini tanımlayan isim, kategori, iletişim bilgileri gibi bilgileri sarar.
- **businessService:** bu yapı bir biriyle ilişkili, grup halindeki teknik servislerin grup adı, kategori gibi tanımlayıcı bilgilerini içerir.
- **bindingTemplate:** bu yapı belirli bir servisi bağlama ve çalıştırma için gerekli bilgileri içerir. Gerekli bilgiler: servisin URL'si, yönlendirmesi, *tModel* içerisinde bulunan arayüz tanımlarına referanslar gibi bilgiler olabilir.
- **tModel:** bu yapı belirli bir servisin arayüzleri ve diğer teknik bilgilerini içerir.

Bu yapılar arasındaki ilişkiler Şekil 2.12.'da gösterilmiştir.



Şekil 2.12 : UDDI yapıları arasındaki ilişki.

UDDI çağırma modeli

Belirli bir web servisini UDDI kayıtçısındaki bilgileri kullanarak çağırmak için çağıran aşağıdaki adımları takip eder;

1. Çağıran, web servisinin işini açıklamak için kayıt ettiği *businessEntity* bilgisini alır.

2. **businessEntity** yapısının içinde yer alan **businessService** yapısına erişerek web servisi hakkında detaylı bilgiyi bulur. Çağırıcı kullanmak için uygun **bindingTemplate'**i seçer.

3. Seçilen **bindingTemplate** karşılık gelen **tModel** içinde var olan teknik bilgi kullanılarak web servisine erişecek istemci oluşturulur ve servis çağırısı gerçekleştirilir.

UDDI sorgulama API'si

UDDI sorgulama API'si belli iş ve servis bilgilerini edinmek için kayıtçıya göz atmayı ve kayıtçıyı gezinmeyi sağlayan işlemlerden oluşur. Bu api'ler sayesinde çağırma modelinde belirtilen işlemler gerçekleştirilebilir.

2.2.3 Servis odaklı mimari ve etmenler

MAS Research Roadmap Project [16] grubunun ortak çalışmasında belirttiği gibi, bugün servis odaklı mimariler tekil kitap rezervasyonu servisinden, işlem bütünlüğü gerektiren otel rezervasyonu ve uçak bileti sağlayan bileşik servislere kadar pek çok alanda kullanılmaktadır. Servis odaklı hesaplamanın içinde olan Web servislerinin kullanımı hızlı bir şekilde gelişmektedir. Ancak vurgu tekil servisin çalışmasına yapılmakta, servisin ne şekilde seçileceği ve işbirliğinin nasıl gerçekleneceği üzerinde pek durulmamaktadır. Eksik olan noktalara baktığımızda bunlar, çoklu etmen sistemlerinin temel özelliği olan otonomi, işbirliği ve proaktiflik olduğunu görebiliriz. Bir benzetme yapmak gerekirse, etmen sistemini beyin, servisleri ise kas olarak değerlendirebiliriz. Literatürde, etmen sistemleri ile servislerin beraber kullanıldığı bir dizi sistem yer almaktadır. Aşağıda bunlardan bir kısmına değinilecektir.

Blake [17] çalışmasında, servislerin yönetilmesi ve ortaklaşa çağrılabilmeleri için etmen sistemin kullanılmasını önermiştir. Etmenler farklı organizasyonların dahil olduğu iş akış sisteminde broker olarak kullanılmıştır. Etmenler ortam bilgisine sahip otonom yazılım varlıkları olarak tanımlanmış, servislerin çalışmasına ve iş yönetimine aktif olarak vekillik etmişlerdir. Ayrıca etmenlere belirli bir servisi ve iş sürecini yürütmeleri için görev verilmiştir.

Li, Genniwab ve Shen [18] ise çalışmalarında etmenler ve web-servislerini beraber kullanmışlardır. Çoklu etmen sisteminin bütünleşik servisleri sunmak için iyi bir

yöntem olduğunu ileri sürmüşlerdir. Web servisi odaklı bir siber pazaryeri, servis odaklı etmen sistemleri olarak tasarlanmıştır. Servis odaklı etmen sistemi işbirliği halinde ya da yarışarak aracılık, fiyatlandırma ve pazarlık etme gibi ekonomik servisleri ya da kurumlar arası entegrasyonu desteklemektedir. Pazarın temel elemanları servislerdir. Yazılım etmenleri, servisleri fonksiyonlar olarak gerçeklerler.

Xue, Chen, Zhao, Wang[56] çalışmalarında etmenleri ve web servislerini kullanarak, dağıtık ve heterojen sistemlerde bilgi paylaşımını ve bilgi taşımasını sağlamak için bilgi platformu oluşturmuştur. Sistem içersinde servisler veri kaynağı olarak değerlendirilmiştir. Servisler fonksiyonlarına göre değil sağladıkları veriye göre belirlenmiştir. Sistemde etmenler kullanıcı ile etkileşimi sağlamadan, kullanıcının ihtiyacını analiz etmeden, kullanıcının ihtiyacına göre uygun web servislerini çağırmadan ve web servis çağrı sonuçlarını kullanıcıya dönmeden sorumludur.

Brazier, Kephart, Van Dyke Parunak, Huhns[57] gün geçtikçe karmaşıklaşan bilgi işlem ortamlarını incelemişlerdir. Bilgi işlem ortamlarının veritabanları, uygulama sunucuları, yük dengeleyiciler gibi farklı farklı sistemlerin kullanılmasıyla, sadece kişiler tarafından yönetilmesinin zorlaştığını ifade etmişlerdir. Bu karmaşıklığın üstüne web servislerinin kullanımı ile bilgi işlem ortamlarına çok farklı kurumlar taraf olmuş ve karmaşıklığı arttırmıştır. Ortaya konan çalışmada, bu sistemlerin bilgisayarlar tarafından yönetilmesi için önerilen kendi kendine konfigürasyon, düzeltme, optimizasyon yapabilen otonom bilgi işlem sistemlerinin gerçeklenebilmesinin ancak etmenler ve servislerin beraber kullanılması ile olabileceği öne sürülmüştür.

Herrera, Bepperling, Lobov, Smit, Colombo, Lastra[58] karmaşık ve heterojen üretim sistemlerini yönetmek için çoklu etmen sistemleri ve web servislerinin beraber kullanıldığı bir fabrika otomasyon sistemi önermiştir. Üretim sistemleri çok fazla heterojen sistemden oluşmaktadır. Bu sistemlerde aletlerin fonksiyonları servis olarak tanımlanabilir. Böylece sistemlerin haberleşmesi web servisleri üzerinden sağlanabilir. Sistemin kontrol mimarisi otonom ve akıllı olan etmenler üzerine kurgulanmıştır. Sistem içersinde, web servislerinin heterojen ortamlardaki etkinliği ile etmenlerin akıllı ve otonom olma özelliklerinden faydalanılmıştır.

Vall, Ramparany, Vercouter [21] etmenleri ve web servislerini akıllı telefonlarda dağıtık servis oluşturmak için kullanmıştırlar. Semantik servis tanımları, servis

fonksiyonlarını tanımlamak için kullanılmıştır. Etmenler bu tanımları servis oluşturmak için kullanmıştır.

2.3 Roller

Bu bölümde yazılım uygulamaları bağlamında rol kavramı hakkında bilgi verilecektir.

2.3.1 Rol nedir?

Davranış biliminde rol, kişinin içinde bulunduğu ortamda kendi konumunun kişiye verdiği yetki ile yapması beklenen, tanımlanmış davranış örüntüsüdür [22]. Organizasyonlarda bireyler özel bir konumda bulunurlar. Yer aldıkları konum belirli bir koltuğu temsil eder ve bu koltuk kişiye belirli hak ve sorumluluklar sunar. Roller, yetenekler ve beklenen davranışlar kümesi olarak tanımlanabilirler.

Yazılım bağlamında ise rol, nesne davranışlarının, nesne etkileşimlerinin alt kümesi ile birlikte sınırlamalar kümesini de kapsayacak şekilde kavramsallaştırılmasıdır [23]. Rol ayrıca erişim ve işlemler için gerekli izinler kümesini, verilmiş izinler kümesini, yönlendirilmiş servis yürütme çizgesini ve durum bilgisini kapsayan bir varlık olarak da tanımlanabilir.

2.3.1.1 Rol tabanlı erişim kontrolü (RTEK)

Rollerin yetkiler ve izinler konusunu iyi bir şekilde kavramsallaştırıyor olması güvenlik yönetimi ve erişim konusunda öncelikle kullanılmasına yol açmıştır. 1970'lerden beri RTEK (Rol Tabanlı Erişim Kontrolü) kavramı güvenlik ve erişim alanlarında kullanılmaktadır [24]. RTEK'in temel felsefesi erişim ve işlem haklarının rollerle ilişkilendirilmesi ve daha sonra uygun rollerin kullanıcılara atanmasıdır. Rollerin temsil ettiği organizasyon yapısının kullanıcılara göre çok daha az değişmesi sebebi ile yetkilerin rollere verilmesi kullanıcılara verilmesinden çok daha avantajlıdır. İhtiyaç duyulması durumunda kolayca rollere yeni yetkiler verilebilir ya da alınabilir. Aynı şekilde kullanıcılara rolleri atamak veya kullanıcıdan rolleri almak çok daha kolaydır.

Yukarda belirtilen özelliklere ek olarak RTEK'e farklı yetenekler eklenmiştir. Roller ile roller arasında, roller ile izinler, roller ile kullanıcılar arasında ilişkiler kurulmuştur. Roller miras yöntemi ile ilişkide olduğu rollerden farklı yetkileri

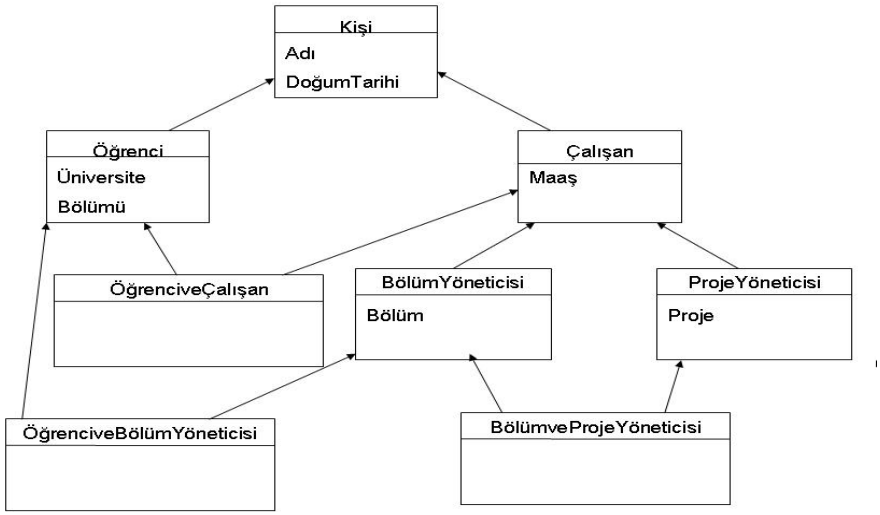
alabilir, aynı yetki birden fazla rolde olamaz, bir kullanıcı aynı anda belirli iki rolü alamaz gibi özellikler RTEK'e eklenmiştir.

Kolay ve etkili işleyen bir RTEK sisteminde aşağıdaki iki özelliğin bulunması gerekmektedir:

- Rol sahipliğinin ve rol yetkilerinin kolay belirleniyor olması
- Rol sahipliğinin ve rol yetkilerinin merkezi ve bir kaç kullanıcı tarafından kontrol edilebilmesi

2.3.1.2 Nesneye yönelik programlamada rol

Programlamada rol kavramının kullanılması 1977'lere kadar gitmektedir [25]. Nesneye yönelik programlara rol özelliğinin uygulanmasına 1980'lerin başından itibaren başlanmış ve pek çok çalışma yapılmıştır.



Şekil 2.13 : Sınıf hiyerarşisi.

Sınıf tabanlı olan nesneye yönelik programlarda bir nesne ve ait olduğu sınıf arasındaki ilişki kalıcı ve durağandır [26]. Sürekli değişen ve evrimleşen gerçek dünya nesnelerinde bu durum bir sorun yaratmaktadır. Günlük yaşamda nesneler sürekli yeni rollere bürünmekte ve davranışları dinamik olarak değişmektedir. Sınıflar arası yapı ve davranış paylaşımından dolayı oluşturulan sınıf hiyerarşisi belirlendikten ve gerçekleme tamamlandıktan sonra kolay değişmemektedir [27]. Sınıf hiyerarşisi belirlenirken farklı davranış özellikleri gösteren farklı sınıflar ortaya çıkmakta, sınıflar arası ilişkiler ve sınıf hiyerarşisi sürekli büyümektedir. Şekil 2.13'de görülen sınıf hiyerarşisinde yeni sınıflar eklendikçe hiyerarşi gittikçe

büyüyecektir. Ayrıca şekle göre yaratılan nesnenin değişmesi ve farklı davranışları göstermesi mümkün olamamaktadır.

Şekil 2.13'deki sınıf hiyerarşisine bakıldığı zaman yaratılan sınıfların bir kısmının aslında rolleri ifade ettiği görülmektedir. Örneğin proje yöneticisi bir çalışanın rolünü ifade etmektedir. Proje yöneticisi olarak işe başlayan çalışan proje yöneticiliğini bırakıp bölüm yöneticisi olabilir ya da ikisini de aynı anda olabilir. Bu gibi durumlarda roller iyi bir çözüm olarak görülmektedir.

2.4 JAVA Dili ve Katkıları

Java dili etmen sistemlerinde yoğun olarak kullanılmaktadır. Bu tezin geliştirilmesinde de kullanılmış olan JADE [43] çatısı java ile geliştirilmiştir. Java dilinin normal özellikleri yanı sıra aşağıdaki 2 temel özelliği bu tez içinde kullanılmış ve uygulamaya temel katkılar sunmuştur. Bu bölümde bunları detayı aktarılacaktır.

- Dinamik nesne yaratmak ve yordam çağırmak için kullanılan yansıma (Reflection)
- Sınıflar, yordamlar ve parametre bilgilerinin çalışma zamanına özellikleri ile taşınmasını sağlayan etiketler (Annotation)

2.4.1 Yansıma

Java Yansıma (Reflection), JVM tarafından yüklenmiş olan sınıflar, arayüzler, sınıfların statik alanları, yordam ve nitelikleri hakkında bilgilere çalışma zamanında programsal olarak erişmemizi sağlayan sınıf ve arayüzleri içeren bir kütüphanedir [44]. Yansıma kullanarak bir nesnenin hangi sınıftan türediğini öğrenebilir, bir sınıfın yapılandırıcılarına, metotlarına, niteliklerine erişebilir, çalışma zamanında bir sınıf yaratabilir, bir sınıfın çalışma zamanına kadar ismini bilmediğimiz yordamlarını çağırabilir, boyunu çalışma zamanında belirleyebildiğimiz diziler yaratabilir, elimize çalışma zamanında gelen sınıflar üzerinden nesneler yaratabiliriz.

Yansıma'nın nasıl çalıştığını gösteren basit bir örnek;

```
import java.lang.reflect.*;

   public class DumpMethods {
      public static void main(String args[])
      {
```

```
         try {
            Class c = Class.forName(args[0]);
            Method m[] = c.getDeclaredMethods();
            for (int i = 0; i < m.length; i++)
            System.out.println(m[i].toString());
         }
         catch (Throwable e) {
            System.err.println(e);
         }
      }

   }
```

Program çıktısı:

```
public java.lang.Object java.util.Stack.push(
    java.lang.Object)
   public synchronized
     java.lang.Object java.util.Stack.pop()
   public synchronized
      java.lang.Object java.util.Stack.peek()
   public boolean java.util.Stack.empty()
   public synchronized

     int java.util.Stack.search(java.lang.Object)
```

Sınıfın yordamlarını dinamik olarak bulabilir ve çağırabiliriz.

```
import java.lang.reflect.*;

   public class method1 {
      private int f1(
       Object p, int x) throws NullPointerException
      {
         if (p == null)
            throw new NullPointerException();
         return x;
      }

      public static void main(String args[])
      {
         try {
           Class cls = Class.forName("method1");

            Method methlist[]
              = cls.getDeclaredMethods();
            for (int i = 0; i < methlist.length;
               i++) {
               Method m = methlist[i];
               System.out.println("name
                 = " + m.getName());
               System.out.println("decl class = " +
                              m.getDeclaringClass());
               Class pvec[] = m.getParameterTypes();
               for (int j = 0; j < pvec.length; j++)
                  System.out.println("
                   param #" + j + " " + pvec[j]);
               Class evec[] = m.getExceptionTypes();
               for (int j = 0; j < evec.length; j++)
                  System.out.println("exc #" + j
                    + " " + evec[j]);
```

```
                System.out.println("return type = " +
                                    m.getReturnType());
                System.out.println("-----");
            }
        }
        catch (Throwable e) {
            System.err.println(e);
        }
    }

}
```

Uygulama çıktısı aşağıdaki gibidir;

```
 name = f1
  decl class = class method1
  param #0 class java.lang.Object
  param #1 int
  exc #0 class java.lang.NullPointerException
  return type = int
  -----
  name = main
  decl class = class method1
  param #0 class [Ljava.lang.String;
  return type = void
  -----
```

2.4.2 Etiketleme

Etiket java'da java elemanlarına eklenerek elemanlar hakkında meta bilgi sunan yapıdır [45]. Sınıflar ve Arayüzler gibi Etiketler Java tiplerini tanımlar ve pek çok Java elemanlarına uygulanabilir. Etiketler derleme zamanında ön derleme araçları tarafından veya çalışma zamanında java yansıma kullanılarak işlenebilir. Aşağıda örnek bir etiketleme yer almaktadır.

```
@MyAnnotation(name="someName",  value = "Hello World")
public class TheClass {
}
```

TheClass @MyAnnotation sınıfı tarafından etiketlenmiştir. Etiketler arayüz gibi tanımlanmaktadır. Aşağıda MyAnnotation etiketleme tanımı yer almaktadır.

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)

public @interface MyAnnotation {
    public String name();
    public String value();
}
```

`@Retention(RetentionPolicy.RUNTIME)` etiketin çalışma zamanında yansıma ile erişilebilineceğini göstermektedir. Eğer bu direktif yer almazsa etiketleme çalışma zamanında var olmaz. Eğer bu ifade varsa etiketleme bilgisi çalışma zamanına taşınır.

`@Target(ElementType.TYPE)` etiketin hedefi olan tipleri belirtir. Etiketlenebilecek tipler aşağıdaki gibidir;

- TYPE : Sadece tipleri uygulanır. Tipler java sınıfları, arayüzleri veya etiket bile olabilir.
- FIELD: Sadece java alanlarına uygulanır. Alanlar nesneler, değişkenler olabilir.
- METHOD: Sadece yordamlara uygulanır.
- PARAMETER: Sadece yordma parametresine tanımına uygulanır.
- CONSTRUCTOR: Sadece sınıf yapıcısına uygulanır.
- LOCAL_VARIABLE: Sadece yerel değişkenlere uygulanabilir.
- ANNOTATION_TYPE: Sadece etiket tiplerine uygulanır.
- PACKAGE : Sadece paketlere uygulanabilir.

Tez boyunca etiketleme ve yansıma yoğun olarak kullanılmıştır. Servis ve rol sınıf tanımları, yordam ve parametre tanımları için etiketleme kullanılmıştır. Bu sayede dinamik olarak rollerin dağıtılması ve roller üzerinden servislerin çıkarılması sağlanmıştır. Rol sınıfları yansıma özelliği kullanılarak dinamik olarak yaratılmıştır. Ayrıca servis çağrıları sonrasında uygun metotlar, uygun parametreler kullanılarak yansıma sayesinde kolay bir şekilde çağrılmıştır.

2.5 JADE Etmen Çatısı

JADE (Java Agent DEvelopment Framework) Java dili ile gerçeklenmiş etmen tabanlı bir yazılım çatısıdır [46]. JADE'in asıl amacı ÇES'lerin ve FIPA (Foundation for Intelligent Physical Agents) uyumlu akıllı etmen uygulamalarının geliştirilmesi için uygun altyapıyı sağlamaktır. JADE temel iki ana ürün içermektedir:

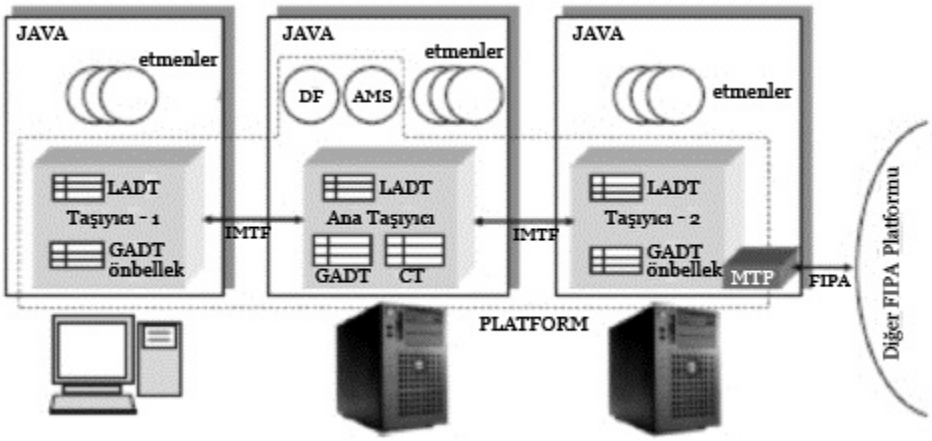
- FIPA uyumlu etmen platformu,
- Java etmenleri üretilebilmesine olanak sağlayan yazılım uygulamasıdır.

JADE tamamen Java programlama dili ile kodlanmıştır. JADE'i kullanmak isteyen bir programcı etmenlerini mutlaka Java dili ile kodlamak zorundadır. JADE'in geliştirmesine temel olan prensipler [47];

- Birlikte işlerlik: JADE FIPA uyumlu olduğu için FIPA uyumlu diğer etmenler ile beraber çalışabilir.
- Bütünlük ve taşınabilirlik: JADE Java versiyonundan ve alttaki ağdan bağımsız olarak homojen bir API kümesi sağlamaktadır. JADE çalışma zamanı hem normal hem de mobil java için tek bir API kümesi sağlamaktadır.
- Kolay kullanılırlık: Çatının karmaşıklığı API kümesinin arkasına gizlenerek kolay kullanım sağlanmıştır. Programcılar çatının sağladığı bütün özellikleri öğrenmek ve kullanmak durumunda değildir. Programcılar sadece istedikleri özellikleri öğrenip kullanabilirler.

2.5.1 JADE mimarisi

Şekil 2.14'de JADE çatısının ana mimari elemanları gösterilmektedir. JADE çatısı ağ üzerine dağıtılmış etmen taşıyıcılardan oluşmaktadır. Etmen taşıyıcısı JADE çalışma zamanı sunan ve etmeni konaklatmak ve çalıştırmak için gerekli servisleri sağlayan Java prosesidir. Etmenler taşıyıcılar içinde yaşarlar. Ana taşıyıcı en başta çalışan ve diğer taşıyıcıların kayıt olarak kendisine katıldığı özel bir taşıyıcıdır ve platformun önyükleyicisidir. Şekil 2.15'de JADE ana elemanları arasındaki ilişkiyi UML diyagramı olarak göstermektedir.



Şekil 2.14 : JADE mimarisi, [46] dan uyarlanmıştır.

Önyükleyici olarak ana taşıyıcının aşağıdaki sorumlulukları bulunmaktadır;

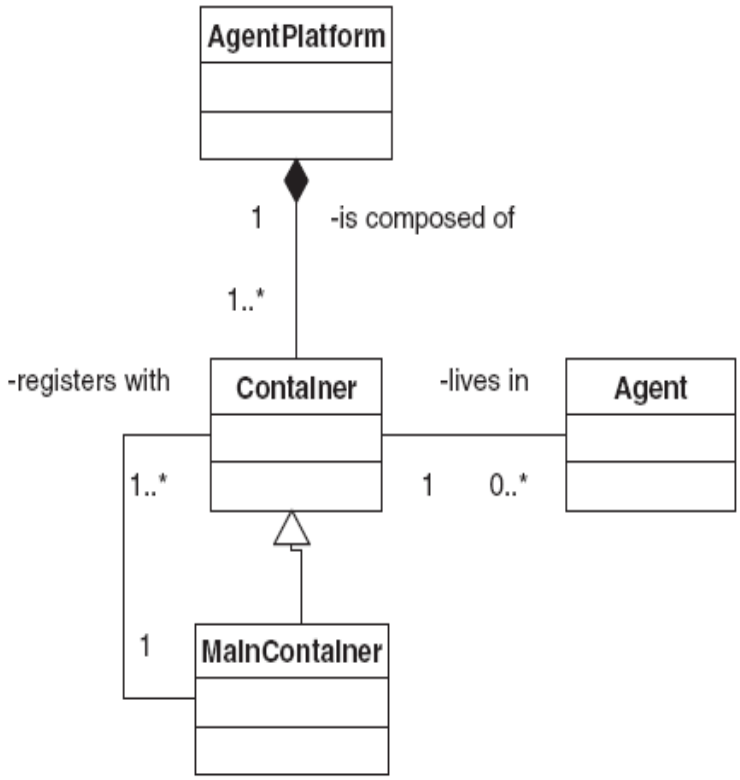
- Platformu oluşturan bütün taşıyıcıların nesne referanslarının ve taşıma adreslerinin kayıt edildiği taşıyıcı tablosunu (container table CT) yönetmek

- Platformda bulunan bütün etmenlerin, statülerinin ve yerlerinin kayıt edildiği global etmen tanımı tablosunu (global agent descriptor table (GADT), yönetmek
- Etmen yönetimini ve beyaz sayfa servisini yapan AMS ve sarı sayfa servisini sağlayan DF'i konaklatmak

Taşıyıcı herhangi bir mesaj için alıcı adresini tespit etmek için öncelikle yerel etmen tanım tablosunu (LADT (local agent descriptor table) tarar. Burada etmeni bulamaz ise uygun referans bilgisini bulmak için ana taşıyıcı ile irtibata geçer ve bu değeri daha sonraki kullanımlar için tamponlar.

Tablolar için tamponlama işlemi hızlı erişim için yapılmaktadır.

Her bir platform adresi mesaj taşıma protokolü (MTP Message Transport Protocol) sonlanma noktasına karşılık gelmektedir. FIPA uyumlu mesajlar bu sonlanma noktasından alınır ve gönderilir. Programcılar isterlerse kendi özel MTP'lerini gerçekleyebilirler.



Şekil 2.15 : JADE mimari UML diyagramı, [46] dan uyarlanmıştır.

Ana taşıyıcı başlar başlamaz iki adet özel etmeni otomatik olarak çalıştırır.

- Agent Management System (AMS): Bütün platformu yöneten etmendir. Etmenler beyaz sayfa servisine erişmek için AMS ile iletişim içinde olmak zorundadır. Bütün etmenler geçerli bir etmen belirleyicisi almak için AMS'ye kayıt olmalıdır.
- DF Directory Facilitator: Etmenler için sarı sayfa servisini gerçekleyen etmendir. Servislerini kayıt ettirmek ve servis aramak isteyen etmenler tarafından kullanılır.

2.5.2 JADE paketleri

JADE Java programlama dili ile yazılmış olan birçok Java Paketinden oluşmaktadır. Uygulama geliştiricilere hazır bir takım fonksiyonlar sunduğu gibi geliştirilmek istenen özel uygulamalar için çok sayıda soyut arayüz de sunmaktadır. JADE için Java programlama dilinin seçilmesindeki en büyük etken bu dile ait çok gelişmiş özelliklerin etmenler için çok uygun olmasıdır. Bu özellikler arasında nesneye yönelik bir dil olması, farklı platformlarda çalışabilmesi, nesne serileştirmeye olanak sağlaması ve RMI (Remote Method Invocation) sayılabilir.

JADE aşağıda yer alan ana paketlerin bir araya gelmesi ile oluşmuştur: Jade.Core paketi sistemin çekirdeğini oluşturmaktadır. Uygulama geliştiricilerin kendi etmenlerini oluşturmak üzere kullandıkları Agent sınıfı bu paket içersinde yer almaktadır. Etmen davranışlarının belirlenmesinde kullanılan Behaviour sınıfı da yine bu paket içersindeki Jade.Core.Behaviours alt paketi içersinde yer almaktadır. Bir etmenden beklenenler ve etmenin yerine getirmesi gereken işler davranışlar ile gerçekleştirilir. JADE'de davranışlar birer mantıksal iş parçacığı olarak görülür. Davranışlar karmaşık görevleri yerine getirmek için eşzamanlı olarak işletilebilirler. Uygulama geliştiriciler yapılmasını istedikleri işleri davranış olarak gerçekleştirir ve bu davranışları işleme sırasını oluştururlar.

Jade.Lang.Acl alt paketi FIPA standartlarına uygun olacak şekilde ACL (Agent Communication Language) mesajlarının işletilmesini sağlar.

Jade.Content paketi kullanıcı tanımlı içerik-dillerinin ve yine kullanıcı tanımlı ontolojilerin işletilebilmesine olanak sağlar.

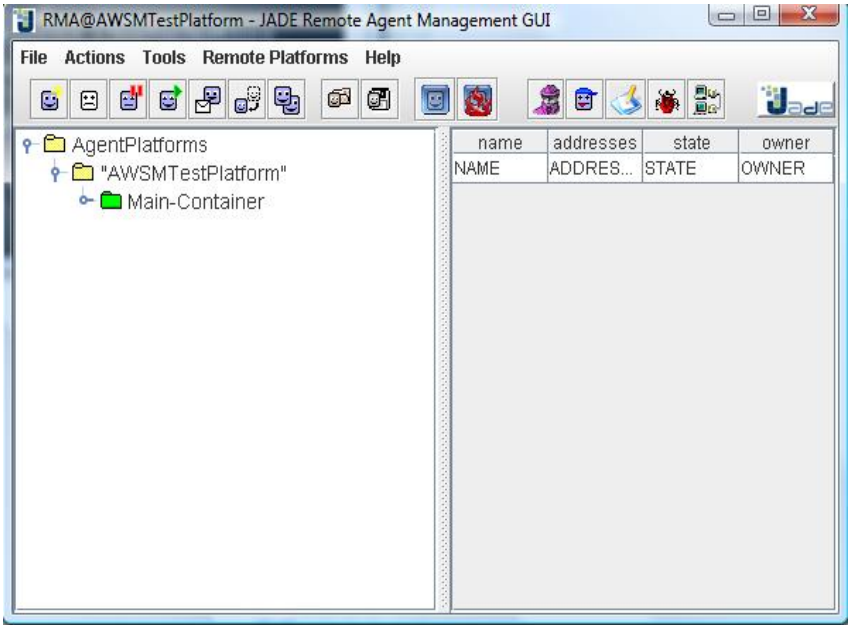
Jade.Domain paketi FIPA tarafından tanımlanan etmen yönetimi ile ilgili AMS ve DF gibi varlıkları içermektedir.

Jade.Gui paketi etmenlerin görüntülenmesi ve tanımlayıcı özelliklerinin değiştirilebilmesi, ACL mesajlarının içeriklerinin görüntülenmesi ve değiştirilebilmesi gibi işlerde çok yararlı olan grafiksel kullanıcı ara yüzlerinin oluşturulabilmesi için genel sınıflar içermektedir. Jade.Mtp paketi etmen çerçeveleri arasındaki etkileşimi sağlamak üzere bir takım sınıflar içermektedir.

2.5.3 JADE araçları

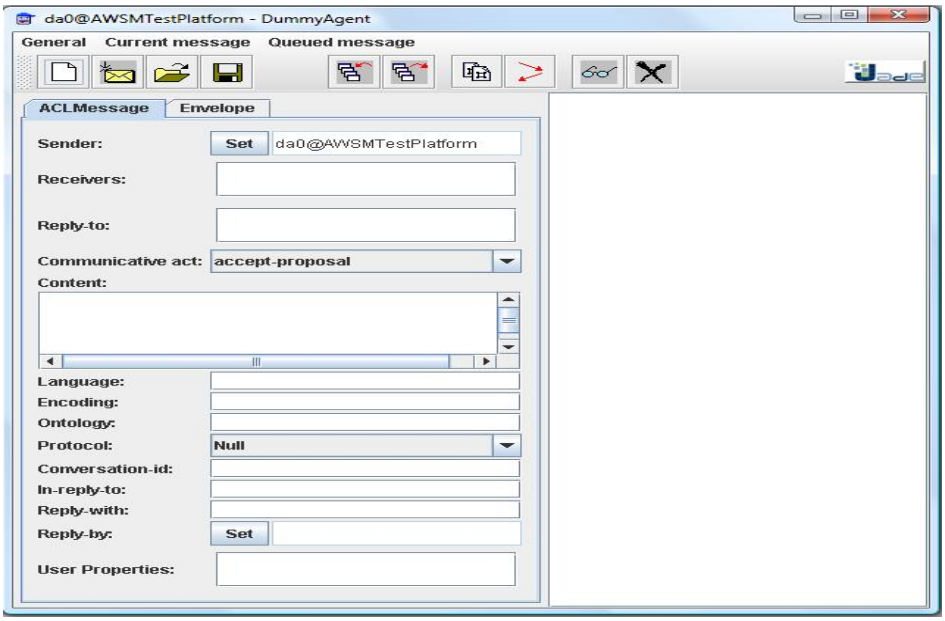
Platform yönetiminin kolayca yapılabilmesi ve uygulamaların geliştirilmesine destek vermek üzere JADE bazı yararlı araçları da bünyesine almıştır.

RMA (Remote Management Agent) bir veya daha fazla taşıyıcı düğümü olan ve farklı makinelerde konaklayabilen platformun yönetilebilmesi ve kontrolü için grafiksel arayüz sağlar. RMA'nın menüsü aracılığı ile diğer sistem araçları da çalıştırılabilir. RMS başlama sırasında platform seviyesindeki bütün olaylardan haberdar olmak için AMS'ye abone olur. Şekil 2.16'da RMA arayüzü gösterilmiştir.



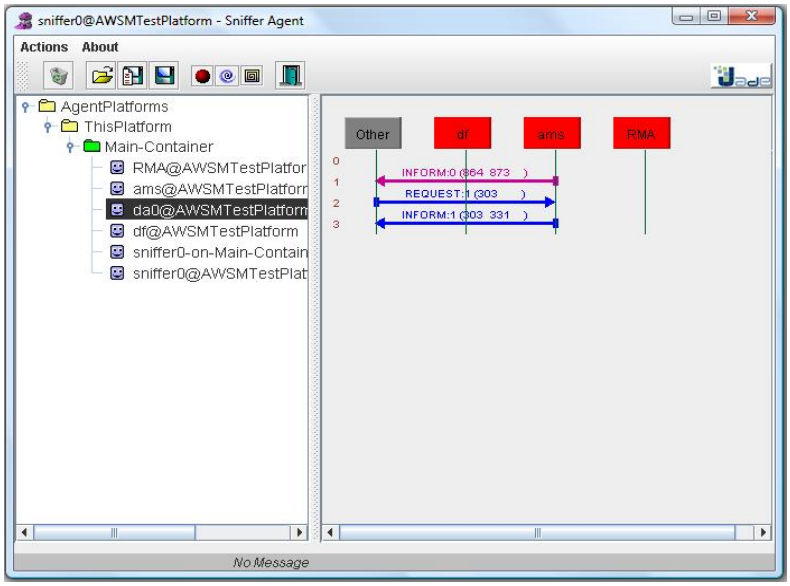
Şekil 2.16 : JADE RMA.

DA (Dummy Agent) grafiksel ara yüze sahip, etmenlere mesaj gönderme ve alam özelliğine sahip bir etmendir. DA belli bir etmene kullanıcının belirlediği mesaj göndermek ve etmenin tepkisini ölçmek için kullanılır. Daha çok geliştirme aşamalarında kullanılır. Şekil 2.17'de DA arayüzü gösterilmiştir.



Şekil 2.17 : JADE DA.

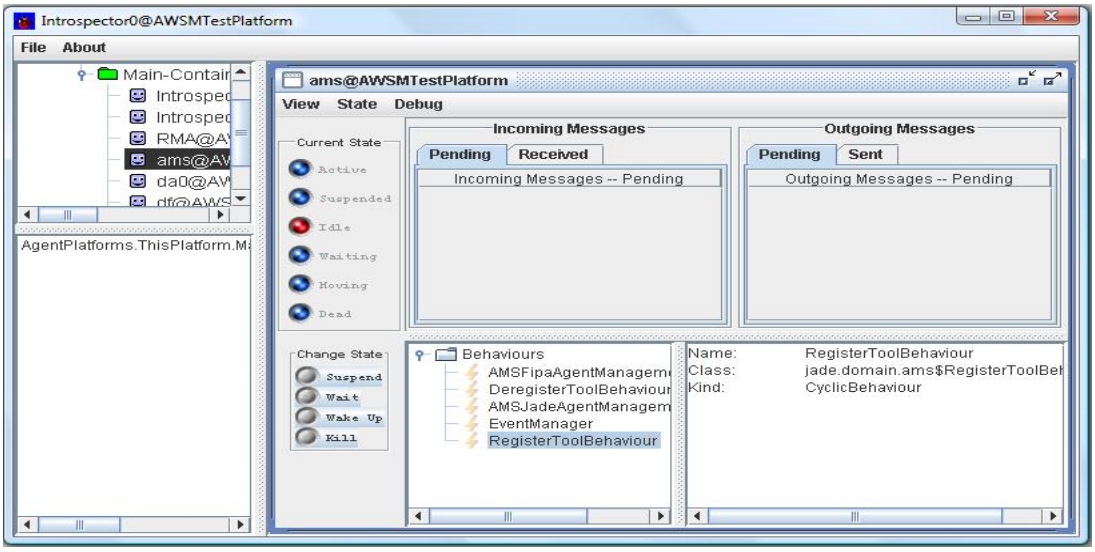
Sniffer platform içersinde akan ACL mesajlarının izini tutar. Bu araç yoğun olarak hata ayıklama noktasında kullanılmaktadır. Ayrıca etmenler arasındaki iletişimin detaylı kayıt edilmesi ve belgelenmesi için kullanılmaktadır. Etmenler arası iletişimde gelen ve giden mesajlar grafiksel olarak ve detaylı olarak gösterilebilmektedir. **Sniffer** etmeni etmenler arası mesajlardan haberdar olmak için AMS'ye abone olmaktadır. Şekil 2.18'de **sniffer** arayüzü gösterilmiştir.



Şekil 2.18 : JADE sniffer etmeni.

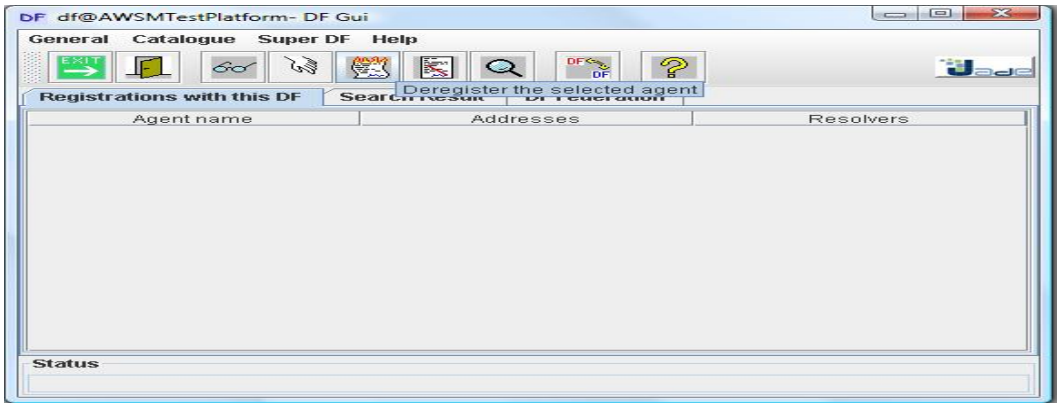
Introspector bir etmenin hayat döngüsünü izlemeye yardımcı olur. Etmenin davranışları ve gönderip aldığı mesajların izini tutar. **Sniffer** etmeni etmenler arası

iletişimi izlemek için uygunken **Introspector** etmeni etmenin davranışlarını izlemek ve hata ayıklaması yapmak için uygundur. Bu araç etmenin yaşam döngüsünü, davranış kuyruğunu, gelen ve giden mesaj kuyruklarını izlemek için kullanılır. Şekil 2.19'da **introspector** arayüzü gösterilmiştir.



Şekil 2.19 : JADE introspector etmeni.

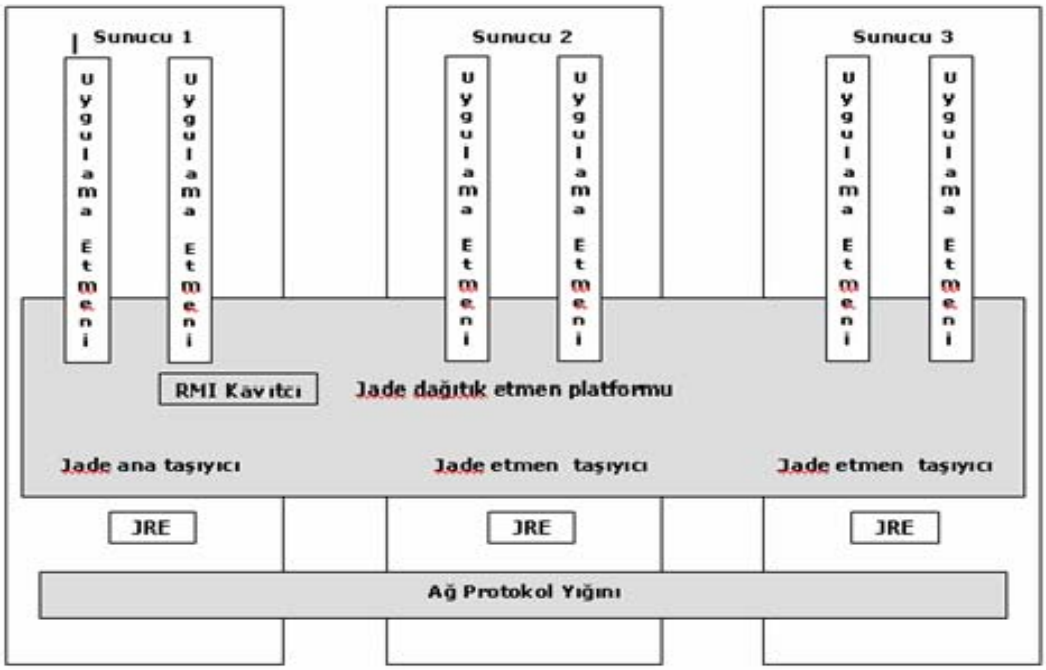
DF GUI (Directory Facilitator Graphical User Interface) platform tarafından verilmekte olan sarı sayfa servisinin grafiksel ara yüz aracılığı ile kontrol edilmesini sağlar. DF belli bir kritere uyan servis kayıt olduğunda ya da değiştirildiğinde uyarılmak isteyen etmenler için de uyarı servisi sağlamaktadır. DF servisleri üzerinde aramalar yapılabilmektedir. Sarı sayfa servisini farklı makinelere dağıtabilmek için birden fazla DF aynı anda çalışabilir. Gerekli ise DF'ler arasında federasyon kurulabilir ve birine yapılan kayıtlar diğer DF'lere de aktarılabilir. Şekil 2.20'de DF arayüzü gösterilmiştir.



Şekil 2.20 : JADE DF etmeni.

2.5.4 JADE diğer özellikleri

JADE tarafından programcılara sunulan bir takım özellikler bulunmaktadır. Bunlar arasında en önemlilerinden bir tanesi dağıtık etmen platformudur. Etmen platformu birden fazla sunucu üzerine Şekil 2.21'de görüldüğü gibi dağıtılabilir. Dağıtık haldeki bu parçalar birbirleri ile RMI aracılığıyla bağlantı kurarlar. Her bir sunucu üzerinde sadece bir Java Sanal Makinesi, işletilebildiği için sadece bir tane Java uygulaması çalıştırılabilmektedir. JADE içinde etmenlerin her biri bir Java iş parçacığı olarak gerçekleştirilmiştir. Etmenler *Etmen Taşıyıcılar* içinde hayatlarını devam ettirirler. Bu taşıyıcılar etmenlerin işlevlerini yerine getirebilmesi için çalışma zamanında destek sunarlar. JADE etmenlerin ve etmen taşıyıcılarının uzak sunucular üzerinden yönetilebilmesi için grafiksel kullanıcı ara yüzü servisi de vermektedir. Uygulama geliştiricilerin servisine sunulmak üzere JADE tabanlı *debug* araçları sunulmuştur.



Şekil 2.21 : Farklı taşıyıcılara dağıtılmış JADE platformu, [46] dan uyarlanmıştır.

JADE platformlar arasında etmenlerin kodları ve durumlarıyla birlikte aktarılabilmesi olanağını sunmaktadır. JADE etmen davranışları aracılığıyla etmen aktivitelerinin eş zamanlı olarak işletilebilmesine olanak tanınmaktadır. JADE tamamen FIPA uyumlu bir platformdur. Etmen platformu çalıştırıldığı anda AMS, DF ve ACC otomatik olarak aktif hale getirilir. İstenildiği takdirde çok alanlı uygulamalar için birden fazla DF çalışma zamanında başlatılabilir. Burada bahsi

geçen alan etmenlerin oluşturduğu mantıksal bir kümedir. Burada yer alan etmenlerin verdikleri servislerin tanıtımı ortak bir DF aracılığıyla duyurulabilir. Her bir DF bir GUI'den türetilmiş olup FIPA tarafından tarif edilen standart yeteneklere sahiptir. Bu yetenekler, kayıtlanma, kayıt silme, etmen tanımlarının değiştirilmesi ve araştırılması ile birden fazla DF'nin birleştirilmesi olarak sayılabilir.

JADE ACL mesajlarının aynı etmen platformu içersinde verimli bir şekilde taşınabilmesini sağlar. Aslında mesajlar karakter dizileri yerine Java nesneleri olarak taşınırlar. Platform içersinde dolaşan mesajlar otomatik olarak FIPA uyumlu hale dönüştürülürler. Bu işlem programcıdan tamamen bağımsızdır. Programcılar sadece ilgili Java nesneleri ile ilgilidir. FIPA etkileşim protokolleri için hazır kütüphaneler sunulmuştur. Etmenlerin AMS'ye otomatik olarak kayıtlanması ve kayıtlarının silinmesi sağlanmıştır. FIPA uyumlu isimlendirme servisi etmenler çalıştırılır çalıştırılmaz devreye girerek onlara platform içersinde tekil bir isim atanmasını sağlar. Uygulamaya özgü içerik dilleri ve ontolojiler için içsel fonksiyonlar mevcuttur. Dışsal uygulamaların özerk etmenleri çalıştırabilmesi için süreçler-içi ara yüz desteği sunulmuştur.

3. ETMEN TABANLI WEB SERVIS PAZARI (EWSP)

Bu bölümde servisler ile etmenlerin birlikte kullanıldığı ve ayrıca rol ile etmenlerin birlikte kullanıldığı literatürde yer alan çalışmalar anlatılacaktır.

3.1 Etmen Web Servisi Sistemleri

Etmenler ile web servislerinin bir arada kullanıldığı çalışmalar literatürde yer almaktadır. Bu çalışmalardan önemli olanları bu bölümde anlatılacaktır. Bu çalışmaların her birinde etmenler ile servisler farklı sorunları çözmek veya yeni çözümler ortaya koymak beraber kullanılmıştır. Çözümlerde etmenlerin ve servislerin daha önceki bölümlerde ortaya konulan güçlü yönleri bir arada kullanılmıştır.

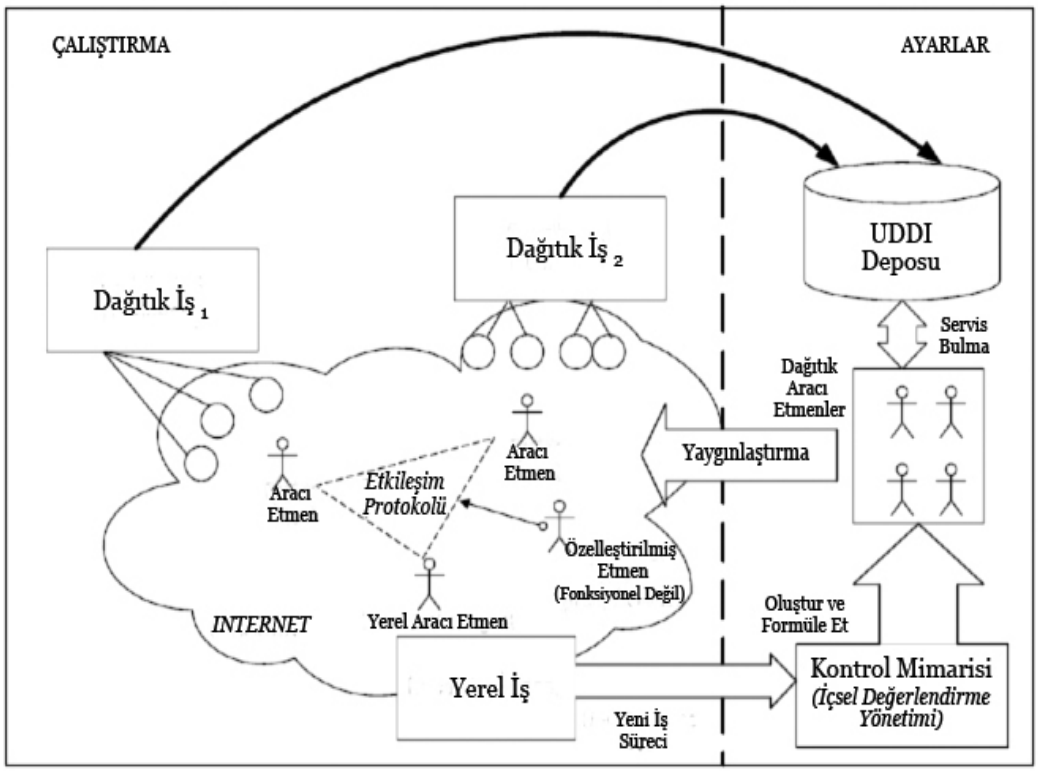
3.1.1 Servis tabanlı organizasyonlar arası iş akışı

Blake [17] çalışmasında, servislerin yönetilmesi ve ortaklaşa çağrılabilmeleri için etmen sisteminin kullanılmasını önermiştir. Etmenler farklı organizasyonların dahil olduğu iş akış sisteminde aracı olarak kullanılmıştır. Etmenler ortam bilgisine sahip otonom yazılım varlıkları olarak tanımlanmış, servislerin çalışmasına ve iş yönetimine aktif olarak vekillik etmişlerdir. Ayrıca etmenlere belli bir servisi ve iş sürecini yürütmeleri için görev verilmiştir.

Bir organizasyonun iş sürecinin diğer organizasyonun iş servisini kendi sürecinde kullandığı durumları tanımlamak için servis tabanlı organizasyonlar arası iş süreçleri (services-based cross-organizational workflow (SCW)) tanımı kullanılmıştır. SCW kavramı bileşen bileştirme kavramı ile ilgili ve onun devamı niteliğindedir. Bileşen bileştirmede bileşenler ve arayüzleri biçimsel dillerle (metin veya görsel) modellenmiştir. Bu tanımlamalar bileşen birleştirmeyi otomatikleştirmektedir.

Web servisleri UDDI kayıtçısında servis tanımlarını tutabilmektedir. Kayıtçı üzerinden servisler aranıp bulunabilmektedir. Servis tanımları (WSDL) bulma işlemi aracılık faaliyeti şeklindedir. SCW ortamında aracılık işlemleri etmenler tarafından gerçeklenmiştir.

Etmen tabanlı programlama iş akışı bileştirme ve servis yönetimi için kullanılmıştır. SCW ortamında aracı olarak yer alan etmenler aracı olarak servis çalıştırma, süreç yönetimi de yapmaktadırlar. Aracı etmenler dağıtık UDDI kayıtçılarında servis araması yapmak için servis tanımları verilerek yaratılmaktadırlar. İş süreci çalıştırması sırasında etmenler birden fazla uygulanabilir servis bulabilirler. Sürecin yürütülmesinde etmenler diğer iş organizasyonlarının servislerini kullanarak organizasyonlar üstü servisler oluşturabilmektedir. Şekil 3.1'de sistem mimarisi gösterilmiştir.



Şekil 3.1 : SCW ortamına entegre etmenler, [17] den uyarlanmıştır.

3.1.1.1 Değerlendirme

İş süreçlerinin dinamik olarak yönetilmesi esas alınmıştır. İş süreçlerinin dinamik olarak yürütülebilmesi için etmenler kullanılmıştır. Ayrıca iş süreçlerinin belli adımlarında web servisler kullanılmıştır. Bu çalışma dinamik iş süreçleri yaratmak için ortaya konulmuştur.

3.1.2 Sanal web servisleri- web servislerinin kişiselleştirilmesi

Rykowski, Cellary [19] statik olarak belli bir noktadan sunulan web-servislerini daha dinamik hale getirmek için Sanal Web Servisi (VWS) sistemini önermişlerdir. VWS, kullanıcıların tek bir web servisine erişir gibi eriştikleri pek çok gerçek ya da sanal web servislerinin toplamı genel ve özel etmenlerdir. Sanal Web Servisi farklı kaynaklardan ve farklı formdaki verileri birleştirerek tek servisle sunmayı sağlamaktadır. Sanal Web Servislerinin iki temel amacı vardır;

- Web servisi kişiselleştirme. Kullanıcı tarafından tanımlanan uygulamalar sunucu ve istemci tarafından çağrılabilir.
- Sanal servis sunucu tarafında servis tarafından sağlanan bilgilerin izlenmesi ve oluşan değişikliklerin uyarı ile ilgilenen kullanıcıya bildirilmesini sağlar.

Ek olarak servis kullanıcısı var olan servisleri kullanmak için kendi servis arayüzünü tanımlayabilir.

Servis kişiselleştirme ve istemci tarafından optimizasyonu ile ilgili temel problemler şunlardır;

- Bütün istemcilerin beklentilerini ve beklentilerinin ne anlama gelebileceğini öngörmenin mümkün olmaması. Herkes için geçerli bir arayüzün ortaya çıkarılamaması.
- Farklı özelliklerde erişim yöntemi ve kanalların sağlanamaması.
- Servis tarafından sağlanan bilginin sunucu tarafında izlenememesi.
- Yeni iş koşullarına yavaş ve maliyetli bir uyumun olması.
- Anlık veya geçici iş kontratlarına sınırlı destek sunulması.

Bu problemlerin çözülmesi için sanal servis önerilmiştir. Sanal servis kişiselleştirmesini, gruplamayı ve değişiklik izleme, uyarmayı destekler, dinamik iş isterlerini hızlıca adapte olur. VWS farklı sanal veya gerçek servisin toplamıdır ve tekil servis gibi kullanıcı tarafından erişilir. VWS belli bir kullanıcı tarafından yaratılır ve kullanıcının ihtiyacına göre kişiselleştirilmiştir. VWS gerçek servisleri sarmalar ve onların kişiselleştirilmesini sağlar. VWS farklı sayıdaki web servislerini bir araya getirir, onların bilgilerinin birleştirilmesini, bir arada çağrılmasını (sıralı veya aynı anda) sağlar.

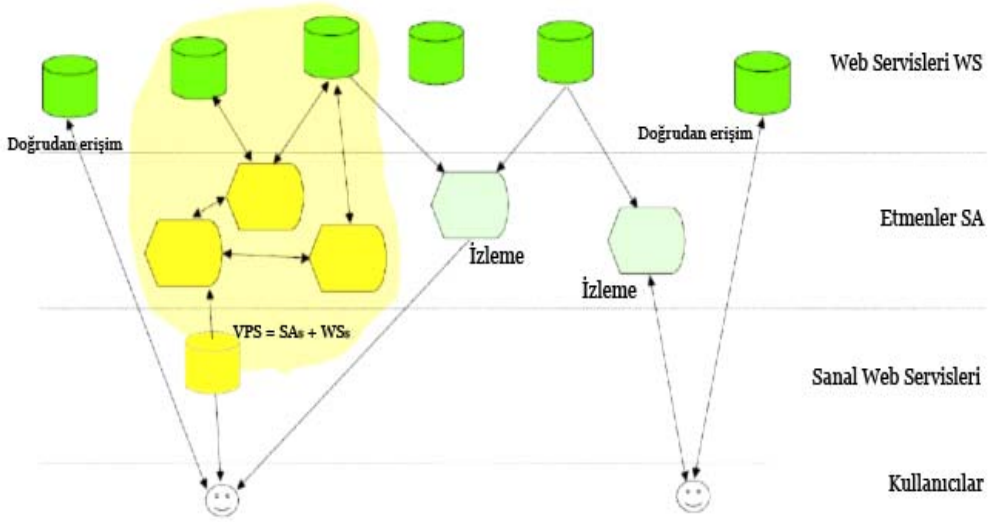
Başarılı bir şekilde VWS'yi gerçeklemek için otonom bir yapıda olan etmenler kullanılmıştır. VWS'de yazılım etmeni kullanmanın iki avantajı bulunmaktadır.

- Tekil web servislerini kişiselleştirme seviyesinin yükseltilmesi
- Pek çok bağımsız web servisinin tek bir web servisi haline getirilmesi için birleştirilmesi

Etmenler bilgi aracısı ve mevcut servisler için sarmalama işlevi görmektedir. Web servisi, servis istemcinin bilgi formatına (şekilsel ve anlamsal) uygun bilgi sağlamak durumundadır. Aracılık/sarmalama algoritmaları servis sağlayıcıları veya istemciler tarafından programlanabilir. Etmenler web servisi tarafından gözlemlenen önemli veri değişikliklerini izleyebilir ve gerekli uyarıları yapabilir.

3.1.2.1 Sanal web servisi için yazılım etmeni ve mimari

Sanal Web Servisleri ACE [28,29] çatısı kullanılarak gerçeklenmiştir. ACE dağıtık etmen sunucularına dayanmaktadır. Her bir sunucu etmen saklama ve işleme yeteneğine sahiptir. Etmenler XML ve Java dili kullanılarak programlanmıştır. Etmenler bu sistem içinde harekete edebilmektedir.



Şekil 3.2 : Web servisleri, etmenler, sanal web servisleri ve kullanıcıları, [19] dan uyarlanmıştır.

Çatının kullanılmasındaki temel amaç etmenleri kullanarak mevcut web servislerin servis kişiselleştirmesini arttırmaktır. Etmen, dağıtılmış web servisleri ve etmen sahipleri arasında kullanıcı tarafından programlanmış aracı gibi davranmaktadır. Etmenler bilginin belli bir servisten alındığı VWS yaratmak için kullanılmaktadır.

VWS'ler başka bir etmen veya servis tarafından da kullanılabilir. ACE etmenleri web servis tarafından sağlanan bilginin değişiminin izlenmesi işlevini de yerine getirmektedir. Etmenler sahipleri tarafından programlanır. Dolayısıyla her bir etmen sahibinin öncelik ve isterlerine göre farklı bir yapıdadır. Sistemde Şekil 3.2'de gösterildiği gibi 4 temel eleman vardır; Erişilecek web servisi, genel Servis Etmeni, Özel etmen ve özel Sanal Web Servisi.

Web Servisleri dışsal yazılım olarak değerlendirilmiştir. Sahipleri adına etmenler tarafından kullanılabilir ya da erişilebilir. WS'ler kullanılabilirler ancak kullanıcı tarafından değiştirilemezler. WS'lerin kullanımı dengeli ve tanımlıdır. WS'ler arasında bağlantı yoktur. WS'lerin daha önceden tanımlanmış (kullanıcı, şifre bilgisi) fonksiyonları vardır.

Genel Servis Etmenleri (SA) güvenli kullanıcılar tarafından yaratılırlar, etkili olarak (paralel veya sıralı olarak) çok fazla kullanıcı tarafından kullanılırlar. SA'lar güvenilir sistem elemanı olarak değerlendirilir. SA'lar aşağıdaki temel işler için kullanılır;

- İletişim sarmalıdırlar. Belli bir protokol ile iletişimi sağlarlar ve son kullanıcıdan veri transferi detaylarını saklarlar. Uzak yazılımlarla (web servisleri vs) standart erişimleri sağlarlar.
- Çok fazla kullanıcının tek bir web servisine erişimini senkronize ederler ve tamponlama yaparlar.
- Web servisleri için sarmamla ve senkronize işlevini yerine getirirler. Çok fazla etmen tarafından belli bir WS için optimum erişimi sağlarlar
- Aracılık yaparlar. Çok fazla WS'den gelen veriyi işlerler ve en sonunda birleştirirler.

Bu işlerin bir veya daha fazlası bir etmende olabilir, ayrıca bunun dışındaki işlemler de olabilir. Bir veya daha fazla SA birleşerek tek bir VWS hazırlayabilirler.

Özel Etmenler (PA) sahibi tarafından yaratılır ve kontrol edilirler. Sahibi tarafından belirtilmediyse etmen başka bir etmen veya servis tarafından erişilemez. Önerilen XML tabanlı Agent Shell Language (ASL) kullanılarak etmen kodu yazılır.

Sanal Web Servisleri tekil olarak PA'lar gibi kullanıcılar tarafından yaratılan ve kullanılan Sanal Web Servisleri (VWS). VWS'ler bağlı etmenler ve servislerin

kümesidir ve tekil servis gibi davranırlar. VWS'ler özel etmenlerden oluşurlar ve pek çok SA ve WS ile bağlıdırlar. Tekil PA VWS'nin giriş noktasıdır. VWS içindeki bağlantılar zaman içinde değişebilirler. VWS iki programlama katmanından oluşur: tekil kullanıcı kontrolü ile çalışan ve özel etmenlerden oluşan özel katman ve sistem etmenlerinden oluşan ve yöneticiler tarafından yönetilen sistem/genel katmanı. Özel etmenler:

- VWS oluşturmak,
- Servis kişiselleştirme,
- İzleme gibi üst seviye işler için kullanılan görevler

için kullanılır. Özel etmen dışarıdan gelen çağrılar sonucu web servis işlevini yerine getirir ve sonuçları çağırana döner ayrıca kendi kendine de izleme işlemleri için çalışabilir.

Genel Katman;

- Bilgi depolama
- Kitlesel hesaplama
- Farklı web servisleri için tekil bir servis arayüzü ile erişme
- Yüksek seviye sarmalama
- Veri tamponlama
- Kitlesel senkronizasyon

görevlerini yetine getirir. Genel katmanın fonksiyonları en baştan belirlenmiştir. Web servisleri için geçitte olan Sistem etmenleri sistem tasarım aşamasında belirlenir ve VWS sahibi tarafından bilinir.

3.1.2.2 Değerlendirme

Kullanıcılar için kişiselleştirilmiş sanal web servislerinin yaratılmasına olanak sağlanmıştır. Sanal web servisleri etmenlerden oluşmaktadır. ACE etmen çatısı kullanılarak gerçekleme yapılmıştır. Kullanıcılar ihtiyaçları doğrultusunda etmen çatısının kendisine verdiği betik dilini kullanarak etmenleri ve kişiselleştirilmiş servisleri yaratabilmektedir. Sanal web servisi kişiselleştirmesinde kullanıcı ortaya koyduğu sanal web servisinin arayüzünü kendisi belirleyebilmektedir. Sanal web

servisleri dinamik olarak ya da bir arama sonrasında ortay çıkmamakta statik bir şekilde ve tanımla yaratılmaktadır. Bu durum sistemi statik bir yapı haline getirmektedir.

3.1.3 Hisse senedi ticareti simülasyon sistemi

Wang, Wang, Deng, Zhao, Yung, Gao [20] web servisi etmen tabanlı Hisse Ticareti Simülasyon Sistemi (Securities Trading Simulation System - STSS) geliştirmiştir. Etmen teknolojisi karmaşık, dinamik ve dağıtık hisse senedi ticaretini gerçekleştirmek için uygulanmıştır. Web servis teknikleri farklı ortamlarda çalışabilirlik ve ağ temelli iş ortamlarındaki ölçeklenebilirliği arttırmak için önerilmiştir. Etmen teknolojisi ile web servislerini entegre etmek ile iki teknolojinin avantajlarının kullanılması sağlanmıştır. Bu yaklaşım hisse senedi ticareti sürecinin benzetim kurulması için daha zeki, daha esnek daha kapsamlı bir platform sağlamıştır.

3.1.3.1 Akıllı etmen sistemleri ile web servis entegrasyonu

Etmen sistemleri farklı noktalarda web servislerini daha gelişkin hale getirebilir [1].

- Web servisleri ontoloji kullanmamaktadır
- Web servisi yalnızca kendisinden haberdardır oysa etmenler kendisinden, aynı ortamda bulunduğu diğer etmenlerden ve onların yetenekleri ve davranışlarından haberdardırlar.
- Web servisleri otonom değildir oysa otonomi etmenlerin karakteristiğidir.
- Etmenler iletişim halindedir oysa web servisleri çağrılıncaya kadar iletişim halinde değildir.
- Etmenler gelişkin servisler sunabilmek için işbirliği halindedir, takımlar ve koalisyonlar oluşturabilirler.

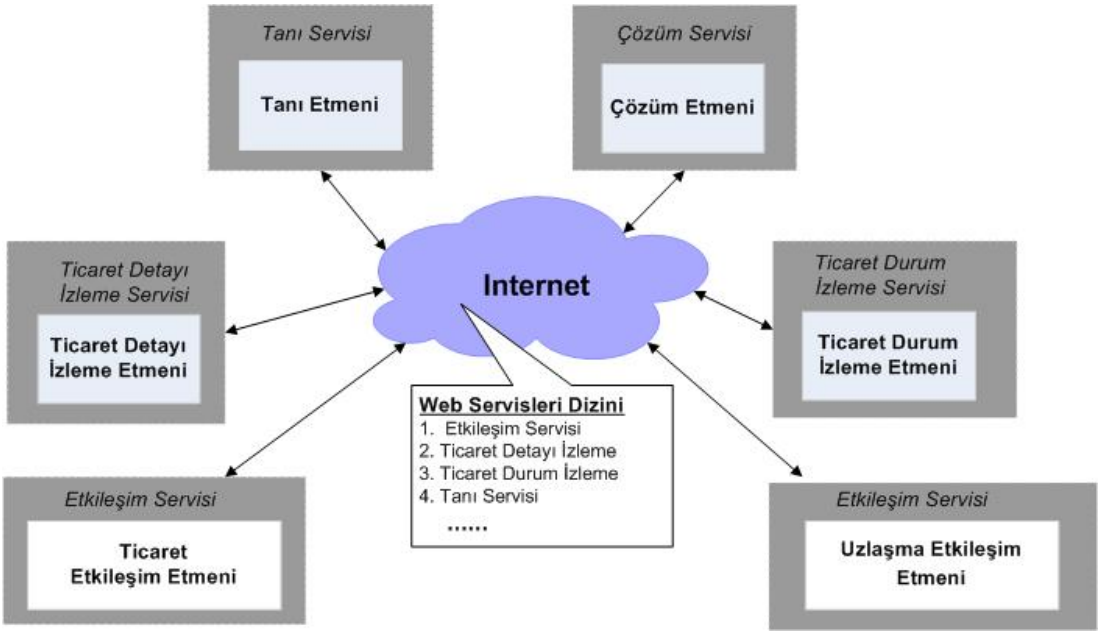
3.1.3.2 STSS'nin mimari tasarımı

Çoklu etmen teknolojilerin STSS'ye uygulamadan önce hisse senedi ticareti sürecini, her etmenin amaç odaklı ve reaktif davranışlarını göstererek ve amacına ulaşmak için diğer etmenler ile işbirliği yaparak yürüteceği görevlerden oluşan otonom alt süreçlere parçalamamız gerekmektedir. STSS, otomatik ticaret sistemi tabanlı karar

destek sistemi olarak katılımcılarına sipariş alacakları ve verebilecekleri bir platform sunmaktadır. Bu platform Chicago Stock Exchange (CHX) tarihsel verilerine dayanmaktadır. Bütün süreç ikiye bölünebilir. Bir parçası ana olarak ortam tarafından yaratılan siparişlere ve işlemlere odaklanmıştır, diğeri ise simülasyon katılımcıları ile olan etkileşime odaklanmıştır. Sistemde görev odaklı şu etmenler tanımlanmıştır;

- Veri Toplayıcı Etmen: Gelen ortam verileri ile uğraşır
- Müşteri Etkileşim Etmeni: Katılımcılar için erişim noktasıdır. Diğer etmenlerle etkileşime geçebilmek için diğer etmenler hakkında bilgi sahibi olmak durumundadır.
- Performans İzleme Etmeni: Bütün katılımcıların performansını izler
- Sipariş İşleme Etmeni: Siparişleri ve süreçlerini yönetir.
- Simülasyon Yönetim Etmeni: İşlem onaylarını gerçekleştirir. Bilgi sağlayıcılık hizmeti verir.

Dinamik ve karmaşık hisse senedi ticareti süreçleri otonom etmenlere delege edilmiştir. Etmenler birbirleri ile açık ortamdan iletişim kuran ve etkileşime geçen web servisleri olarak sarmalanmıştır Şekil 3.3'de web servis etmen tabanlı STSS mimarisi gösterilmektedir.



Şekil 3.3 : STSS mimarisi, [20] den uyarlanmıştır.

Bütün web servisi etmenleri otonom ve işbirliği halinde internet üzerinden çalışmaktadırlar. Her bir etmen kendi ana görevine odaklanmıştır. Belli bir işin içinde birden fazla etmen varsa, etmenler birbirlerinin bilgi ve yeteneklerinden amaçlarını yerine getirmek için faydalanırlar.

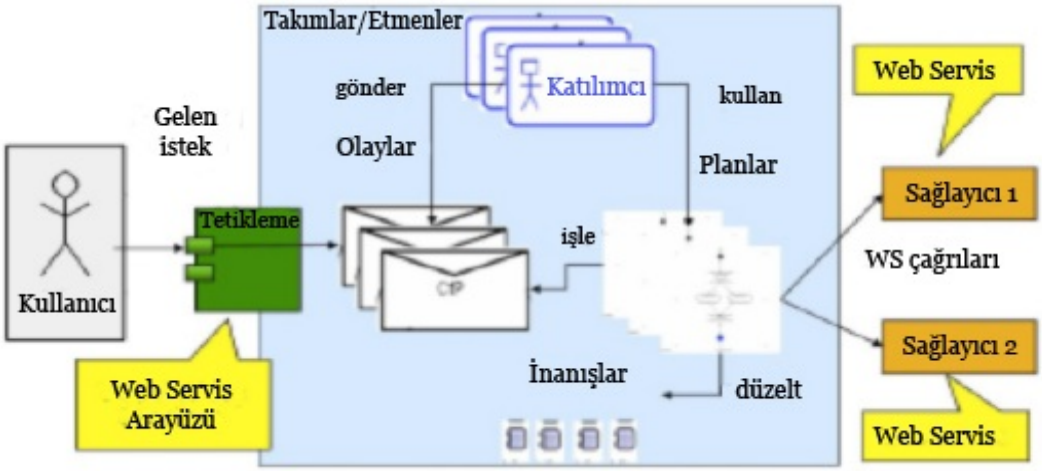
3.1.3.3 Sistem gerçeklenmesi

Uygulama Java ile geliştirilmiştir. Etmen iletişim dili olarak Knowledge Query and Manipulation Language (KQML) kullanılmıştır. Web servis teknolojisi standartları (SOAP, UDDI, WSDL) kullanılmıştır. KQML SOAP yapısı tarafından sarmalanmıştır. XML olarak ifade edilen SAOP mesajı http üzerinden taşınmıştır. Etmenler arası haberleşme Java XML api'si ile sağlanmıştır. Web servis etmenleri oluşturulduktan sonra WSDL tanımları oluşturulur ve yayınlanır.

3.1.3.4 Değerlendirme

Hisse senedi ticareti simülasyon sisteminde etmenlerin akıllı ve otonom hareket etme özelliğinden faydalanılarak problem alt problemlere bölünerek simülasyon işlemi gerçekleştirilmiştir. Web servisleri ise etmenler arası haberleşmede ayrı dış sistemler ile haberleşmede kullanılmıştır. Web servislerinin farklı ortamlarda çağrılma ve etkileşim özelliği kullanılmıştır.

3.1.4 JACK etmen çatısının web servisi için genişletilmesi



Şekil 3.4 : JACK etmen çatısının genişletilmiş hali, [31] den uyarlanmıştır.

JACK [30] çoklu etmen sistemleri oluşturmak, çalıştırmak ve entegre etmek için kullanılan bir ortamdır. JACK bileşen tabanlı bir yaklaşım kullanmıştır. JACK'ın

bileşen tabanlı yaklaşımı onun eklentiler aracılığı ile genişletilmesine olanak sağlamaktadır. Web servisi ile tümleştirmek için Şekil 3.4'da görüldüğü gibi JACK etmen çatısı genişletilmiştir [31]. Etmen amacına ne şekilde ulaşacağını tanımlayan plan ile beraber programlanmıştır. Her bir plan amaca ne şekilde ulaşılacağını göstermektedir. Etmen o anki verilerine göre amaca ulaşmak için planında gerekli düzenlemeleri yapar. Etmenler genel amaca ulaşmak için gruplar oluşturabilmektedirler.

Web servisi sağlayıcısı etmen olarak haritalandırılır. Web servisi etmenlerin planlarına olaylarla ilişkili olarak haritalandırılır. Arayüzler (istemci, sunucu kodları) ise ayrı bir modül olarak tanımlanır. Bu durumda web servisi adaptörleri ile planlar birbirinden ayrılmış olur. Bu metodoloji aşağıdaki adımlara sahiptir;

- İçsel işlevleri koordine etmek ve web servisi çağrıları oluşturmak için plan ve etmen grupları oluşturulur.
- Servisin WSDL tanımı temel alınarak adaptör kodları (istemci ve sunucu tarafları) üretilir
- Kullanılan ve sunulan web servisi tarafından sağlanan veri yapısına uyumlu olarak etmen verileri (**belief**) ayarlanır.
- Web servisi (sağlanan) çağrıları ile bağlantılı olarak olayları belirlenir
- Planlar bu olayları ele alacak şekilde ayarlanır.
- Web servisi çağrılarını tetikleyecek plan adımlarını ekle (etmen web servisi çağırıyorsa) ve adaptör kodlarını plan içinden doğrudan çağır.

3.1.4.1 Değerlendirme

JACK etmen çatısına web servisi çağırma ve web servisi sağlama özelliği sağlanmıştır. JACK çatısının bileşen yapısı kullanılarak adaptörler üzerinde bu özellikler sağlanmıştır. Bu sistem JACK etmen çatısına web servisi etkileşim özelliği getirmek için ortaya konmuştur.

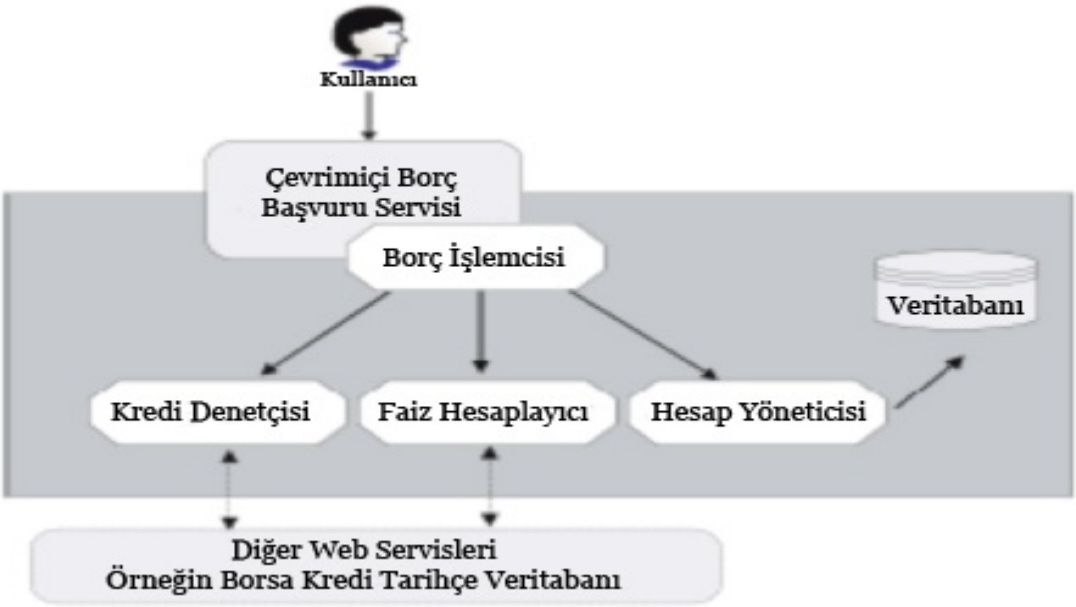
3.1.5 Web servisleri için etmen yönelimli yöntem

Etmenler ile web servislerini tümleştirmek için Hongen Lu ve Manish Chhabra etmenleri web servislerinin içine sarmalayarak kullanmayı önermiştir [32]. Dışarıdan web servisine gelen çağrılar adaptör arayüzü aracılığı ile etmenlere aktarılmakta,

etmenler sonucu web servisine geri bildirmektedir. Sistemi geliştirmek amacıyla Wrapping Agents into Service (WAiWS) isimli bir yöntem önerilmiştir. Bu yöntemin adımları aşağıdaki gibi tanımlanabilir.

1. Adaptör arayüzünü tanımla. Bu adaptör, etmen ile WS (web servisi) arasındaki iletişimde kullanılmaktadır. Bu adaptör arayüzünün metotların gövdesi etmende gerçeklenmektedir. Metotlar olay üretmektedirler ve bu olaylara bağlı etmenin planı çağrılmaktadır.
2. Etmenin yöneteceği olayları tanımla. Bu olaylar adaptör metotları tarafından etmenin işlemesi için üretilir. Etmen yönelimli WH aktiftir, olayları yönetebilir ve kendisi de diğerlerinin işlemesi için olay yaratır. Bu özellik bütün WS'yi daha dinamik ve ortamdaki değişikliklere karşı daha uyumlu hale getirir.
3. Olay yaratıldığı zaman başlayan iş süreci. İş süreci etmenin olaya bağlı olan planı ile ilişkilidir. Herhangi bir iş süreci kendisini yürüten etmenin planına bağlıdır. Bu durum servisi daha esnek ve gelecekteki değişiklere daha iyi adapte olabilir yapmaktadır.
4. Etmen ve etmenin içinde adaptör arayüzü tarafından 1. adımda tanımlanan metotların gövdesini gerçekle. Etmenin gövde kısmında, iş sürecini çalıştırma planı verilir. JACK'de etmen planı ile iş süreci yakın bir şekilde ilgilidir. Hatta planlar yeniden kullanılabilirdir. Bu geliştirme zamanını önemli miktarda kısaltmaktadır.
5. En sonunda WH yaratılır ve etmen çağrılır. Etmen çağrısı adaptör aracılığı ile gerçekleştirilir.

Şekil 3.5'de örnek bir olay üzerinden sistemin yapısı gösterilmektedir. Kredi başvuru servisi (Loan Application Service) kullanıcıların kredi başvurularını bankaya gönderebileceği bir servistir. Bu servis submitLoanApplication() metodunu müşterilerden başvuru almak için tanımlamıştır. Bu metot yeni başvuruyu işler ve yeni kredi başvurusu olayını yaratır. Dolayısıyla bu olay ile Kredi İşleme Etmeni (Loan Processor) çağrılmış olur. Etmen diğer etmenlerle de irtibata geçerek servis için gerekli bilgiyi toplar.



Şekil 3.5 : Etmen web servis çağrısı örneği, [32] den uyarlanmıştır.

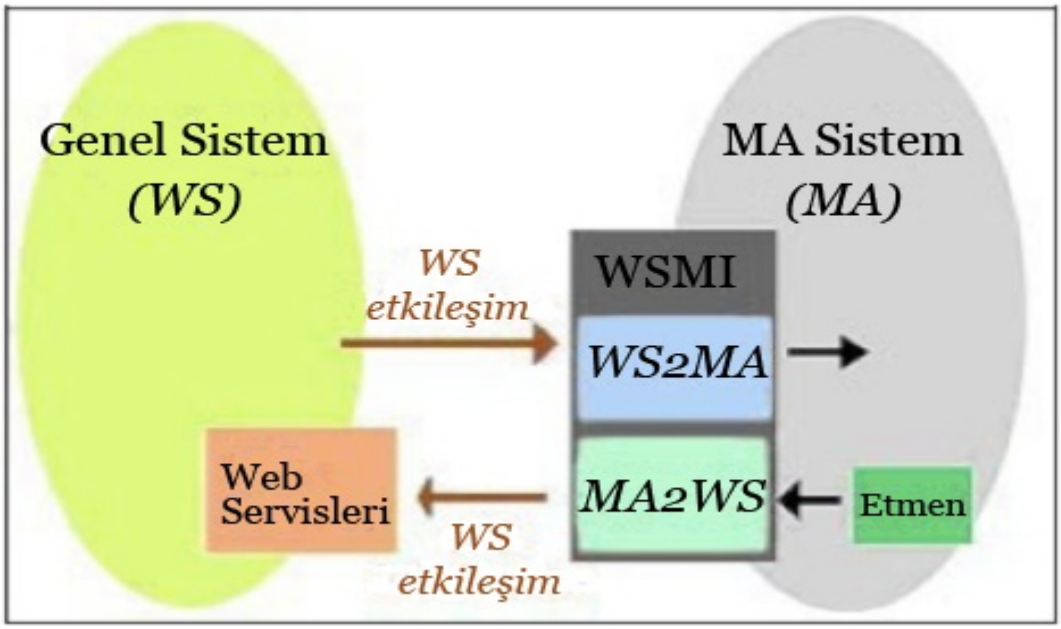
Bu yöntem ile etmenler ile WH iletişim haline geçirilmiş ve servislerin yerine getirmesi gereken işler etmenlere aktarılmıştır. Böylece etmenlerin temel özellikleri WH içersinde de kullanılabilmesi sağlanmaktadır.

3.1.5.1 Değerlendirme

Bu çalışmada JACK etmen çatısı üzerinde etmenleri ve fonksiyonlarını web servisleri içersine sarmalayarak kullanma yöntemleri geliştirilmiştir. Web servisi çağrıları sonrasında, gelen çağrı etmen metoduna aktarılmakta ve işlemler etmen içersinde sonlandırılmaktadır. Etmenler ve web servisleri bir arada kullanılarak iki sistemin avantajlarından yararlanma ve daha iyi bir sistem ortaya çıkarmak amaçlanmıştır.

3.1.6 Web servisleri ile hareketli etmen sistemlerini tümleştirmek

Bellavista, Corradi, Monit çalışmalarında web servisleri ile hareketli etmenleri tümleştirmek amacıyla WSMI (Hareketli etmen tümleştirmesi için web servisi) ismiyle bir altyapı geliştirilmiştir [33]. WSMI dış sistemlerin hareketli etmen sistemlerinin avantajlı yönlerini kullanmasını sağladığı gibi, hareketli etmen sistemlerine de web servislerini kullanma olanağı sağlamaktadır. WSMI altyapısı hareketli etmenlerin asenkron dünyası ile senkron SOM (Servis Odaklı Mimari) dünyası arasında köprü görevi görmektedir.



Şekil 3.6 : Hareketli etmen web servisi tümleştirmesi, [33] den uyarlanmıştır.

Şekil 3.6'da görüldüğü gibi WSMI iki yönlü tümleştirme gerçekleştirmektedir.

- WS2MA: WSMI dış sistemin hareketli etmen sistemi fonksiyonlarını çağırabilmesi için web servisi (WS) arayüzü tanımlamaktadır.
- MA2WS: WSMI hareketli etmen sisteminin hareketli etmen olmayan WH bileşenlerini çağırabilmesi için arayüzler tanımlamaktadır.

3.1.6.1 MA2WS

Hareketli etmenin web servisini çağırmasını sağlayan bileşendir. Klasik WS çağrısından farklı olarak sonlanma noktası (çağıran) hareketlidir. HE (Hareketli Etmen) web servisi çağrısı yaptıktan sonra hareket edebilir. Bu problemi çözmek için WSMI vekil-tabanlı mimari önermiştir. Vekil WS çağrısında aşağıdaki işlemleri gerçekleştirmektedir.

- HE'den WS isteğini alır,
- WS sağlayıcısına isteği iletir,
- WS sağlayıcısından yanıtı alır,
- Yanıtı HE'nin o anki yerine farklı iletim yöntemleri ile yönlendirir.

Yanıt iletme için farklı aşağıdaki iletim yöntemleri kullanılır:

- Kara tahta tabanlı yöntem: yanıt etmenin yanıtı çekebileceği kara tahtaya yazılır.
- Mesajlaşma yöntemi: yanıt etmene mesaj olarak iletilir.

3.1.6.2 WS2MA

WS2MA dış kullanıcıların HE sistemi fonksiyonlarını kullanması için WH arayüzü sunar. İki tip WS2MA fonksiyonları tanımlanmıştır; çalışma zamanı MA sistem özelliklerini tanımlayan tanımlayıcı servisler, MA sistem aksiyonlarına erişen ve çağıran aktif servisler.

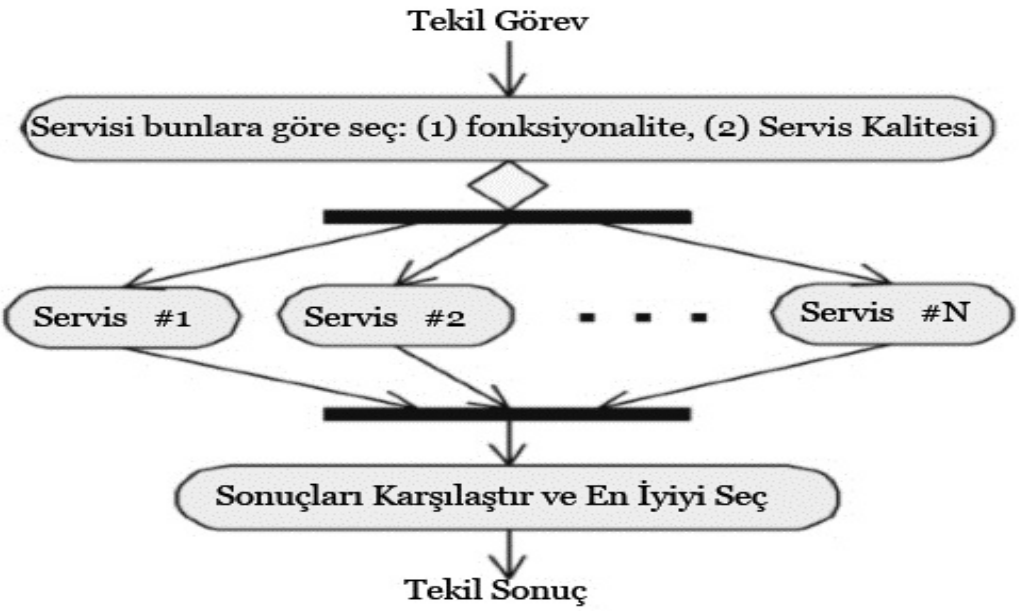
3.1.6.3 Değerlendirme

Hareketli etmenler ile web servislerini bütünleştirmek için yöntemler geliştirilmiştir. Senkron olan web servisleri ile asenkron olan hareketli etmenleri haberleştirmek için kara tahta yöntemi ve mesajlaşma yöntemi önerilmiştir. Hareketli etmen sisteminin sağladığı web servisleri gelen çağrıları hareketli etmenlere iletmek için bu yöntemleri kullanmıştır. Hareketli etmenlerin web servisi çağırması içinde yöntem ortaya konulmuştur.

3.1.7 Web servisi sağlamlığını artırmak için etmen kullanımı

Huhns, N. M. Web servislerinin sağlamlığını arttırmak için etmenler ile birlikte kullanılması önermiştir [34]. Aynı servis fonksiyonunu yerine getiren birden fazla servis olabilirse, birisi çöktüğü zaman bir diğeri gerekli servisi sağlayabilir. Aynı fonksiyonu sağlayan birden fazla servisin bulunduğu ortamlar yaratılabilirse sağlamlık önemli ölçüde artmış olur. Bunun için aynı fonksiyonu yerine getirecek etmen toplulukları önerilmektedir.

Şekil 3.7'da görüldüğü üzere güçlü yanları, zayıf yanları ve belli hataları olan ve aynı fonksiyonu yerine getiren servisler içeren bir grup oluşturulmuştur. Yanıtlanması gereken soru şudur: bu etmenler arasında hangi şekilde bir seçim yapılmalıdır ki en etkin sonuç alınabilsin.



Şekil 3.7 : N adet etmenin web servisi vermesi, [34] ten uyarlanmıştır.

Bu seçimle ilgili üç temel yaklaşım bulunmaktadır;

- Ön işlem: daha önce ilan edilmiş servis karakteristiklerine göre fonksiyonu yerine getirecek en iyi servisi seçmek. Ancak, bu durumda yeni servis eklendiğinde veya servisler değiştiğinde ön işlem yapmak zorlaşır.
- Son işlem: servisleri birkaç kez çalıştırarak en iyi servisi seçmek. Bu durumda servisler pek çok kez çalıştırılacağı için kaynak israfı olacaktır.
- Grup olarak hangi etmenin görevi yerine getireceğine karar vermek. Bu durumda etmenlerin kendi servisleri ve diğer etmenler hakkında bilgi sahibi olmaları gerekmektedir.

Yukarda anlatılan durumları denemek için iki yönlü listeyi tersine çevirmek ve daha sonra sıralamak için 25 değişik algoritma kullanılarak etmen servisleri yaratılmış ve denemeler yapılmıştır. En iyi sonuç son işlem ile elde edilmiştir.

3.1.7.1 Değerlendirme

Web servislerinin dayanaklılığını arttırmak için etmenlerden yararlanılmıştır. Web servislerine gelen çağrılar etmenlere aktırılarak işlemler etmenlere yaptırılmış ve sonuçlandırılmıştır. Web servislerinin dayanıklılığını arttırmak için aynı web servisinin arkasına aynı işi yapan n adet etmen yerleştirilmiştir. Bu etmenlerden hangisinin web servisi çağrısı sonrası çağrılacağı bir algoritma ile belirlenmiştir.

3.2 Etmen Web Servisi Sistemlerinin Değerlendirilmesi

Bir önceki bölümde özellikleri anlatılan etmen web servisi sistemleri farklı farklı noktalara yoğunlaşmaktadır. İncelenen sistemler, etmenlerin otonom hareket etme ve akıllı olma özellikleri ile web servislerinin farklı ortamlarda çalışma ve çağrılma yeteneklerini birleştirilmeye çalışmıştırlar. Servis tabanlı iş akış sistemi, Hisse senedi ticareti simülasyon sistemi ve Web servisi sağlamlığını artırmak için etmen kullanımı çalışmalarında, ortaya koydukları problemi çözmek için web servisleri ve etmenleri bir arada kullanmıştır. JACK etmen çatısı ile web servis entegrasyon çalışmalarında daha çok JACK sistemine özgü entegrasyon teknik problemleri çözülmüştür.

Tez çalışmasında etmenler ile web servislerinin sadece belli bir konuda bir arada kullanılması değil bir bütün olarak etmenlerle servislerin bütünleştirilmesi önerilmiştir. Etmen fonksiyonları web servislerine dönüştürülmüştür. Böylece etmenlerin yerine getirdiği bütün fonksiyonlar web servisi şeklinde dış dünyaya sunulabilecektir. Web servisleri sadece etmenlerin etkileşimini arttırmak amacı ile kullanılmamış etmenlerin fonksiyonları da servis odaklı olarak geliştirilmiştir. Roller üzerinden etmenlerin servisleri yönetilmiştir. Servislerin etmen sistemi içinde yönetilmesi (roller aracılığı ile) için yapılar geliştirilmiştir. Etmenlerin servislerini satmak için web servisi pazarı geliştirilmiştir. Ayrıca etmen servislerin tanımlanması dinamik olarak gerçekleştirilmiştir. Önerilen sistemin dış dünyaya web servisi verme (web servis sağlayıcılığı) özelliği bulunmaktadır.

3.3 Rol Tabanlı Etmen Sistemleri

Etmen sistemleri otonom olarak ve diğer sistemlerle etkileşim halinde kendi görevlerini yerine getiren sistemlerdir. Çoklu etmen sistemlerinde, etmen koordine eden veya edilen varlık durumlarına gelebilirler [35]. Bu etkileşim sırasında ve açık çoklu etmen sistemlerinde etmenlerden farklı işlevleri yerine getirmesi beklenebilir. Ek olarak dinamik ortamlarda da etmenlerin davranış değişikliği göstermesi beklenebilir.

Etmen tasarımında rol kavramının kullanımı etmene yeni davranışlar, yeni koordinasyon ve yeni işbirliği özellikleri eklenmesini ve ayrıca çoklu etmen sisteminin yeniden kullanır olabilme özelliğini katan bir yaklaşımdır. Rollerin ve etmen sistemlerinin birlikte kullanıldığı rol tabanlı etmen sistemi yukarda belirtilen

ihtiyaçlardan ortaya çıkmıştır. Literatürde yer alan rol tabanlı etmen sistemleri ile ilgili çalışmalar bir sonraki bölümde tanıtılacaktır.

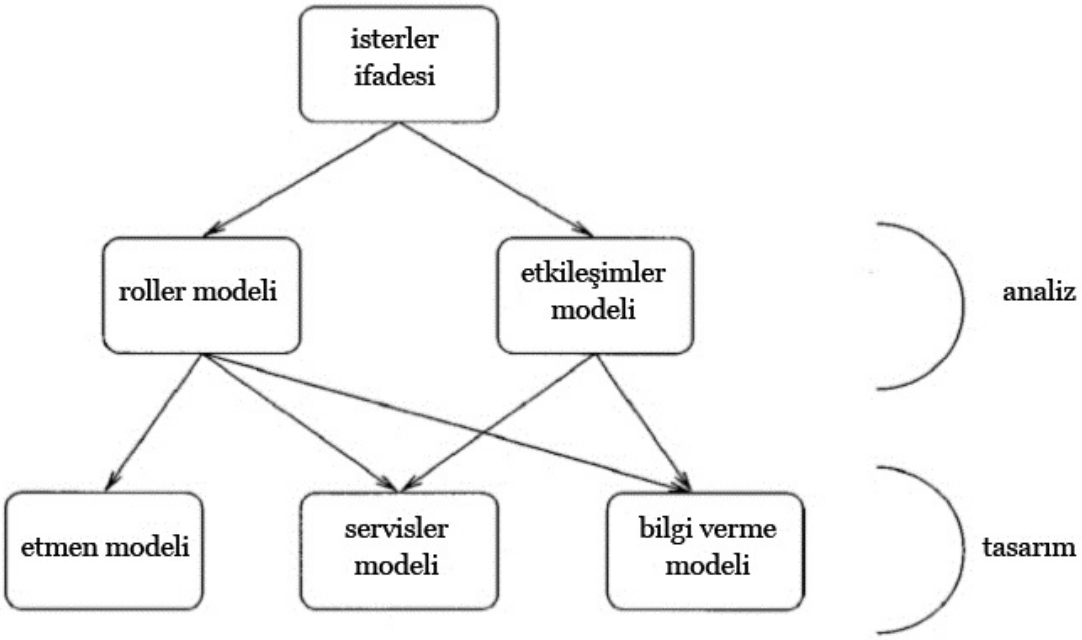
3.3.1 Gaia

Gaia[36], etmene yönelik analiz ve tasarım için bir yöntemdir. Gaia'nın amacı çoklu-etmen sistemlerini içerisinde farklı rollerin etkileşim halinde olduğu organizasyon şeklinde modellemektir.

Şekil 3.8'de görüldüğü gibi Gaia yönteminde kavramsaldan somuta doğru giden bir yol izlenmektedir.

Analiz: Analiz aşamasında organizasyon yapısı anlamaya çalışılır. Organizasyon bir biriyle ilişki halinde olan rollerin toplamı olarak değerlendirilir. Roller dört nitelikleri ile tanımlanmıştır:

- Sorumluluklar
- İzinler
- Yerine getirilmesi gereken aktiviteler
- Diğer rollerle ne şekilde etkileşim halinde olacağını belirleyen protokoller



Şekil 3.8 : Gaia metodolojisinin modelleri arasındaki ilişkiler, [36] dan uyarlanmıştır.

Rol modelde, sistemde bulunan roller ve bunların sahip olması gereken ve yukarda belirtilen dört özellik detaylı olarak ortaya çıkarılır. Etkileşim modelinde, sistemde roller arası bulunan etkileşimler bulunur. Bu etkileşimler protokoller halinde ifade edilir.

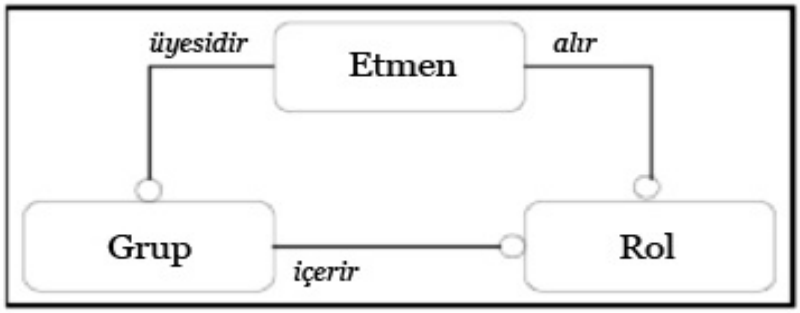
Tasarım: Tasarım aşamasında üç model yaratılır. Etmen modelinde, sistemin çalışması sırasında kullanılacak etmenler tipleri belirlenir. Roller birleştirilerek yeni etmen tipleri oluşturulur. Servis modelinde her bir rolün servisleri ve bu servisin özellikleri belirlenir. Gaia'nın servisten kastı etmenin fonksiyonudur diyebiliriz. Analiz aşamasındaki aktiviteler servislere dönüşür. Tanıma modelinde etmen tipleri arasındaki iletişimler tanımlanır.

Gaia'da, roller sadece analiz aşamasında kullanılmakta tasarım aşamasında roller etmen tiplerine dönüşmektedir. Gaia daha çok analiz ve tasarım aşamasına yoğunlaşmış, rollerin özellikleri ve roller etmen ilişkisi üzerinde fazla durmamıştır.

3.3.2 AALAADDIN

AALAADIN [37], organizasyon modellerini tanımlayan bir meta-modeldir. Şekil 3.9'de görüldüğü gibi üç temel prensibe dayanır;

- Etmen
- Rol
- Grup



Şekil 3.9 : Temel model, [37] den uyarlanmıştır.

Etmen grup içersinde bir rolü oynayan iletişim halindeki bir yapı olarak değerlendirilmiştir. Tek etmen için bir içyapı sınırlaması ya da tanımlaması yoktur. Bu geliştiriciye etmenin yapısını belirlemekte bir serbestlik getirmektedir.

Rol grup içindeki etmenin işlevlerinin veya servislerinin kavramsal sunumudur. Her etmen birden fazla gruba atanmış rolün sorumluluklarını alabilmektedir. Etmen bir rolü almak istediğinde grup içinde istekte bulunur, kendisine rol verilir veya verilmez. Rol verme kararı geliştirici tarafından ortaya konulacak yöntem ile belirlenir. Rol şu özelliklerle belirlenir;

- Tekillik: Rol grup içersinde tekil veya çoğul olabilir. Tekil rolleri grup içerinde tek etmen alabilir.
- Yeterlilik: Etmenin grup içersinde belirli bir rolü alabilmesinin koşullarını tanımlar.
- Yetenek: Rolü alan etmene verilen yetenekleri ifade eder.

Grup yöneticisi rolü grubu yaratan etmene verilen özel bir roldür. Bu role sahip etmen gruba kabul veya rolü alma isteğini değerlendiren ve karar veren etmendir.

Grup etmenler kümesidir. Görevleri başarıya ulaştırmak ve yürütmek açısından atomiktirler. Fazla sayıda etmenin bulunduğu işlemlerde grup yararlı bir kavram olarak karşımıza çıkmaktadır.

- Bir etmen birden fazla gruba üye olabilir
- Grup herhangi bir etmen tarafından kurulmuş olabilir. Gruba katılmak isteyen etmen gruba kabul isteğinde bulunur.
- Grup bir yerel veya birden fazla makineye dağıtılmış olabilir.

Yukarıdaki belirtilen özelliklere bağlı olarak analiz ve tasarım aşamasında grup ve kurum yapısı dikkate alınarak analiz ve tasarım yapılır. Grup ve kurum yapısına kısaca bakılacak olursa.

Grup yapısı, grup içindeki bütün rolleri ve etkileşimleri ifade eder. Şu özelliklerle tanımlanır:

- Grup içindeki etmenlerin alabileceği bütün roller,
- Roller arasındaki bütün etkileşimler,
- Etkileşimi sağlamak için belirli bir dil.

Kurum yapısı ise çoklu etmen sistemini ortaya çıkaran bütün grup yapılarının kümesidir şeklinde tanımlanır.

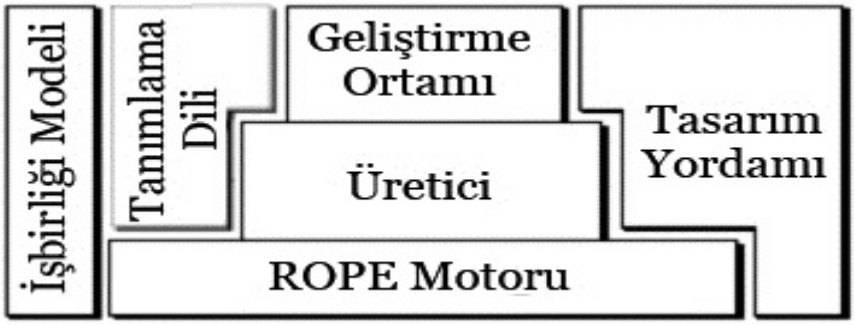
AALAADIN'de roller etmen gösterimine sıkı bir şekilde bağlıdır; bu durum geliştiriciler rolleri etmenden bağımsız tanımlamak istediklerinde sorun yaratabilmektedir. Roller ve etmenlere bir bütün olarak dışarıdan bakılarak değerlendirilir. Etmenler ise içinde bulundukları organizasyonda yerine getirdikleri fonksiyona göre anlamlıdırlar, tek başlarına mimarileri ve davranışları dikkate alınmazlar.

3.3.3 ROPE

ROPE [38], rollerin birinci-sınıf varlıklar olarak tanımlamaktadır. Role Yönelik Programlama Ortamı (Role Oriented Programming Environment) olarak adlandırılan bütünleşik, işbirliği halindeki etmenlerden oluşan sistemler geliştirmek için kullanılabilecek bir ortam önerilmektedir. Şekil 3.10'de bu ortamın temel bileşenleri görülebilir.

Geliştirme ortamı (Development Environment) grafiksel işbirliği işlemini tasarlamak için görsel tasarım bileşenini içerir. Üretici (Generator) belirtme dili ile betimlenen işbirliği işlemini çalıştırılabilir ve çalışabilir işbirliği işlemine dönüştürür. İşbirliği modeli (Cooperation model) ve belirtme dilinin (specification language) amacı dönüşebilir işbirliği yapısını desteklemektir. ROPE engine işbirliği işleminin dağıtık olarak çalışması için gerekli bütün kodları sağlar. Bu işlemi katılımcı etmenlere çok az sınırlama getirerek yerine getirir.

ROPE işbirliği işlemlerini öne çıkarmıştır. Sistem değiştirilmeden, yeniden başlatılmadan ya da mevcut işlemler durdurulmadan yeni işlemlerin sisteme eklenebilmesi amaçlanmıştır.



Şekil 3.10 : ROPE bileşenleri, [38] den uyarlanmıştır.

Rol, gerekli izinler kümesi R, verilen izinler kümesi G, A servisini yürütüp S durumuna ulaşan çizgeye sahip olan bir varlık olarak tanımlanmıştır. A etmenin davranışını göstermektedir. S işbirliği işleminin hangi aşamaya geldiğini gösterir.

Roller etmenler ile işbirliği işlemleri arasına etkin bir arayüz sağlamaktadır. Böylece etmenler kolayca işbirliği işleminde belirtilen görevleri yerine getirebilmektedir. İşbirliği işlemlerinde roller etmenlerin kavramsallaşmış halidir yani etmen ile işbirliği işlemi ayrılmıştır. Böylece etmen yapısının değişikliği işbirliği işlemini, işbirliği işleminin değişikliği de etmeni etkilememiş olur.

Etmenler rollerin başlattığı hareketleri yerine getirirler. Etmen ile işbirliği işlemleri ayrılmış ve araya rol eklenmiştir. Etmenler sistemde bir rolün gerektirdiği işlemleri yerine getirmektedirler.

Tasarım aşamasında ROPE içinde etmenler kendi yeteneklerini ve izinlerini belirten servisleri sağlar. Rol bu servisleri tanımlar. İşbirliği işlemleri ve öntanımlı kurallar için petri-net genişletilmiş ve rol kavramı eklenmiştir.

ROPE gerçekleme ve çalışma aşaması için bütünleşik bir altyapı sağlamıştır. ROPE yaklaşımı daha çok işbirliği halindeki etmen sistemlerini öne çıkarmaktadır. Bütün işlemler işbirliği halinde değerlendirilmiştir. Birbiriyle yarış halindeki etmen sistemleri dikkate alınmamıştır. ROPE içersinde rollerin dinamikliği ve rol etmen ilişkisinin dinamikliği desteklenmemiştir.

3.3.4 TRUCE

TRUCE [39] (Tasks and Roles in a Unified Coordination Environment) etmenlerin koordinasyonu için betik temelli dil çatısıdır. Adapte olabilen, heterojen, uyumlu ve birden fazla işbirliği içinde olunabilen bir yapıyı hedef almıştır. Bu amaçlar için TRUCE, etmenlerin rollerine uygulanabilen koordinasyon kurallarının tanımlanmasına izin vermiştir. Koordinasyon işlemi koordinatör etmenler aracılığı ile yürütülmekte ve böylece koordinasyon diğer kısımlardan ayrılmıştır. TRUCE dilinin bazı özellikleri şöyledir;

- Koordinasyonu işlemini koordinatör etmene atayarak basitleştirmiştir.
- Koordinasyon diğer fonksiyonel kısımlardan ayrı olarak geliştirilebilir
- Pek çok koordinasyon yapıcı ilkellere sahiptir,

- Koordinasyon işleminin basit şekilde ifade edilmesini sağlamaktadır. Bu ifadeler daha sonra karmaşık işlemlere çevrilmektedir,
- Heterojen ortamlarda programlama görevlerini basitleştirir,
- Dinamik işbirliği için dinamik yetenekler sağlamaktadır.

Ayrıca koordinasyon, protokol ve politikalar olarak tanımlanmıştır.

TRUCE'de rol etmenin görünümünü betimler, diğer detayları saklar ve etmen sadece alınan rolle yeteneklere sahip olabilir. Etmen kendisine atanan role uygun görevleri yerine getirebilir. Rol ayrıca etmene belirli koordinasyon protokollerinin bağlanması olarak görülebilir.

İlkel Adımlar	Tanım
truce	truce ilan etmek ve tanımlamak için
cast	belirtilen rol kümesini ilan etmek için
packet	paket formatını ilan etmek ve tanımlamak için
protocol	olay tabanlı kurallar kümesini ilan etmek ve tanımlamak için
set	özelliğe değer atamak için
tell	başka role bilgi göndermek için
wait	rolü bekleme durumuna getirmek için
proceed	bekleyen rolün çalışmasına devam etmesini söylemek için
retract	mevcut özellik veya kuralın tanımını kaldırmak için
recover	tanımın kaldırılmışını kurtarmak için
trigger	truce çalıştırmak veya olay sinyali yükseltmek için
step	belirtilen sıralı adımları ilan etmek ve tanımlamak için
if	koşullu sıralı adımları çalıştırmak için
echo	normal çıktıya metin yazmak için
break	geçerli içeriği atlamak için

Şekil 3.11 : TRUCE ilkel basamakları, [39] dan uyarlanmıştır.

TRUCE içerisinde her komut step olarak adlandırılır ve çalıştırıcılara sahiptir. Bu çalıştırıcılar komutu yerine getirmek durumunda olan rollerdir. Step'i yerine getirirken 4 taraf olabilir.

- Executor: hareketi yerine getiren

- Receiver : hareketin sonucunu alan
- Object: hareket sürecinde kullanılan
- Observer: işleme katılmayan ama hareketin sonucunu kullan

TRUCE çatısının ilkel adımları Şekil 3.11'de görülmektedir. Bu adımlara ek olarak kullanıcı bu adımları kullanarak yeni adımlar oluşturabilir.

TRUCE yapılandırılmamış çok sayıda sanal etmenden oluşan etmen uzayı tanımlar. Bu etmenler içinden işbirliği grupları oluşur ve işbirliği betiğini yorumlayarak koordinasyon gerçekleşir. Sorumluluklar ve yetenekler çerçevesinde etmenlerin hangi role sahip olması gerektiği belirlenir. Betik başlamadan önce işbirliği halindeki etmenlere gerekli tek bir rol bağlanır.

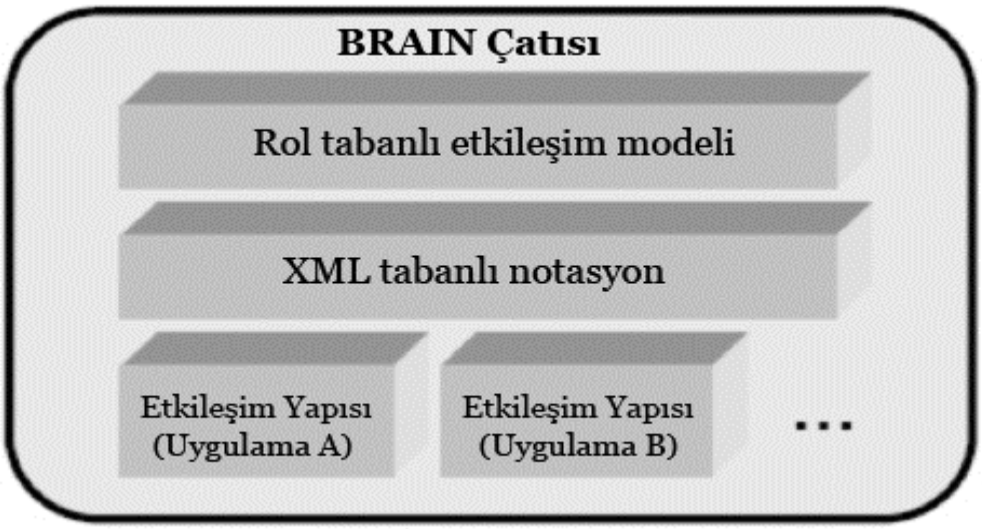
TRUCE dinamik olarak etmenlerin farklı koordinasyon protokollerine uyum göstermesine imkan vermektedir; bu onu açık bir işbirliği çatısı yapmaktadır. TRUCE koordinasyon kavramına yoğunlaşmıştır. TRUCE sadece gerçekleme seviyesine odaklanmıştır. Rol-etmen ilişkisi dinamik değil statik bir yapıdadır.

3.3.5 BRAIN

BRAIN [40], etmen temelli uygulamaların geliştirilmesinin farklı aşamalarındaki etkileşimi kapsamak için basit ve genel bir rol tabanlı etkileşim modeli önermektedir. Etkileşimleri roller şeklinde modellemenin belirli avantajları vardır. Bu durum öncelikle etmen tabanlı uygulamaların geliştirilmesinde algoritmik sorunlar ile etkileşim sorunlarının ayrılmasını (ilgilerin ayrılması) sağlar. İkinci olarak, deneyimlerin ve çözümlerin yeniden kullanılmasını izin verir. Daha önce başka uygulama için geliştirilen roller benzer uygulamalarda yeniden kullanılabilir. Son olarak roller bir çeşit tasarım kalıbı olarak değerlendirilebilir. Sistemdeki etkileşim temelde olay-hareket (event-action) şeklinde değerlendirilmiştir.

Şekil 3.12'de genel yapısı görülen BRAIN çatısı;

- Basit ama genel bir rol tabanlı etkileşim modeli,
- XML tabanlı rolleri betimlemek için geliştirmenin farklı aşamalarında kullanılacak esnek bir notasyon,
- Daha önceki model ve notasyona dayalı etkileşimin gerçekleme altyapısı önermektedir.



Şekil 3.12 : BRAIN çatısı, [40] dan uyarlanmıştır.

Rol-tabanlı etkileşim modeli: Rol, yetenekler kümesi ve beklenen davranışlar olarak modellenmiştir. Burada yetenekler etmenlerin yapacağı hareketlere, beklenen davranışlar ise etmenlerin alacağı olaylara karşılık gelmektedir. Roller tekrar kullanılabilir ve etmenden bağımsız olarak tanımlanmıştır. BRAIN çatısında iki etmen arasındaki etkileşim birinci etmenin hareket yapması ile başlar, bu hareket olaya dönüştürülür ve ikinci etmen etmene bildirilir, ikinci etmenin bu olaya karşılık bir hareket yapması ile sonuçlanır.

XRole notasyonu: XRole notasyonu etmenlerin rollerini tanımlamak için kullanılır. XRole yapısının temel amacı etmen etkileşiminin geliştirmenin değişik aşamalarında desteklemektir. XRole XML tabanlı bir notasyondur. XRole, rolü tanımlamak için kullanılan *temel bilgiler*, rolün yapmaya izinli olduğu hareketleri gösteren **izinli** *hareketler* ve etmenin alması beklenen muhtemel olayların gösterildiği **tanınan** *olaylar* olmak üzere üç temel bölümden oluşur. XRole kullanımının avantajları;

- Daha önce anlatılan etkileşim modelinin gerçeklenmesini sağlar,
- Geliştiriciye yazılım geliştirmenin değişik aşamalarında gösterimi farklı formatlarda sunma ve kullanma imkanı sağlar. Örneğin analiz aşamasında XRole gösterimi HTML olarak sunulabilir. Ayrıca çalışma sırasında XML belgesi kullanılarak etmenin rolleri kolayca değiştirilebilir.
- XRole belgesi XML temelli olduğu için farklı alt yapılarla etkileşim sırasında kullanılabilir.

BRAIN yapısı analiz, tasarım ve gerçekleme adımlarında etmenleri dikkate almış, rollerin etkileşimin ve tanımlamasını XML tabanlı bir notasyon ile basitleşmiştir. BRAIN sistem içersindeki etkileşimlere ağırlık vermiştir. Geliştirdiği notasyon ve etkileşim modeli geliştiriciler için kolaylık içermektedir. BRAIN içerisinde etmenler aynı anda birden fazla rol alma ve rolleri aktarma gibi özelliklere sahip değildir.

3.3.6 RoMAS

RoMAS [41], etmeni tekil olarak değil rol ile birlikte var olan varlık olarak değerlendirmiş ve şu rol tanımını yapmıştır.

- Rol, kavramsal olarak etmenin içinde yer aldığı etkileşimler ve belirli bir yönde gösterdiği değişim için bir sınırlamadır. Etmen rolle tanımlanmış sınırlamalara göre davranır.
- Rol, gerçekleme bakış açısı ile rol etmenin sahip olduğu belirli özellikler ve davranışların bir bütünüdür.

Bu tanımın sonucunda RoMAS siteminde aşağıdaki özellikler mevcuttur;

- Etmen aynı anda birden fazla role sahip olabilir
- Etmen dinamik olarak rollerini değiştirebilir
- Etmen bir işlem yapar durumda iken yaparken rol bir işlem yapmaz.
- Rol izole değildir, rolle ilgili başka roller vardır.
- Rol, etmenin diğer etmenlerle etkileşimi için yol sağlayan bir pencere gibidir
- Rol yeniden kullanım için olanak sağlamaktadır.

RoMAS rol organizasyonu modelini kullanmıştır. Analiz ve tasarım aşamalarında roller ve aralarındaki etkileşimler belirlenerek organizasyon oluşturulur. RoMAS aşağıdaki özelliklere sahip rol-tabanlı modelleme dili önermektedir.

- Kavramsal ve dilbilimsel yönden rol ile etmen ayrılmıştır
- ÇES'in geliştirilmesi süreci boyunca roller bulunur.

Ana geliştirme süreci aşağıdaki adımlardan oluşur.

1. Kullanım durumlarını oluştur
2. Kullanım durumlarından rolleri belirle

3.Rol organizasyonunu oluştur

4.Her rol için etmen yoksa (5) adımına git, varsa

- Rolü etmene bağla
- Etmen ile rol arasındaki ilişkinin özelliklerini belirle
- (6) adımına git

5.Rollere göre etmenleri yarat. (4) adımına git

6.Etmenler ve bağlı roller için kodları üret

Kullanım durumu oluşturma: Sistemdeki olayları ve etkileşimler göstermek için kullanım durumları oluşturulur.

Rolleri belirleme: Roller kullanım durumları ve senaryolarda kullanılan sözcükler kullanılarak belirlenir.

Rol organizasyonunu oluşturma: Rol izole bir yapı değildir. Roller diğer rollerle etkileşim halinde olabilir. Ayrıca roller arasında nesne ilişkileri (miras, birleştirme gibi) de bulanabilir. Bu ilişkilere göre rol organizasyonu belirlenir.

Rolü etmene bağlama: Her rol için bir etmen var olabilir ya da olmayabilir. Var olan etmen adı, özellikleri, hareketleri ve davranış kuralları ile tanımlanır. Var olmayan etmenler rollerden tanımlanır. Etmen rollerini dinamik olarak değiştirebilir. Bunu sağlamak için sonlu durum makinesi kullanılır. Sonlu durum makinesinde durum rol olarak değerlendirilir, geçişler üzerine koşullar eklenerek etmenin hangi durumda role sahip olacağı belirlenir.

RoMAS analiz ve tasarım aşamasında etmenler, roller ve aralarındaki ilişki için genişletilmiş UML notasyonu kullanmış ve bu süreçler için yöntemler ortaya koymuştur. Ancak gerçeklemenin ne şekilde olacağı ve bu adımda rollerin ne şekilde gösterileceği belirtilmemiştir.

3.4 Rol Tabanlı Etmen Sistemlerinin Değerlendirilmesi

Bir önceki bölümde özellikleri anlatılan rol-tabanlı etmen sistemleri farklı farklı noktalara yoğunlaşmaktadır. Gaia analiz ve tasarım noktasında yoğunlaşırken TRUCE işbirliğine odaklanmıştır. Bu sebeplerden dolayı değerlendirme yapabilmek

için bazı kriterler koymamız gerekmektedir. Bu kriterlerden bir kısmı roller ile ilgili olurken bir kısmı sistemin geneli ile ilgilidir. Kriterlerin bir kısmı [42]'dan alınmıştır.

Çizelge 3.1 : Rol tabanlı etmen sistemlerinin değerlendirilmesi.

	Gaia	**AALAADIN**	**ROPE**	**TRUCE**	**BRAIN**	**RoMAS**	**EWSP**
Analiz	Bir model öneriliyor.	Bir model öneriliyor.	Yok	Yok	Var	Var	Var
Tasarım	Bir model öneriliyor.	Bir model öneriliyor.	Bir model öneriliyor.	Yok	Var	Var	Var
Gerçekleme	Yok	Var	Bir model öneriliyor.	Bir model öneriliyor.	Var	Yok	Var
Notasyon	Var	Yok	Var	Yok	Var	Yok	Var
Rol-Etmen İlişkisi	Roller sadece analiz adımında. İlişki statiktir.	Etmen gruba atanmış rollerden alabilir.	Rol etmen ilişkisi statiktir.	Rol etmen ilişkisi statiktir.	Dinamik ilişki var. Aynı anda tek rol alınır.	Dinamik ilişki var. Aynı anda birden fazla rol alınır.	Dinamik ilişki var. Aynı anda birden fazla rol alınır.
Genişletilebilir	Hayır	Evet	Evet	Evet.	Evet	Evet	Evet
Diğer Yapılarla Etkileşim	Destek yok.	Destek yok.	Destek yok.	Destek yok.	Destek yok.	Destek yok.	Destek var.

- **Analiz:** Analiz adımın ne şekilde yürütüleceği belirtiliyor mu?
- **Tasarım:** Tasarım aşamasında rollerin ne şekilde belirleneceği, hangi rollerin kullanılacağı belirtiliyor mu, yöntem önerilmiş mi?
- **Gerçekleme:** Gerçekleme aşamasında roller kullanılıyor mu?
- **Notasyon**: Rolleri, rollerin özelliklerini ve roller arasındaki ilişkileri daha iyi anlatabilmek ve anlayabilmek için kullanılacak bir notasyon var mı?
- **Rol-Etmen İlişkisi:** Dinamik ve değişen sistemlerde roller ile atandıkları etmenler arasındaki ilişki ne şekilde yürütülüyor? Çalışma esnasında etmen rol değiştirebiliyor mu? Aynı anda birden fazla rol atanabiliyor mu? Rol etmenden etmene aktarılabiliyor mu?

- **Genişletilebilir:** Sistem oluşturulduktan sonra yeni eklemeler yapılarak genişletme imkanı var mı?
- **Diğer Yapılarla Etkileşim:** Oluşturulan başka etmen sistemleri ya da programlarla etkileşim halinde çalışabiliyor mu?

Belirtilen kriterlere göre rol tabanlı etmen sistemlerinin değerlendirildiği Çizelge 3.1 oluşturulmuştur.

Tez çalışmasında ortaya konulan rol, etmen ve servisle ilişkilendirilmiş dinamik bir yapıdır. Rol sistemin analiz adımında rollerin ve bağlı olan servislerin ortaya çıkarılması için yoğun olarak kullanılmaktadır. Roller xml notasyonu üzerinden tanımlanmaktadır ve tasarım aşamasında yer almaktadır. Gerçekleme adımında roller kullanılmakta ve etmenlere roller verilip alınmaktadır. Roller aynı zamanda çalışma zamanında da dinamik olarak etmenlere verilebilmektedir. Rol tanımları için EWSPX isimli bir notasyon ortaya konulmuştur. XML kullanılarak roller ve ilişkili oldukları servisler rahatlıkla tanımlanabilir. Roller ile etmenler arasında dinamik bir ilişki bulunmaktadır. Etmenler birden fazla rol alabilmektedir. Sistem başlangıcında ya da çalışma zamanında etmenlere roller sanal firma yöneticisi tarafından verilip alınabilmektedir. Sistem kolay bir şekilde genişletilebilir. Sisteme farklı sanal firmalar çalışma başlangıcı sonrasında kendi rolleri ile dahil olabilmektedir. Sistem diğer yapılarla web servisleri üzerinden etkileşim halindedir.

4. ETMEN TABANLI WEB SERVIS PAZARI (EWSP)

Etmen tabanlı web servis pazarı, rol tabanlı sanal firmaların dinamik bir şekilde oluşumu, sanal çalışanların firmalara katılımı, sanal firma servislerinin ihale yoluyla satılması ve servislerin çağrılabilmesi için bir altyapı sunmaktadır. Sanal firmalar ve servislerinin oluşumu için çoklu etmen sistemleri, roller ve web servisi kavramları kullanılmıştır. Etmenler ile web servisi teknolojileri bir araya getirilmiş ve etmen servislerinin ve fonksiyonlarının web servisi olarak sunulması sağlanmıştır. Roller servislerle bağdaştırılmış ve çoğu zaman birden fazla servisi içerecek şekilde tanımlanmıştır. Sanal firmalar ise çalışanların/yöneticilerin alabileceği belirli rollere ve sanal çalışanlara sahip çoklu etmen sistemleri olarak tanımlanmıştır. Sanal firma yöneticisi sanal çalışanlarına roller atayabilmektedir. Edindiği rol aracılığıyla servis sunma yeteneği kazanan çalışan servisini sisteme kayıt ettirmekte ve bu şekilde müşterilerden gelecek servis istekleri karşısında açılacak ihalelere katılma hakkı elde etmektedir. EWSP sisteminin bir diğer önemli özelliği de, gerek duyulduğunda dinamik olarak sanal firmaların servislerinin bir araya getirilip yeni "bütünleşik web servisleri " oluşturulmasına olanak sağlamasıdır.

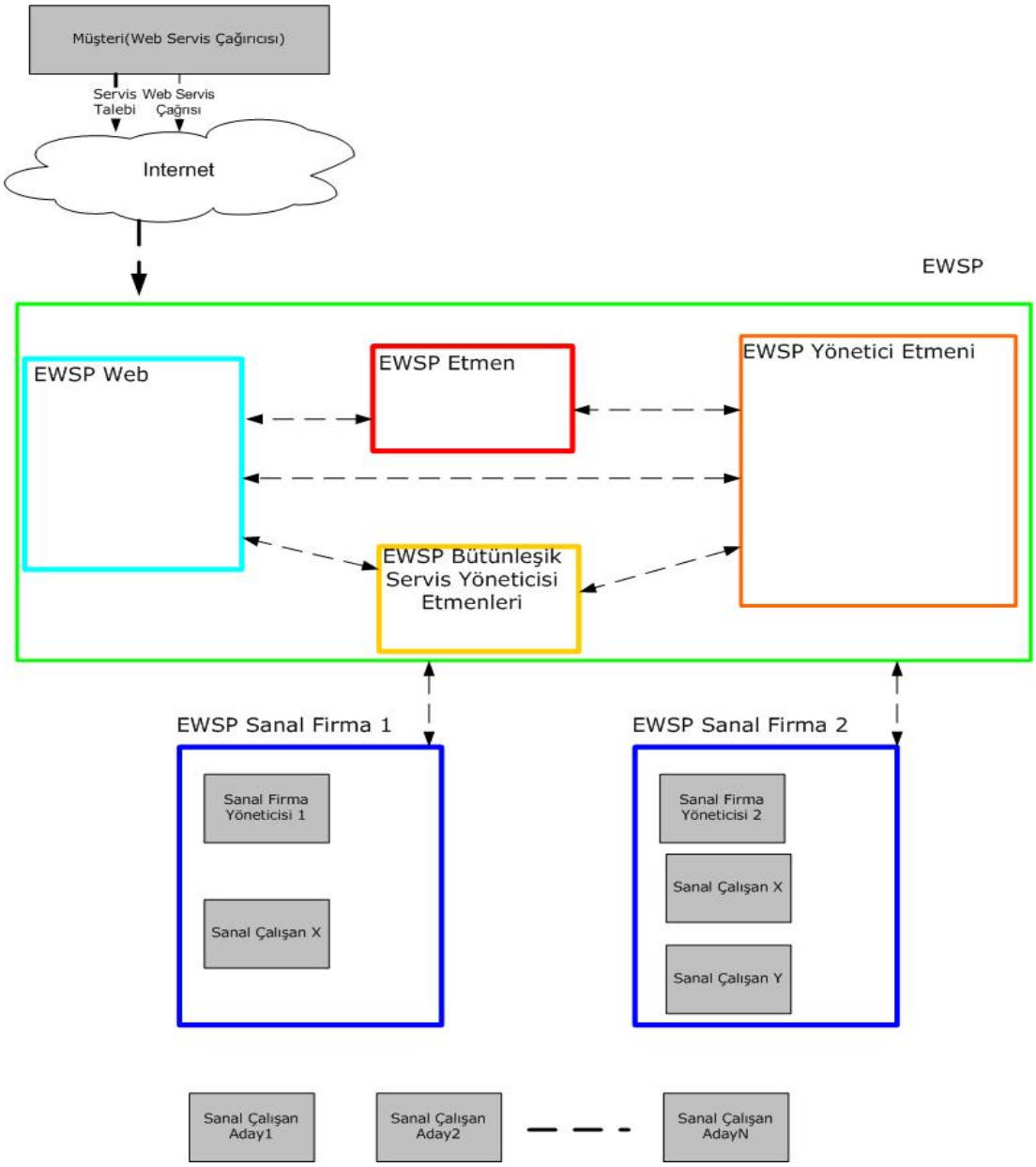
Sistemin temel özellikleri şunlardır;

- Roller, servisler bütünü olarak tasarlanmıştır. Bu sayede servis sunma yeteneği rollerle birlikte farklı çalışanlara verilebilmektedir. Servislerin yönetilmesi ve sanal çalışanlara dağıtılması sanal firma yöneticisi tarafından kolay bir şekilde yapılabilmektedir.
- Etmenler, sanal firma yöneticisi, sanal firma çalışanı ve sistemi yöneten yönetici etmenler olarak sistemde yer almıştır.
- Roller, sanal çalışan/yönetici etmenlerine dinamik olarak verilip, alınabilmektedir. Bu sayede, sanal çalışan hızlı bir şekilde servis yetenekleri kazanmaktadır.
- Etmen ile web servisi entegrasyonu sağlanmıştır. Etmen fonksiyonları ve servisleri web servisi olarak tanımlanabilmektedir.

- Sanal firmalar servislerini web servisi olarak müşterilere açabilmektedir.
- Müşterilere, aradıkları özelliklere sahip bir web servisinin bulunabilmesi için ihale açma olanağı sağlanmıştır.
- Aranan web servisini bir tek etmen sağlayamıyor ise, birden fazla etmenin işbirliği ile istenen özellikler yerine getiren bütünleşik servis oluşturma olanağı sağlanmıştır.
- İhale sonrası müşterinin servis kullanımı izlenebilmektedir.

4.1 EWSP Sistem Yapısı

EWSP sistemi internete açık bir web sunucusu ve JADE etmen çatısı üzerinde çalışmaktadır. Şekil 4.1'de EWSP sistemi genel mimarisi gösterilmektedir. İnternetten gelen web servisi için ihale veya yürütme çağrılarını yöneten EWSP Web uygulaması sistemin dış dünya ile etkileşimini sağlamaktadır. Müşteri, dış dünyadan web çağrısı yapabilen bir programdır. EWSP Etmeni sistemi ayağa kaldıran ve servis kayıtlarını gerçekleştiren etmendir. EWSP Yönetici etmen dış dünyadan gelen istekler için açılan ihaleleri yöneten etmendir. Sistem kullanım durumuna göre tek bir yönetici etmen gelen taleplere yeterli sürede yanıt vermeyebilir. Sistemin performansını arttırmak için birden fazla yönetici etmene ihtiyaç duyulabilir. Sistem açılışında ihtiyaç duyulan kadar EWSP Yönetici etmeni yaratılabilir. Konunun detayları yönetici etmen bölümünde aktarılacaktır. EWSP Bütünleşik Servis Yöneticisi Etmenleri, ihale sonrası bütünleşik bir servis oluşması durumunda söz konusu olan servisi yönetmek için oluşturulan etmendir. Vekil etmen, bütünleşik servisi oluşturan alt servislere ilişkin çağrıları yönetir. Sanal firma kendi servisleri, rolleri ve çalışanları olan bir organizasyondur. Sanal firma bir sanal firma yöneticisi tarafından yönetilmektedir. Sanal çalışan, aynı anda bir veya daha fazla sanal firmada görev alabilen ve firmaya katılımı ile belirli rolleri kazanmış olan etmendir.



Şekil 4.1 : EWSP sistemi genel yapı.

4.2 Sanal Firma

Sanal firma belirli bir hiyerarşik yapı içinde rollere, servislere ve sanal çalışanlara sahip olan bir sanal varlıktır.

Sanal Firma = {Sanal Firma Yöneticisi, Roller, Servisler, Sanal Çalışanlar}

Her sanal firma, bir sanal firma yöneticisi tarafından kurulur. Sanal firma kuruluş anında sadece sanal firma yöneticisinden oluşmaktadır ve yapılan başvurular sonrasında sanal firma çalışanlarını bünyesine katar. Sanal firma yöneticisi, kendisine yapılan başvuruları değerlendirir ve sanal çalışan adayını işe kabul eder

veya etmez. Sanal firma çalışanı, sanal firmaya kabul edildiği taktirde, başvurmuş olduğu rolü de alır. Bu sayede, edindiği rolün sağladığı servisleri sunma yeteneğini de kazanmış olur. Sanal firma yöneticisi açılan servis ihalelerine hangi sanal çalışanın girip girmeyeceğine de karar vermektedir.

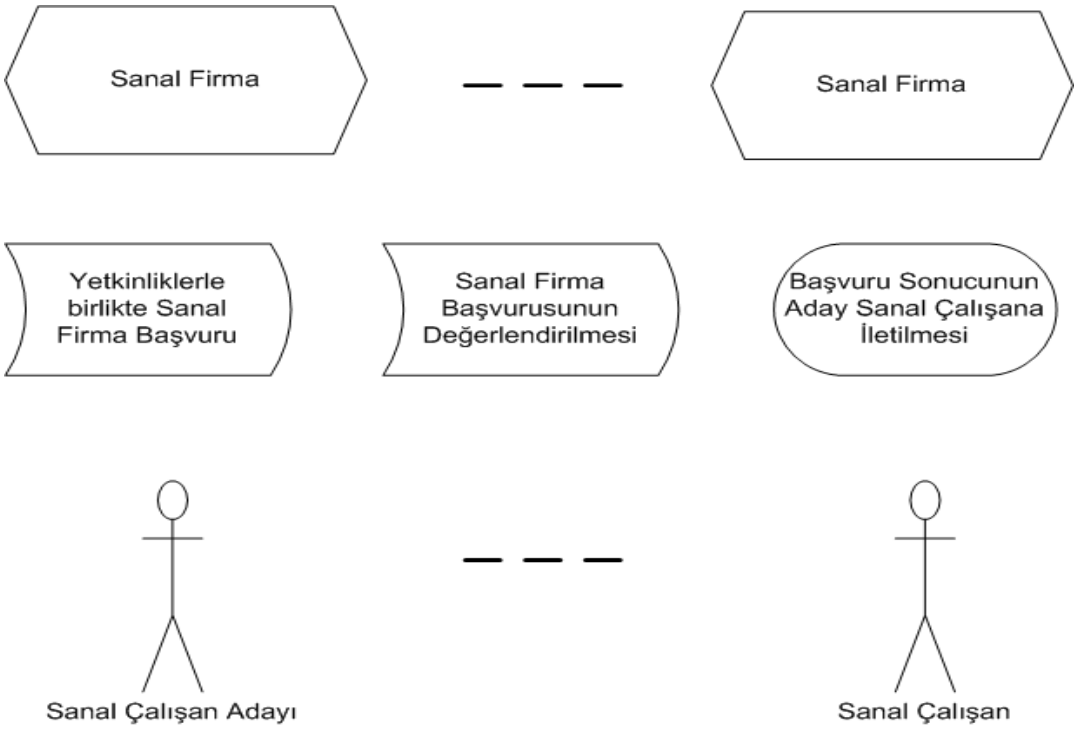
Sanal firma oluşturulmadan önce sanal firma içindeki servisler ve roller belirlenmiş olmalıdır. Bölüm 5.3'te detaylı olarak anlatılacak olan rol tanım xsd[14] yapısı ile roller, servisler ve bunlar arasındaki ilişki belirlenir. Tanımlanan roller ve servisler sanal firma kurulumu sırasında sanal firma yöneticisi tarafından sisteme yüklenir.

4.2.1 Sanal firma başvuru süreci

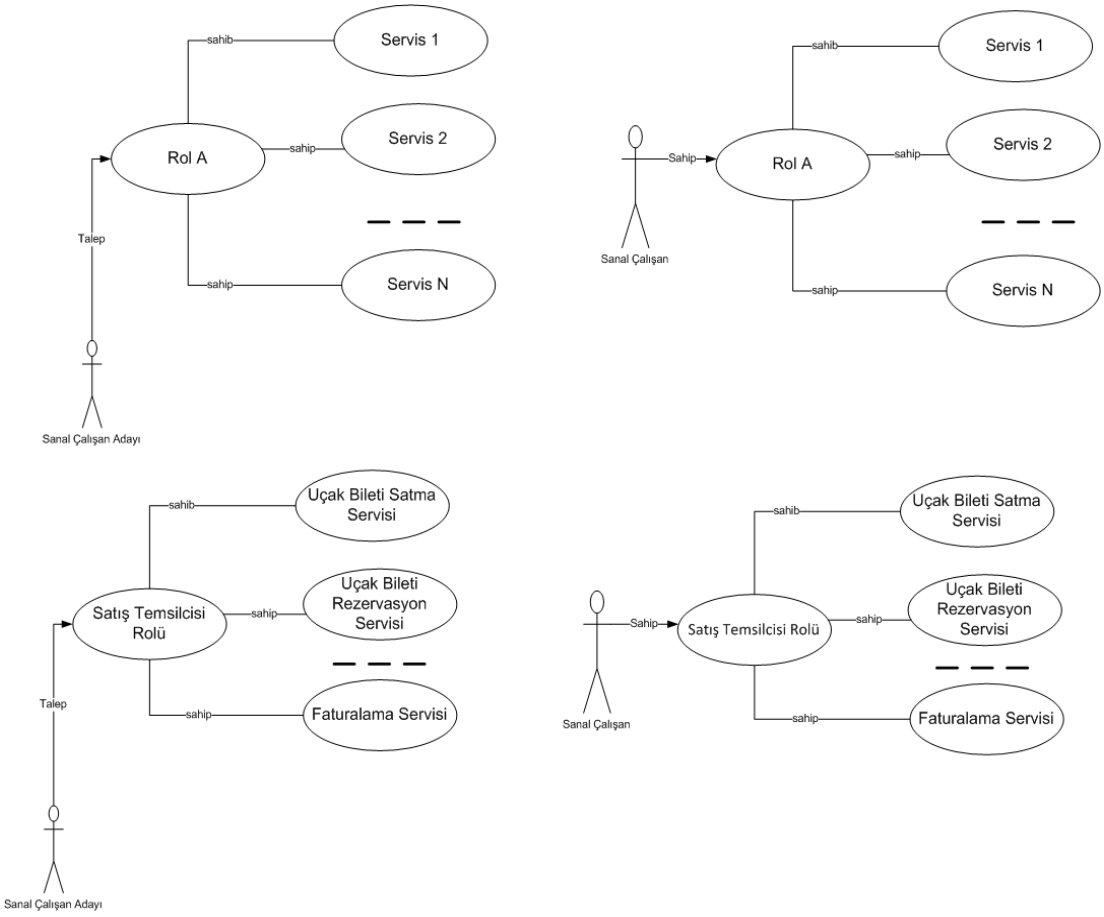
Sanal firma çalışan adayı sanal firmaya başvuru işleminde bulunur. Başvuru işlemi Şekil 4.2'de gösterildiği şekilde yürümektedir. Sanal çalışan sanal firma başvurusu sırasında başvuru yaptığı roller için istenen bilgileri ve yetenekleri sanal firma yöneticisine bildirir. Her rol için Bölüm 5.3'te aktarılacak olan rol tanım xsd kullanılarak belli bilgiler istenebilir. Örneğin iş tecrübesi, mezun olduğu okul, yabancı dil bilgisi gibi bilgiler istenebilir. Çağrı merkezi elemanı için İngilizce dil bilgisi istenmezken, altın müşteri temsilcisi için İngilizce dil bilgisi istenebilir. Başvuru sanal firma tarafından değerlendirilir. Sanal firma yöneticisi kendisine gelen sanal çalışan başvurusunu aşağıdaki kriterlere göre değerlendirir;

- Başvuru sırasında sanal çalışanın sağladığı bilgiler ve yetenekler. Örneğin ne kadar süredir çalışıldığı, mezun olunan okul yaş gibi rol başvurusu için gerekli olan bilgiler.
- Başvuru yapılan rolün sanal firma içinde verilebileceği rol maksimum değerinin aşılıp aşılmadığı bilgisi. Bir rolü firma içinde rol tanımı yapılırken belirlenen maksimum rol alınma adeti kadar sanal çalışan alabilir. Örneğin bir sanal firmada maksimum bir adet genel müdür rolüne sahip sanal çalışan bulunabilirken 10 adet çağrı merkezi görevlisi yer alabilir.

Gerekli kriterleri sağlıyorsa, sanal çalışan adayı sanal firmaya kabul edilir. Başvuru yapan sanal çalışan aday firmanın çalışanı olur ve başvuruda bulunduğu rolü elde eder.



Şekil 4.2 : Sanal çalışan-sanal firma başvuru.



Şekil 4.3 : Sanal çalışan rol servis ilişkisi.

Sanal çalışanın rol edinmesi Şekil 4.3'de gösterilmiştir Rolü elde eden sanal çalışan aynı zamanda o rolün sağladığı servisleri sunma yeteneğini de kazanmış olur. Kazandığı servisleri DF kayıt ettirir ve kaydedilen bu bilgilerin EWSP Etmenine iletilmesi ile söz konusu olan servisler web servisi haline dönüşmüş olur.

4.3 EWSP'de Roller

EWSP'de rol, servislerin bir araya gelmesiyle oluşan bir bütün olarak değerlendirilmiştir. Sanal firma içindeki her rol bir veya daha fazla servis ve özelliklere sahiptir.

Örneğin bir turizm firmasında **Satış Temsilcisi** rolü **Uçak Bileti Satma Servisi** ve **Uçak Bileti Rezervasyon Servisi**'ni içerebilir. Aynı şekilde **Platin Satış Temsilcisi** rolü **Uçak Bileti Satma Servisi, Uçak Bileti Rezervasyon Servisi, Müşteri Şikayeti Servisi**'ni içerebilir.

Rol={Servisler, Özellikler}

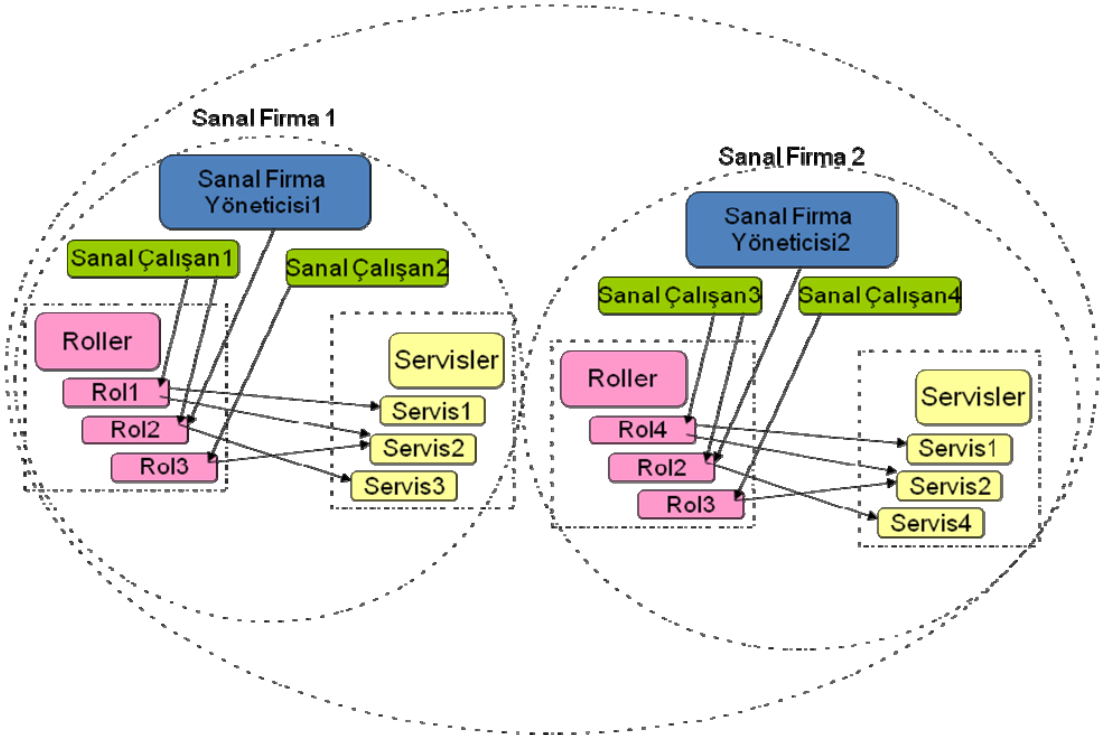
Roller bir veya daha fazla servis içerebilir. Özellikler rollerin sahip olduğu isim, role için gerekli olan kriterler gibi bilgileri göstermektedir. Servisler Bölüm 4.4'te detaylı olarak anlatılacaktır.

Sanal firma içinde yer alacak olan bütün servisler analiz ve tasarım aşamasında ortaya çıkarılır. Sanal firmanın amaç ve işlevleri göz önüne alınarak, firmada bulunması gereken roller belirlenir. Bu roller çalışanlara da atanacağı için sorumluluk ve yetkileri gözeterek oluşturulmalıdırlar. Roller içinde yer alacak servisler daha önce tanımlanmış olan servisler arasından belirlenir. Aynı servis bir veya daha fazla rol içinde yer alabilir. Rol servis ilişkisi dinamik olarak tasarlanmıştır. Çalışma zamanında rol servis ilişkisi belirlenmektedir.

Roller ile servisler arasında yer alan ilişki Bölüm 5.3'te anlatılacak olan xsd yapısı ile tanımlanmaktadır. Roller bir xml dosyası ile tanımlanırlar ve sanal firma kurulurken, firma yöneticisi tarafından yüklenirler. Bir sanal firmada yer alan roller Şekil 4.4'de gösterilmiştir. R firmada yer alan rollerin oluşturduğu kümedir.

$R= \{r_1, r_2, r_3, r_4\}$.

Sanal firma, sanal çalışan, rol ve servis ilişkisi Şekil 4.4'te gösterilmiştir.



Şekil 4.4 : Sanal firma rol servis ilişkisi.

4.4 EWSP'de Servisler

Bir s servisi sanal firma içinde atomik düzeyde bir görevi yerine getirir. Sanal firma kuruluşu sırasında hangi servislerin bulunacağı kararı verilir. S firmanın sunduğu servislerin oluşturduğu kümedir. Servisler bir veya daha fazla role atanmış olabilir.

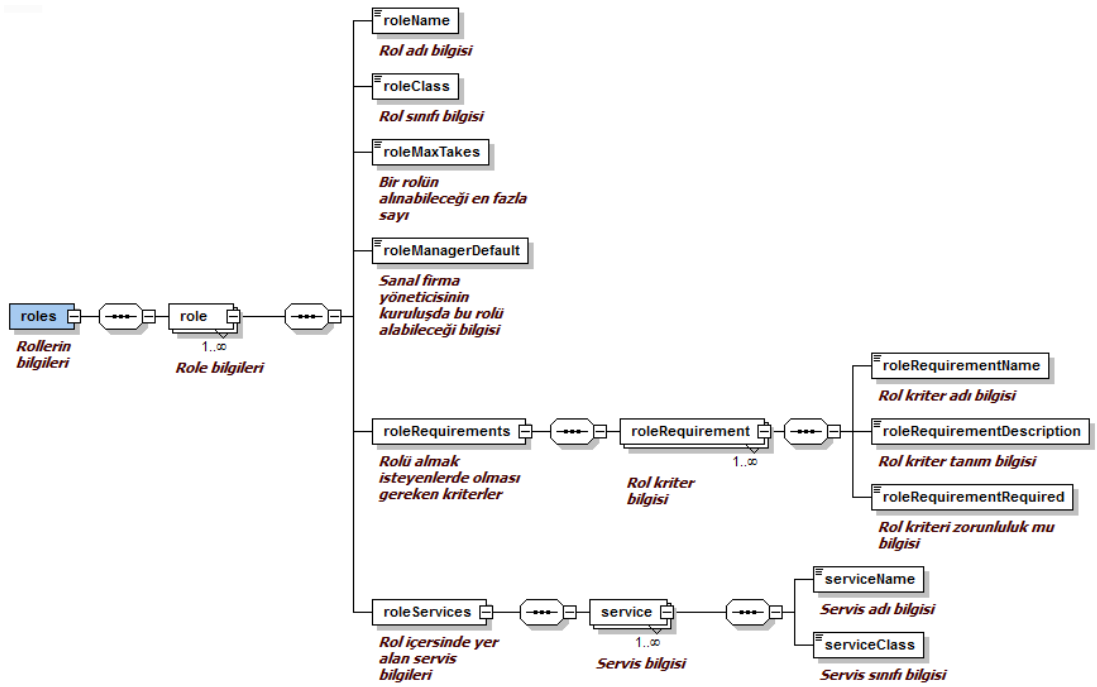
$S= \{s_1, s_2, s_3, s_n\}$

Örneğin;

$r_1=\{s_1, s_2\}$, $r_2=\{s_1, s_2, s_3\}$ olabilir.

Sanal firma içinde roller ve içerikleri firma kuruluşu sırasında belirlenir. Sanal firma içindeki işlevler değerlendirilerek servisler ortaya çıkarılır. Servisler sistemde rollere bağlı olarak tasarlanmıştır.

Sanal firma yöneticisi, firmayı kurarken tanımlı rolleri ve bunların tanımını Şekil 4.5'de gösterilen şemaya sahip bir xml dosyasından yükler. Şema ile ilgili detaylı bilgi Bölüm 5.3'te anlatılmıştır.



Şekil 4.5 : Rol tanım XSD.

1)**roleName**: Rol adını ifade eder.

2)**roleClass:** Rolün gerçeklendiği sınıfı ifade eder. Örneğin **com.ewsp.example.role.SatısTemsilcisi.**

3)**roleMaxTakes:** Rolü alabilecek maksimum sanal çalışan sayısını ifade eder. Bir sanal firma içinde bu rolü alabilecek maksimum sayıdaki sanal çalışanı gösterir. Örneğin bir firmada maksimum bir adet genel müdür bulunur, ancak on adet satış temsilcisi bulunabilir.

4)**roleManagerDefault:** Sanal firma yöneticisinin varsayılan rolü olup olmadığını belirtir. Sistem yöneticinin sistem başlaması sırasında edinmesi gereken rolleri belirtir. Sistem yüklemesi sırasında bu değeri bir olarak belirlenmiş roller sistem yöneticisi tarafından edinilir.

5)**roleRequirements:** Rolün bir sanal çalışan tarafından edinilmesi için arana kriterleri ifade eder. Bu kriterler sanal çalışan başvurusu sırasında başvuru ekranında çıkar. Sanal çalışan başvuru sırasında bu kriterlerin değerlerini sağlar. Sanal çalışan rol başvuru formu bu bilgilere göre oluşturulur. Bu eleman xml içinde birden fazla olabilir.

a)**roleRequirementName:** Rol kriterinin adını ifade eder. Örneğin **iş tecrübesi süresi** gibi.

b)**roleRequirementDescription:** Rol kriterinin tanımını, uzun anlatımını ifade eder.

c)**roleRequirementRequired:** Bu kriterin başvuru formunda doldurulmasının ve başvurunun zorunluluğunu ifade eder.

6)**roleServices:** Rol içinde yer alan servisleri ifade eder. Rolün içinde yer alan servisler tanımlanır. Bu eleman xml içinde birden fazla olabilir.

a)**serviceName:** Servisin adını ifade eder.

b)**serviceClass:** Servisin gerçeklendiği sınıfı ifade eder. Örneğin **com.EWSP.example.service.DomesticHotelReservation**

Rol xsd tanımına uygun örnek xml dosyası Bölüm 5.3'te gösterilmiştir.

4.5 Örnek Uygulama Servis-Rol-Etmen İlişkisi

Servis, rol ve etmenler arasındaki, yukarıdaki bölümlerde ifade edilen ilişki örnek bir sanal firma aracılığı ile anlatılacaktır. Unicorn uçak bileti rezervasyonu ve satışı yapan bir firmadır. Unicorn değişik havayollarına ait biletleri rezerve edebilir ve satabilir. Müşteriler Unicorn firmasını arayarak uçuşlar hakkında bilgi veya satış sonrası destek alabilirler. Unicorn dört kişinin çalıştığı küçük bir firmadır. Unicorn'un EWSP sisteminde çalışmasını ve servislerini satmasını sağlayabiliriz. Servis tabanlı rol sistemine uygun olarak Unicorn'u modelleyebiliriz.

Öncelikle Unicorn'un iş tanımına ve firmanın iç görevlerine uygun olarak sunulacak servisler şöyle belirlenir:

s_1= Uçak Bileti Satma Servisi

s_2= Uçak Bileti Rezervasyon Servisi

s_3= Proje Yönetimi Servisi

s_4= Personel Yönetim Servisi

s_5= Ödeme Servisi

s_6= Faturalama Servisi

s_7= Yardım Masası Servisi

Sistemdeki bütün servisler $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$

Bir sonraki aşamada sistemdeki roller belirlenir:

r_1= Yönetici Rolü

r_2= Satış Yöneticisi Rolü

r_3= Satış Temsilcisi Rolü

r_4= Müşteri Hizmetleri Rolü

Sistemdeki bütün roller R= $\{r_1, r_2, r_3, r_4\}$

Sistemdeki roller ile servisler problem tanımına uygun olarak eşleştirilir. Bir rol bir veya daha fazla servise sahip olabilir.

Yönetici Rolü= { Proje Yönetimi Servisi, Personel Yönetim Servisi } → $r_1 = \{s_3, s_4\}$

Satış Yöneticisi Rolü = { Uçak Bileti Satma Servisi, Uçak Bileti Rezervasyon Servisi, Proje Yönetimi Servisi } → $r_2 = \{s_1, s_2, s_3\}$

Satış Temsilcisi Rolü = { Uçak Bileti Satma Servisi, Uçak Bileti Rezervasyon Servisi, Ödeme Servisi, Faturalama Servisi } → $r_3 = \{s_1, s_2, s_5, s_6,\}$

Müşteri Hizmetleri Rolü = { Yardım Masası Servisi } → $r_4 = \{s_7\}$

Bir firma yönetici, bir tane satış grubu yöneticisi ve iki adet satış temsilcisi Unicorn'da çalışmaktadır. Firma çalışanları etmenler ile temsil edilirler ve kendilerine aşağıda gösterilen roller atanır.

Etmenler;

$Etmen_1$: EWSP Sanal Firma Yöneticisi

$Etmen_2$: Satış Grup Yöneticisi, Sanal Firma Çalışanı

$Etmen_3$: Satış Temsilcisi, Sanal Firma Çalışanı

$Etmen_4$: Satış Temsilcisi, Sanal Firma Çalışanı

$Etmen_5$: Çağrı Merkezi Elemanı, Sanal Firma Çalışanı

Etmenler atanan roller :

$Etmen_1 = \{r_1\}$

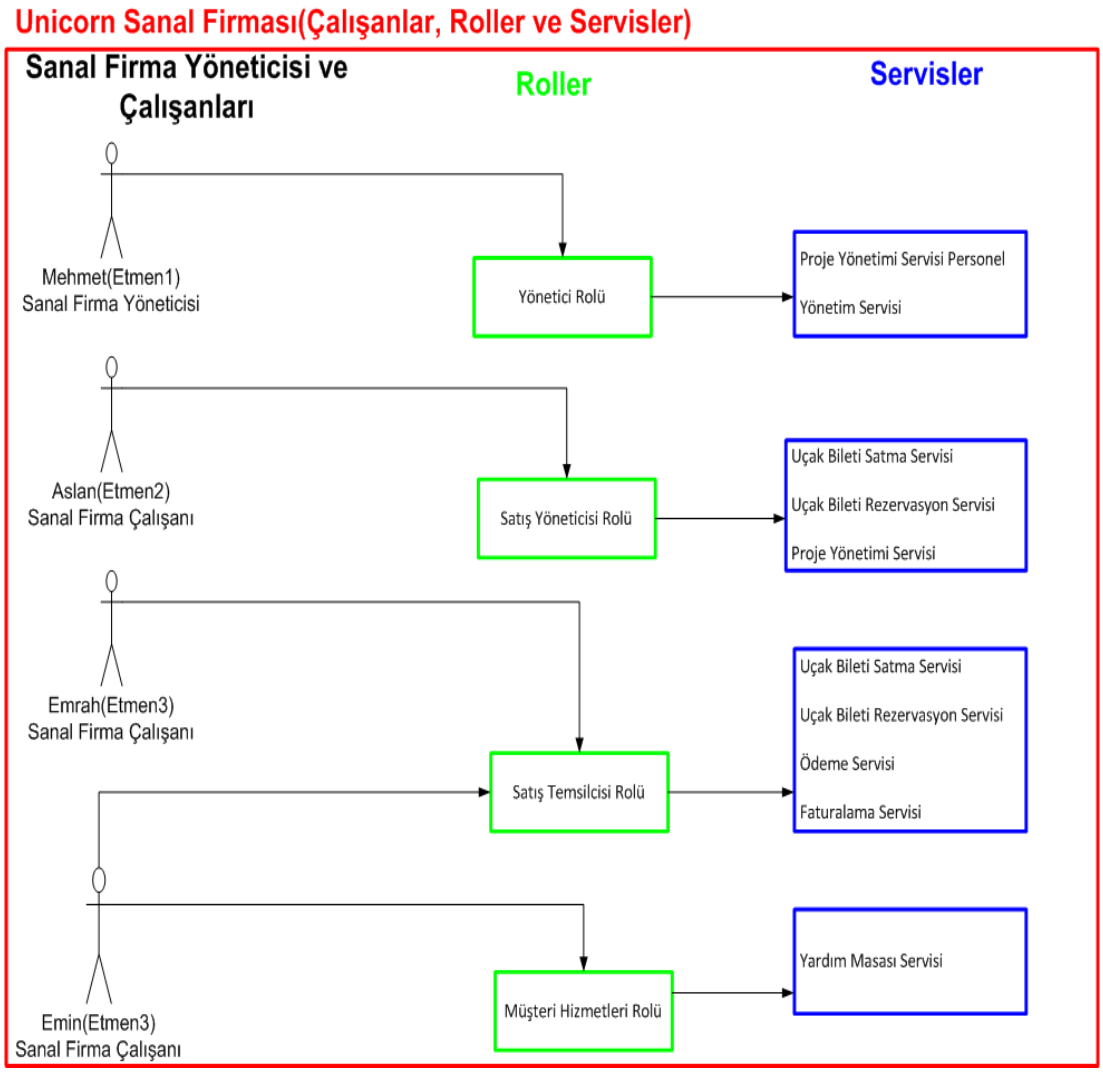
$Etmen_2 = \{r_2\}$

$Etmen_3 = \{ r_3\}$

$Etmen_4 = \{ r_3 \}$

$Etmen_5 = \{ r_3 , r_4 \}$

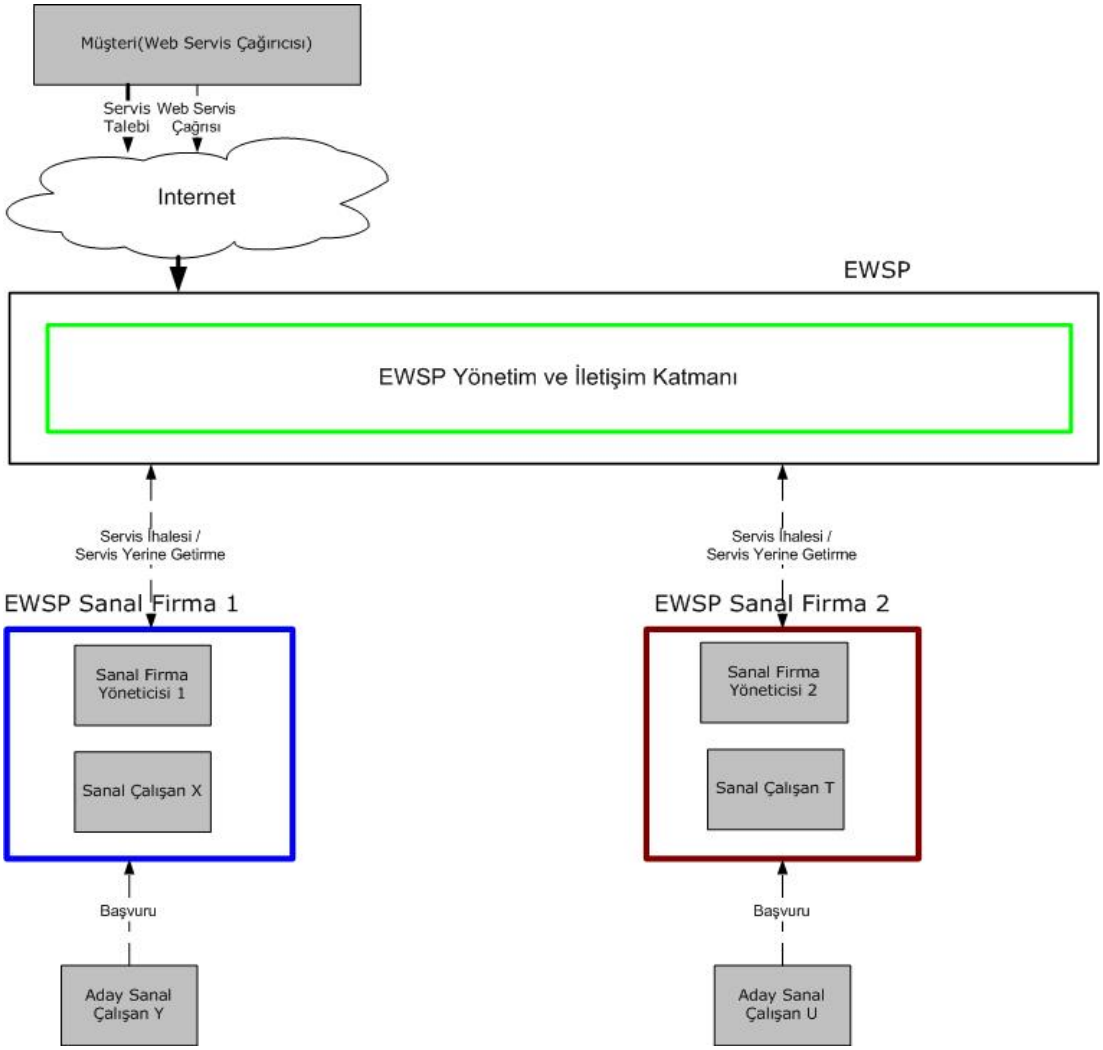
Bu örnekteki bazı servisler web servisi olarak kayıt edilebilir. Örneğin s_1, s_2, s_7. Bu servisler müşterilere satılabilir. Anlatılan örneğe göre oluşan sanal firma, roller, rollerin dağılımı, servisleri ve servisleri roller arasında dağılımı Şekil 4.6'da gösterilmiştir.



Şekil 4.6 : Örnek sanal firma rol servis.

5. ETMEN TABANLI WEB SERVIS PAZARI (EWSP) MİMARİSİ

Sanal firmalar için alt yapı sunan EWSP genel mimarisi Şekil 5.1'de gösterilmiştir. Müşteri sistemden ihtiyaç duyduğu servisi sağlamaya çalışan ve bunun için ödeme yapan varlıktır. Bu web servisi çağrısı yapabilen bir programdır. Müşteri EWSP sistemine Yönetim ve İletişim Katmanı aracılığı ile erişmektedir. Bu katman sistemi yöneten ve dış dünya ile etkileşimi sağlayan katmandır. Sanal firmalar kendi içlerinde organize olmuş, çalışanlara, rollere ve servislere sahip varlıklardır. Sanal firmaların müşteri ile etkileşimi EWSP Yönetim ve İletişim Katmanı aracılığı ile gerçekleşmektedir.



Şekil 5.1 : EWSP genel mimari.

5.1 EWSP Mimarisi Bileşenleri

Şekil 5.2'de gösterilen EWSP mimari bileşenleri;

- Web sunucusu içinde çalışan ve EWSP sistemin dış istemciler ile internet üzerinden haberleşmesini sağlayan EWSP Web,
- Web ile etmenler arasında geçit şeklinde çalışan EWSP Etmeni,
- İstemcilerden gelen servis bulma talebini yerine getiren ve ihale düzenleyen EWSP Yönetici Etmeni
- Bütünleşik servisleri ve çağrılarını yöneten EWSP Bütünleşik Servis Yöneticisi Etmenleri,
- Sanal firma içersinde (bir firma çalışanı veya yöneticisi) yer alan, web servisi sağlayan, ihalelere katılan ve servis isteklerini yerine getiren Sanal Firma ve Çalışan Etmenleri,
- Web servisi talebinde bulunan ve onu kullanan istemcilerdir.

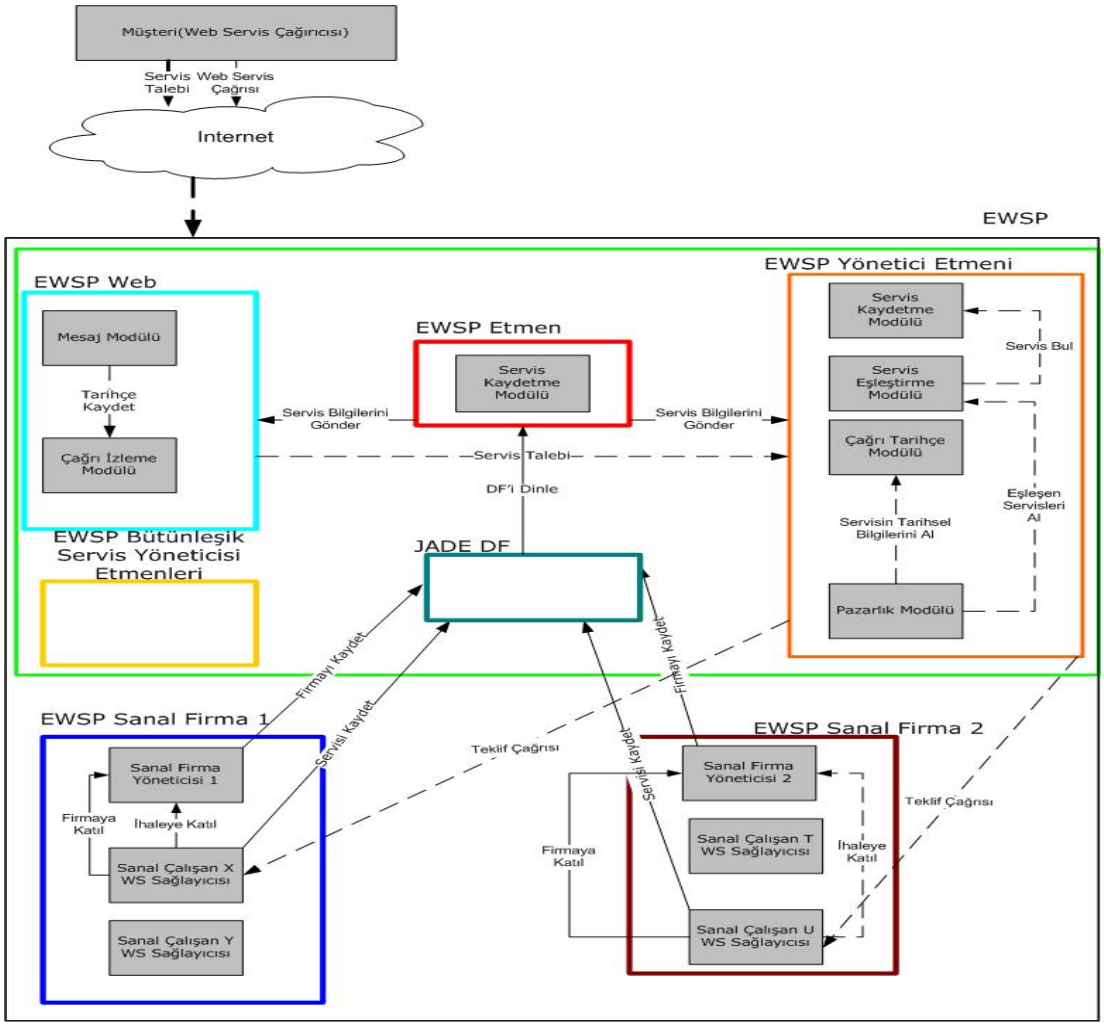
Bu bölümde bu bileşenlerden her biri detaylı olarak aktarılacaktır.

5.1.1 EWSP web

Web sunucusu içinde çalışan EWSP Web, EWSP sisteminin dış istemciler ile internet üzerinden haberleşmesini sağlayan önyüzdür. İstemcilerin mesajları web servis çağrıları şeklinde EWSP Web aracılığı ile etmenlere ulaştırılmaktadır. Benzer şekilde, istemciler çağrılarının sonuçlarını EWSP Web aracılığı ile almaktadırlar. EWSP Web gelen servis bulma çağrılarını EWSP Yönetici etmenlerine gönderir. EWSP Web bir yük dengeleme algoritması uygulayarak o an en az işe sahip olan yönetici etmene yeni gelen talebi iletir. EWSP Web, servis bulma servisi hariç, gelen çağrılar için Bölüm 5.4'te detaylı olarak aktarılacak olan **awarId (ihale sonucunu belirten tekil değer)** kontrolü yapmaktadır. İstemcilerin talepleri sonucunda yapılan ihale sonucunu ifade etmek üzere **awardId** kullanılmaktadır.Servis çağrıları için geçerli bir **awardId** veremeyen çağrılar en başta engellenmektedir. Bu yetkisiz erişim çağrılarını engellemek ve yapılan ihale sonuçlarını izleyebilmek için kullanılmaktadır. EWSP Web modülleri ve sorumlulukları aşağıda anlatılmıştır.

Mesaj modülü: EWSP'ye gelen çağrılar web servisi çağrısı olarak yapılmaktadır. Gelen mesajların etmen servis çağrılarında kullandıkları XML formatına

dönüştürülmesi bu modül tarafından gerçeklenir. Bu modül HTTP/SOAP olarak gelen mesajları XML mesajlarına dönüştürür. Aşağıda bir SOAP mesajı ve dönüşüm sonunda oluşan XML mesajı görülmektedir.



Şekil 5.2 : EWSP mimarisi.

SOAP mesajı:

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:ServiceFinder">

<soapenv:Header/>

<soapenv:Body>

<urn:findService soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

```
<clientName xsi:type="xsd:string">Bulut</clientName>
<sServiceName xsi:type="xsd:string">Araba Kiralama</sServiceName>
<sServiceDescription xsi:type="xsd:string">Araba Kiralama
</sServiceDescription>
```


…….


```
</urn:findService>
</soapenv:Body>
</soapenv:Envelope>
```


XML mesajı:


```
<service>
<serviceName>ServiceFinder</serviceName>
<operation><operationName>findService</operationName>
<operationParameters>
<clientName>Bulut</clientName>
<sServiceName> Araba Kiralama</sServiceName>
<sServiceDescription Araba Kiralama Servisi</sServiceDescription>
```


…….


```
</operationParameters>
</operation>
</service>
```


Benzer şekilde, etmenlerden gelen XML mesaj yanıtları da SOAP mesajları haline dönüştürülerek alıcılara iletilir. Mesaj Modülü'nün temel fonksiyonları şunlardır:

- HTTP/SOAP istek mesajlarını almak/vermek
- İstekleri ve ilgili bilgileri EWSP etmenine aktarmak

- İsteklerin sonuçlarını SOAP mesajına dönüştürmek
- HTTP/SOAP cevaplarını hazırlamak ve onları istemcilere göndermek

Çağrı izleme modülü: Bu modül servis çağrılarını izleme işlemini yerine getirmektedir. EWSP Yönetici etmeni talep etmesi durumunda çağrı bilgileri gönderilir. Servis çağrısı ile ilgili olarak aşağıdaki bilgiler izlenir:

- Çağrı süresi: Servis yerine getirme işleminin süresi
- Bulunurluk: Servis talebinde bulunulan etmenin aktif/çalışıyor olup olmadığı, talebe yanıt verip vermediği bilgisi
- Hatalar: Servis çağrısı ile ilgili oluşacak hatalar

Sistemdeki findService hariç servis çağrıları servisi ifade eden, aramalarda bulunmasını sağlayan kelime veya kelime öbeklerinden oluşan anahtar değerleri içermek durumundadır. Anahtar değerler kullanılarak elde edilen çağrı istatistikleri EWSP Yönetici Etmenine gönderilir.

5.1.2 EWSP etmeni

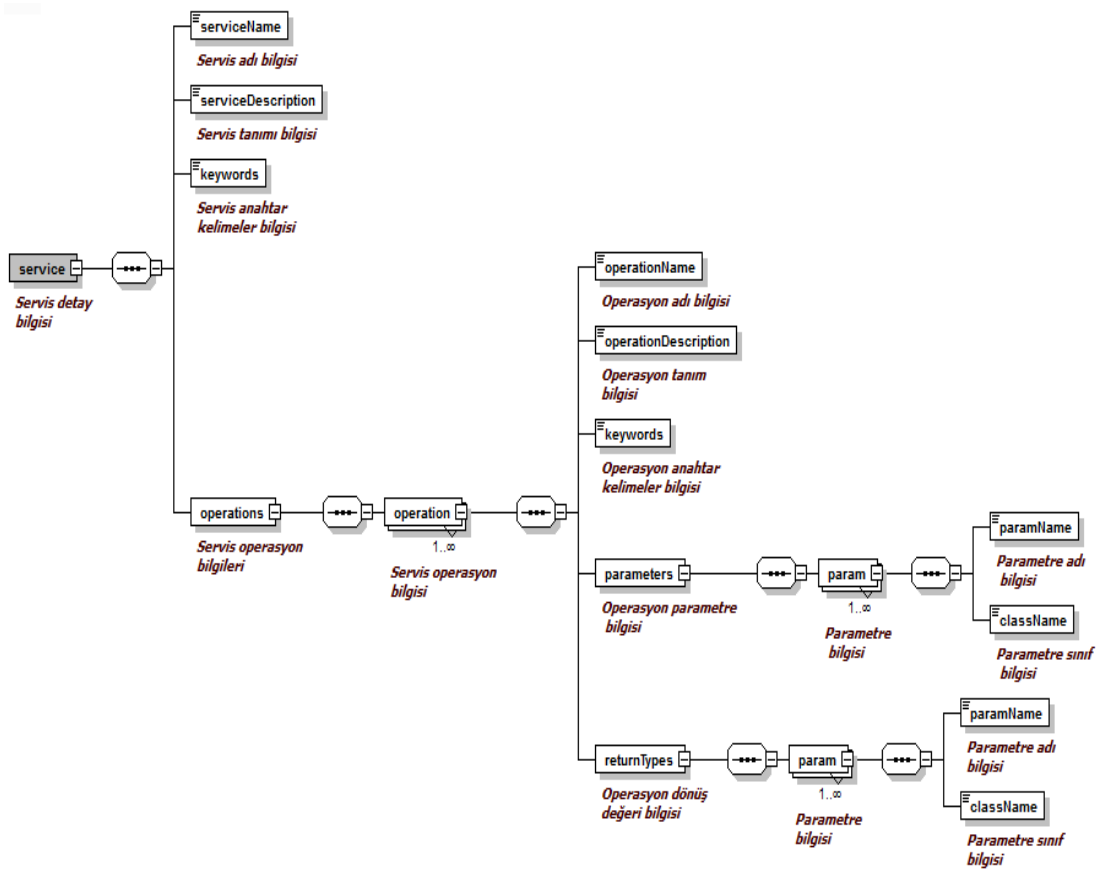
EWSP Etmeni web ile etmenler arasında geçit şeklinde çalışır. Etmenlerin servislerini web servisi için uygun formata (web servisi tanımına (WSDL)) dönüştürür. Web servisi tanıma dönüştürdüğü servislerin sahibi etmenler hakkındaki bilgileri EWSP Yönetici etmenine bildirir. EWSP Etmeni aşağıdaki işlemlerden sorumludur:

- Sistemin ilk açılış aşamasında EWSP Yönetici etmenini yaratmak
- EWSP Web'den aldığı bilgileri ve yapılması gereken işlemleri ilgili etmene yönlendirmek. Etmenin yaptığı işlemin sonucunu almak.
- JADE DF'ye kendini kayıt etmek ve yeni kayıt olan/kayıt silen etmenler hakkında uyarılar almak.
- DF'ye kayıt olan etmenlerin servislerini web servisler için WSDL'e çevirmek ve yayınlamak.
- Servis olarak kayıt olan etmenler hakkındaki bilgileri EWSP Yönetici etmenine göndermek.

Etmen servislerini xml formatında DF etmenine iletirler. Etmenler servislerini DF iletirken Şekil 5.3'de gösterilen xsd şemasına bağlı kalarak xml belgesi oluştururlar.

Xsd içinde yer alan **service** servisin adını belirten **serviceName**, servisin tanımını gösteren **serviceDescription**, servisin bulunmasını ve tanımlanmasını kolaylaştıran anahtar kelimeleri belirten **keywords** ve servisin sahip olduğu operasyonları belirten **operations** elemanlarından oluşmaktadır.

operations bir veya daha fazla tekrar eden **operation** elemanından oluşmaktadır. **operation** operasyonun adını belirten **opeationName**, operasyonun tanımını gösteren **operationDescription**, operasyonun anahtar kelimelerini belirten **kewyords**, operasyon giriş parametrelerini belirten **parameters** ve dönüş parametrelerini belirten **returnTypes** elemanlarından oluşmaktadır. **parameters** bir veya daha fazla tekrar eden **param** elemanından oluşmaktadır. **param** parametrenin adını gösteren **paramName** ve parametrenin sınıfını belirten **className** elemanlarından oluşmaktadır. **returnTypes** bir veya daha fazla tekrar eden **param** elemanından oluşmaktadır.



Şekil 5.3 : Servis XSD.

Servis xsd şemasını uygun olarak üretilmiş bir servis tanımı xml Şekil 5.3'de gösterilmiştir.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<service>
<serviceName>StudyAbroadRegistration</serviceName>
<serviceDescription>find oversea/overseas university or high school,register oversea/overseas
unviservity or high school, search for university or high school that meets search
criteria</serviceDescription>
<keywords>register overseas university, register oversea university,register overseas high school,
register oversea high school,  find university, find high school, study abroad </keywords>
<operations>
<operation>
<operationName>registerUniversity</operationName>
<operationDescription>registerUniversity</operationDescription>
<keywords>register university</keywords>
<parameters>
<param>
<paramName>studyId</paramName>
<className>java.lang.String</className>
</param>
</parameters>
<returnTypes>
<param>
<paramName>studyId</paramName>
<className>java.lang.String</className>
</param>

                    .........

<param>
<paramName>returnCode</paramName>
<className>java.lang.Integer</className>
</param>
</returnTypes>
</operation>
<operation>
<operationName>findUniversity</operationName>
<operationDescription>findUniversity</operationDescription>
<keywords>abroad university, find university</keywords>
<parameters>
<param>
<paramName>countryName</paramName>
<className>java.lang.String</className>
</param>
<param>
<paramName>majorName</paramName>
<className>java.lang.String</className>
</param>
</parameters>
<returnTypes>
<param>
<paramName>studyId</paramName>
<className>java.lang.String</className>
</param>

                    .........
```

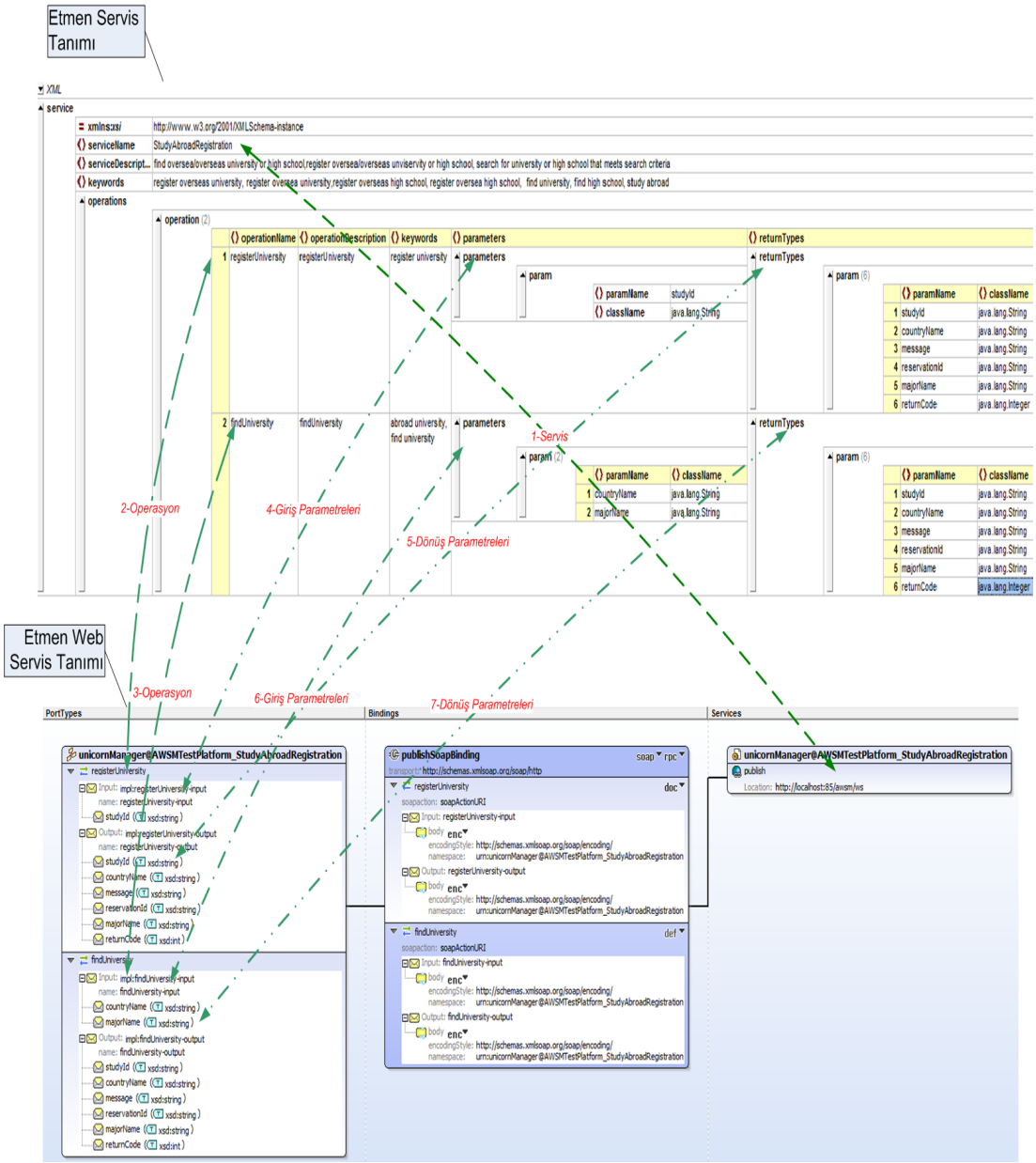
```
<param>
<paramName>returnCode</paramName>
<className>java.lang.Integer</className>
</param>
</returnTypes>
</operation>
</operations>
</service>
```

EWSP Yönetici etmen servis tanımı xml'ni aldıktan sonra bu xml tanımından web servis tanımını (WSDL) belgesini oluşturmaktadır. Örnek xml servis tanımından aşağıdaki şekilde bir WSDL belgesi oluşmaktadır.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:impl="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
name="unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"
targetNamespace="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration">
<wsdl:types>
<xsd:schema
targetNamespace="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"
xmlns:impl="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:annotation/>
</xsd:schema>
</wsdl:types>
<wsdl:message name="registerUniversity-input">
<wsdl:part name="studyId" type="xsd:string">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="findUniversity-input">
<wsdl:part name="countryName" type="xsd:string">
  </wsdl:part>
<wsdl:part name="majorName" type="xsd:string">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="registerUniversity-output">
<wsdl:part name="studyId" type="xsd:string">
  </wsdl:part>

                    .........

<wsdl:part name="returnCode" type="xsd:int">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="findUniversity-output">
<wsdl:part name="studyId" type="xsd:string">
  </wsdl:part>

                    .........

<wsdl:part name="returnCode" type="xsd:int">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration">
<wsdl:operation name="registerUniversity">
```

```
<wsdl:input name="registerUniversity-input" message="impl:registerUniversity-input">
  </wsdl:input>
<wsdl:output name="registerUniversity-output" message="impl:registerUniversity-output">
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="findUniversity">
<wsdl:input name="findUniversity-input" message="impl:findUniversity-input">
  </wsdl:input>
<wsdl:output name="findUniversity-output" message="impl:findUniversity-output">
  </wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="publishSoapBinding"
type="impl:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration">
<wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="registerUniversity">
<wsdlsoap:operation soapAction="soapActionURI" style="document"/>
<wsdl:input name="registerUniversity-input">
<wsdlsoap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"/>
</wsdl:input>
<wsdl:output name="registerUniversity-output">
<wsdlsoap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="findUniversity">
<wsdlsoap:operation soapAction="soapActionURI"/>
<wsdl:input name="findUniversity-input">
<wsdlsoap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"/>
</wsdl:input>
<wsdl:output name="findUniversity-output">
<wsdlsoap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="unicornYönetici@EWSPTestPlatform_StudyAbroadRegistration">
<wsdl:port name="publish" binding="impl:publishSoapBinding">
<wsdlsoap:address location="http://localhost:85/EWSP/ws"/>
</wsdl:port>
</wsdl:service>

</wsdl:definitions>
```

Servis xml tanımının servis WSDL dönüşümü Şekil 5.4'de görsel olarak gösterilmiştir. Servis adı bilgisi Servis adına dönüştürülmekte, servis operasyonları WSDL **port type operation** yapılarına dönüştürülmektedir. Ayrıca operasyon giriş ve çıkış parametreleri, port type operation giriş çıkış değerlerine dönüştürülmektedir. Servis tanımı parametrelerinin sınıf tipleri WSDL'e uygun tiplere dönüştürülmektedir.



Şekil 5.4 : Servis tanımı WSDL dönüşümü.

5.1.3 EWSP yönetici etmeni

EWSP Yönetici etmeni web istemcisinin talep ettiği servis bulma isteklerini yerine getirmeye çalışır. Bu etmen EWSP etmeni tarafından yaratıldıktan sonra kendi sunduğu **findService** servisini sisteme web servis olarak kayıt ettirir. Değişik kriterlere göre web servisi bulmak isteyen web servisi istemcileri bu servisi kullanarak isteklerine uygun servisleri bulmaya çalışırlar. EWSP Yönetici Etmeni gelen web servisi taleplerini değerlendirir, istenen özelliklere sahip uygun web servisini bulmak için gerekiyorsa ihale yapar ve sonuçları talepte bulunan istemciye gönderir. Bölüm 5.4'de ihalenin ne şekilde yapıldığı detaylı olarak anlatılacaktır.

Sistemdeki EWSP yönetici etmen sayısı parametrik bir değerdir. Çok fazla sayıda gelen servis bulma talepleri karşısında tek bir EWSP Yönetici etmeni yeterli olmayabilir. Yüksek miktarda servis talebini karşılamak için birden fazla EWSP yönetici etmeni yaratılabilir. Sistem üzerinde gelecek talep Sistem yapılandırmasında EWSP etmen sayısı ayarlanabilmektedir. EWSP Yönetici etmenine servis bulma taleplerinin sıralı (yük dağılmalı) gönderilmesi EWSP Web sorumluluğundadır. EWSP Yönetici etmenlerinden bir tanesi birincildir ve servis kayıtlarını bu yapar. Diğerleri sadece servis bulma sürecinde görev almaktadırlar. EWSP Yönetici etmeni, ihale işini kazanan bütünleşik servis ise bütünleşik servisi sunmak üzere EWSP Bütünleşik Servis Yöneticisi Etmenini yaratırlar. EWSP Yönetici Etmeni aşağıdaki modüllere sahiptir.

Servis Kaydetme Modülü: Bu modül web servislerini kayıt etme ve silme işlemlerinden sorumludur. Bu modül servis sağlayıcı etmenin kimliği, servis adı, servis tanımı gibi servisler hakkındaki bilgileri tutar. Diğer modüller servis bilgilerini almak için bu modülle etkileşim halindedir. Servis kayıtları ve silmesi için aşağıdaki çağrılar kullanılmaktadır;

registerService(serviceName,agentId,serviceInformation)

deregisterService(serviceName,agentId)

Servis Eşleştirme Modülü: Bu modül talep edilen servis kriterlerine uygun aday servisleri bulmak ile yükümlüdür. *serviceName*, *serviceDescription*, *keywords* alanları ve daha önceki benzer aramaları kullanarak eşleşen servisleri bulmaya çalışır. Talep edilen kriterlere uygun servisleri bulmak için aşağıdaki çağrılar kullanılmaktadır.

matchByServiceName(serviceName)

Aynı isme sahip servisleri bulur.

matchByServiceDescription(serviceDescription)

Benzer tanımlara sahip servisleri bulur.

matchByKeyword(keywords)

Benzer anahtar kelimeler ile tanımlanmış servisleri bulur.

Bütün servis istekleri ve sonuçları sisteme kayıt edilir. Bu bilgi daha sonra benzer kriterler ile yapılan servis taleplerinde yapılan eşleştirme süreçlerinde kullanılır. Son iki fonksiyon benzerliği kendilerine girdi olarak verilen sözcüklerin eşleşme oranlarından yola çıkarak bulur. Anahtar kelimelerin yüzde kaç oranında benzediği zaman servisin aday servis olarak kabul edileceği sistem parametresi olarak tanımlanmıştır.

Aranan servis tekil servisler arasında bulunamamış ise servis bütünleşik servis olarak aranır. Bütünleşik servisler arasında yapılan arama anahtar kelimeler üzerinden yapılmaktadır.

AK_{aranan}= Aranan anahtar kelime. Servis isteğinde bulunan müşteri tarafından sağlanır.

AK_{aday1}= Aday servisin sağladığı anahtar kelimeler. S_1 sağladığı aday kelimeler.

AK_{aday1} belirli bir oranda (sistemde tanımlanmış) AK_{aranan} içeriyorsa AK_{aday1} bütünleşik servis adayı olarak belirlenir.

$AK_{aranan\ yeni} = AK_{aranan} - AK_{aday1}$

$AK_{aranan\ yeni}$ diğer adaylar arasında aranır. Eğer sistemde tanımlanmış oranda uyum sağlayan bir aday servis S_2 bulunursa, S_1 ve S_2 talep edilen servisi sağlayabilir olarak değerlendirmesi yapılır ve ihaleye katılırlar. Böylece aday servis şu şekilde oluşur;

$S_{aday}= S_1 + S_2$ olarak belirlenir.

Çağrı Tarihçesi Modülü: Bu modül erişilebilirlik oranı ve yanıt süresi gibi gerçekleşmiş servis çağrılarına ait performans verilerini toplamak ve depolamaktan sorumludur. Bu modül EWSP Web ile servis çağrı detaylarını alabilmek için etkileşim halindedir. Etmen aktiviteleri yeni ihalelerde kullanılmak üzere başarım değerlendirmesi (örneğin sunulan servisler ne kadar tatmin edici) yapmak ve ihale kazanan etmenlerin yerine getirmedikleri taahhütlerini belirlemek için izlenmektedir. Başarım değerlendirmesi, etmenin daha önceki çağrılarında oluşan ortalama yanıt süreleri ve ortalama erişilebilirlik oranları baz alınarak yapılmaktadır. Etmenin daha önceki taahhütlerini yerine getirme oranı da değerlendirmede kullanılmaktadır. Bütün değerlendirmeler sonuca eşit oranda katkı sunmaktadır. Servis sağlayıcı etmenin geçmiş performansı aşağıdaki fonksiyon kullanılarak hesaplanmaktadır.

calculateHistoricalPerformance (serviceId)

Etmenin daha önceki taahhütlerini yerine getirme oranı da izlenmektedir. Etmenin daha önce kazandığı ancak taahhüdünü yerine getirmediği ihale sayısı ve oranları aşağıdaki fonksiyon kullanılarak hesaplanır ve çağırıcıya iletilir.

failureMeetCommitment(serviceId)

İhale Modülü: Bu modül servis ihale işlemlerinden sorumludur. Modül, istemcinin servis isteğini aldıktan sonra ihale sürecini başlatır. Modül ihale sürecini başlatabilmek için gerekli olan bilgileri toplamak amacıyla Servis Kaydetme Modülü, Servis Eşleştirme Modülü ve Çağrı Tarihçesi Modülü ile etkileşime geçer. Modül, uygun servis sağlayıcı etmenleri bulduktan sonra içlerinden en uygun olanını belirlemek için ihale sürecini ilgili etmenlere duyurur. İhale sürecinin detayları daha Bölüm 5.4'te ayrıntılı olarak anlatılacaktır.

5.1.4 EWSP bütünleşik servis yöneticisi etmenleri

EWSP Bütünleşik Servis Yöneticisi Etmenleri EWSP Yönetici etmenleri tarafından yaratılmaktadır. EWSP Yönetici etmeni, bir servis bulmak için ihale düzenlediğinde, bazı durumlarda bir servis talebinin tek bir servis ile karşılanamadığını fark edebilir. Bu durumda, ihale düzenleyen EWSP yönetici etmen uygun içerikte birden fazla servisi bir araya getirerek yeni bir bütünleşik servis oluşturmaktadır. Bütünleşik servis ihale sonucunda ihaleyi kazanırsa kendisini yönetecek bir varlığa ihtiyaç duyacaktır. Bütünleşik servisi yönetmek için EWSP Bütünleşik Servis Yöneticisi Etmeni yaratılır. Bu etmen yeni sahip olduğu servisi sisteme kayıt ettirir; böylece servis dış dünyadan çağrılabilir duruma gelir. Etmen kendisine gelen servis çağrılarını alt servis çağrılarına böler ve her bir alt servis çağrısını sırayla çağırır. Alt servisten gelen çağrı sonuçlarını alır. Aldığı bu sonuçları daha sonra birleştirerek kendi servisinin sonucunu oluşturur ve kendine çağrı yapan sisteme dönüş yapar.

5.1.5 Sanal firma ve sanal çalışan etmen

Sanal firma, içinde bir firma çalışanı veya yöneticisi yer alan, web servisi sağlayan, ihalelere katılan ve servis isteklerini yerine getiren etmendir. Etmenlerin hangi servisleri sağlayabilecekleri almış oldukları roller tarafından belirlenir. Rol bir veya daha fazla servisi sağlayabilir. Sistemde temel olarak 2 tip etmen bulunmaktadır.

Sanal Firma Çalışan/Aday Etmeni: Sistem içinde yer alan sanal firmalarda kendisine verilen rollerde çalışan etmendir. Sanal firma çalışanı etmen henüz bir

firmada çalışmıyorken sanal firma çalışan adayı konumundadır. Uygun gördüğü sanal firmaya başvuru yapabilir.

Sanal Firma Yönetici/Kurucu Etmeni: Sanal firmayı kuran ve bazı temel kararları alan etmendir. Sanal firma yönetici etmeni çalışmaya başladığı zaman rol tanımlarını sisteme yükleyerek sanal firmanın sahip olduğu rolleri belirler. Sanal firmaya yapılan iş başvurularını değerlendirir ve bunlarla ilgili son kararları verir.

Etmen yeni ihale isteği aldıktan sonra, ihaleye katılıp katılmama kararını sanal firma yönetici etmenine danıştıktan sonra verir. Sanal firma çalışanları içinde birden fazla etmen aynı role sahip olabileceği için, aynı ihale isteği birden fazla etmene gelmiş olabilir. Bu durumda, bir ihaleye aynı sanal firmadan birden fazla etmenin katılması pek istenen bir durum olmadığı için, sanal firma yönetici etmeni ihaleye kimin katılacağını belirler. Sanal firma yönetici etmeni, ayrıca o anki iş durumunu da dikkate alarak, ihaleye katılmanın uygun olmadığını da düşünebilir. Sanal firma yönetici etmeni yukarıdaki koşulları dikkate alarak ihaleye kimin katılacağını belirler. İhaleye kazanan etmen web servisine talep gelmesi halinde gerçekleştirir .

Etmenler xml olarak sisteme yüklenen rol ve servis tanımlarından web servisi oluşturabilmek için gerekli tanımları çıkarmaktadırlar. Bu servis tanımlarını yaratabilmek için Bölüm 5.2'de detaylı anlatılacak olan yöntemi kullanmaktadırlar. Bu yöntem sayesinde dinamik servis yaratma ve rol atama mümkün olabilmektedir.

Etmenler yukarıdaki işleri yerine getirebilmek için gerekli olan etkileşim protokollerini gerçekleştirmiş olmak durumundadırlar. Yeni etmen yazacak geliştiriciler bu protokolleri gerçekleştirmeli, rol yükleme ve servis yaratma arayüzlerini kullanmalıdırlar.

5.1.5.1 Servis tanımı

Roller bir veya daha fazla servisten oluşabilmektedir. Servisler rollere eklenip çıkarılabilmektedir. Web servis tanımları rol tanımına eklenmiş olan servis sınıflarından çalışma zamanı sırasında çıkarılmaktadır. Java annotation yöntemi servisleri tanımlama sırasında kullanılmaktadır. Annotation sayesinde etiketler program parçalarına iliştirilir. Derleyicinin veya JVM'nin bu etiketlere bakarak program hareketlerini yorumlar ve gerekiyorsa ek kodlar üretir. Bu özellik dinamik olarak rollerin dağıtılmasına ve roller üzerinden servislerin çıkarılmasını sağlamaktadır.

Çizelge 5.1 : EWSP'deki **annotation** sınıfları ve tanımları.

Adı	Tanımı
WebService	Web servisinin ve özelliklerini tanımlamak için kullanılır. Örneğin servis adı ve anahtar kelime gibi tanımlar.
WebServiceOperation	Web servis operasyonlarını ve özelliklerini tanımlamak için kullanılır. Örneğin operasyon adı ve tanımı gibi.
WebServiceParameter	Operasyon giriş parametrelerini tanımlamak için kullanılır.
WebServiceResult	Operasyon çıkış parametrelerini tanımlamak için kullanılır.

Sistem içinde kullanılan **annotation** sınıfları Çizelge 5.1'de tanımlanmıştır. Bunlar kullanılarak oluşturulan servis ve operasyon tanımları aşağıda gösterilmiştir.

```
  //Servis ve özelliklerini tanımlamak için Web servis annotation'ı
@WebService(serviceName="StudyAbroadRegistration",
serviceDescription="find oversea/overseas university or high
school,register oversea/overseas unviservity or high school, " +
"search for university or high school that meets search criteria",
keywords="register overseas university, register oversea
university," +
"register overseas high school, " +
"register oversea high school, " +
" find university,"+
" find high school,"+
" study abroad ")
public class StudyAbroadRegistration extends Service {
 public String serviceName="StudyAbroadRegistration";

public String getServiceName() {
return serviceName;
}


  //Operasyon ve özelliklerini tanımlamak için Operasyon annotation'ı
@WebServiceOperation(operationName="findUniversity",operationDescrip
tion="find abroad university",
keywords="abroad university, find university")

//Operasyon çıkışını tanımlamak için Servis Sonuç annotation'ı

@WebServiceResult(resultName="findUniversityResult")
```

```
  //Operasyon girişlerini tanımlamak için Servis parameter annotation'ı
public StudyAbroad
findUniversity(@WebServiceParameter(parameterName="countryName")
String countryName,
@WebServiceParameter(parameterName="majorName") String majorName){

StudyAbroad studyAbroad = new StudyAbroad();
studyAbroad.setReturnCode(101);
studyAbroad.setMessage("I couldnt find university for;
countryName("+countryName+")"
+"; majorName("+majorName+")");

System.out.println("I am findUniversity");


return studyAbroad;
}
}
```

5.1.6 İstemci (Müşteri)

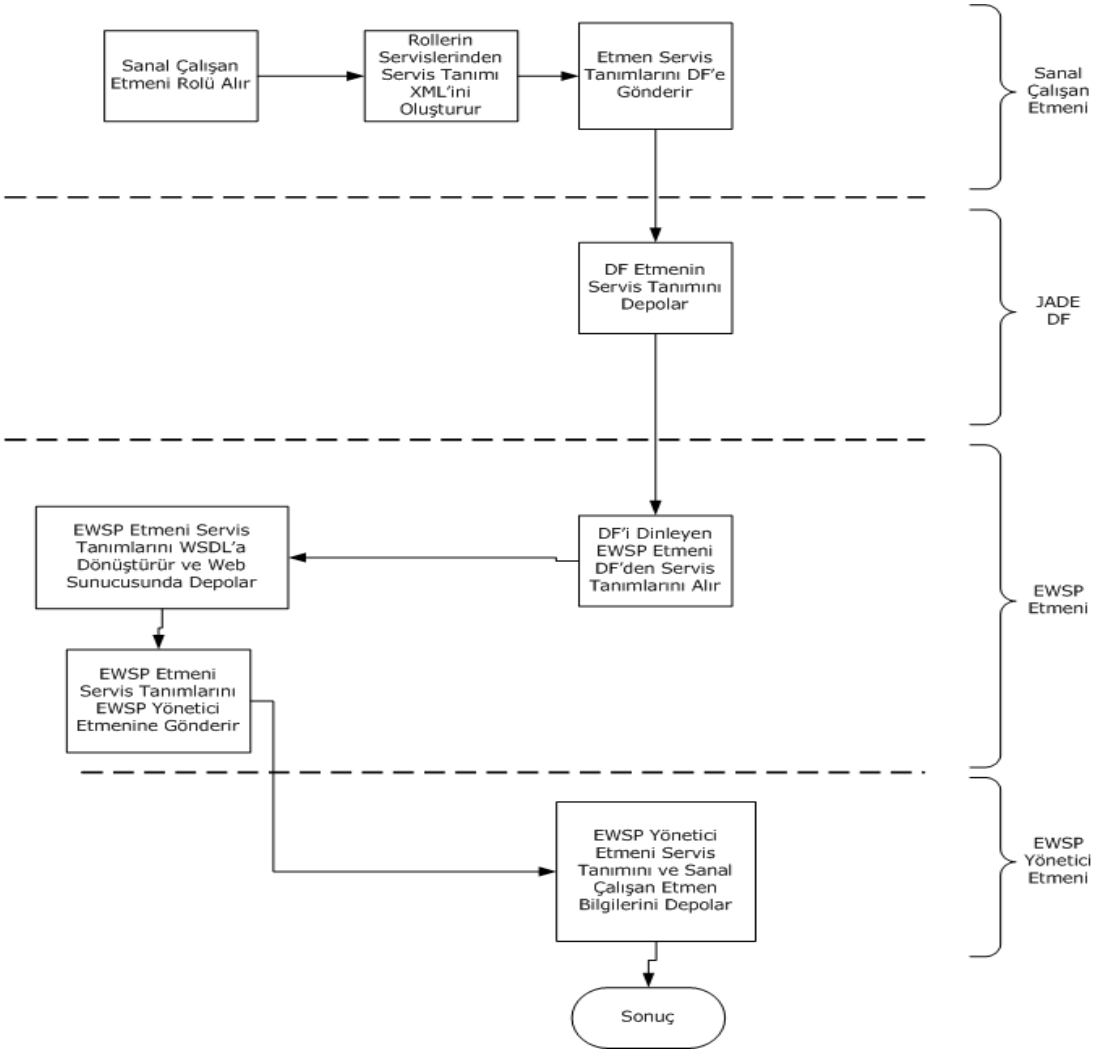
İstemci web servisi talebinde bulunan ve onu kullanan dışsal birimdir. İstemci web servisi çağırabilme yeteneğine sahip herhangi bir uygulama olabilir. İstemci uygun web servis sağlayıcı etmeni bulabilmek için servis adı, servis tanımı gibi gerekli kriterleri sağlayarak EWSP Yönetici etmeni tarafından sağlanan **findservice** web servisini çağırır. İstemci uygun servisi bulabilmek için en doğru ve kendisine en uygun kriterleri sağlamak durumdadır. İstenen koşullara uygun bir etmen servisi bulunursa, servisi sağlayan etmeni belirleyen **awardId** bilgisi istemciye döner. İstemci servis çağrılarını **awardId** değerini kullanarak yapar. İstemcinin talebini tek servis karşılamıyorsa EWSP sistemi birden fazla servisi bir araya getirerek bütünleşik servis oluştur. İstemcinin yaptığı servis çağrıları sistem tarafından kayıt edilmektedir.

5.2 Servis Kaydı

Sanal çalışan etmeni rol başvurusu kabul edildikten sonra rolü alır. Rolün tanımını gösteren XML tanımını edinen etmen, aldığı rolün XML tanımı üzerinden rolleri alır ve servisleri oluşturur.

RoleServiceManager sınıfı annotation yöntemini ve diğer programlama yöntemlerini kullanarak sınıf tanımlarından XML servis tanımlarını oluşturur. Sanal çalışan etmeni bu tanımları DF'ye gönderir. DF'yi dinleyen EWSP etmeni servis tanımları XML'ni alır ve onu WSDL'e dönüştürür, daha sonra bu bilgileri web sunucusunda saklar. WSDL yaratma işlemi tamamlandıktan sonra EWSP Etmeni

bütün tanımları EWSP Yönetici Etmenine iletir. EWSP Yönetici etmeni servisleri kayıt eder ve servisler ve onu sağlayan etmen hakkındaki bilgileri depolar. Servis kayıt akışı Şekil 5.5'de gösterilmiştir.



Şekil 5.5 : EWSP servis yaratma.

5.3 Role Servis XSD (EWSPX)

Şekil 5.6'da gösterildiği gibi rol servis tanımlarında kullanılmak için EWSPX isimli rol xsd tanımı yapılmıştır. Bu **xsd** kullanılarak sistem açılış sırasında kullanılacak rol xml tanımlarını kolay bir şekilde yapılabilecektir.

XSD tanımının en üst seviyesinde **roles** yer almaktadır. **Roles** altında bir veya daha fazla sayıda tekrar eden **role** bulunmaktadır ve bu durum sistem içinde birden fazla rolün yer aldığını ifade etmektedir. **Role** altında rolün adını belirten **roleName**, rolü gerçekleyen ve çalışma zamanında yüklenecek olan sınıfı belirten **roleClass**, rol

yöneticisi tarafından yükleme anında bu rolün alınıp alınmayacağını gösteren **roleManagerDefault** yer almaktadır. Ayrıca rol için gerekli olan koşulları belirten **roleRequirements** ve rolün sahip olduğu servisleri belirten **roleServices** bulunmaktadır.

roleRequirements bir veya daha fazla **roleRequirement** elemanından oluşmaktadır. **roleRequirement** koşulun adını ifade eden **roleRequirementName**, koşulun tanımını belirten **roleRequirementDescription** ve koşulun zorunluluğunu gösteren **roleRequirementRequired** elemanlarından oluşmaktadır.

roleServices bir veya daha fazla **service** elemanından oluşmaktadır. Bu bilgiler rolün sahip olduğu servisleri ifade etmektedir. **service** servisin adını belirten **serviceName** ve servisi gerçekleyen ve çalışma zamanında yüklenecek sınıfı belirten **serviceClass** elemanlarından oluşmaktadır.

EWSPX elemanları detayları şu şekildedir;

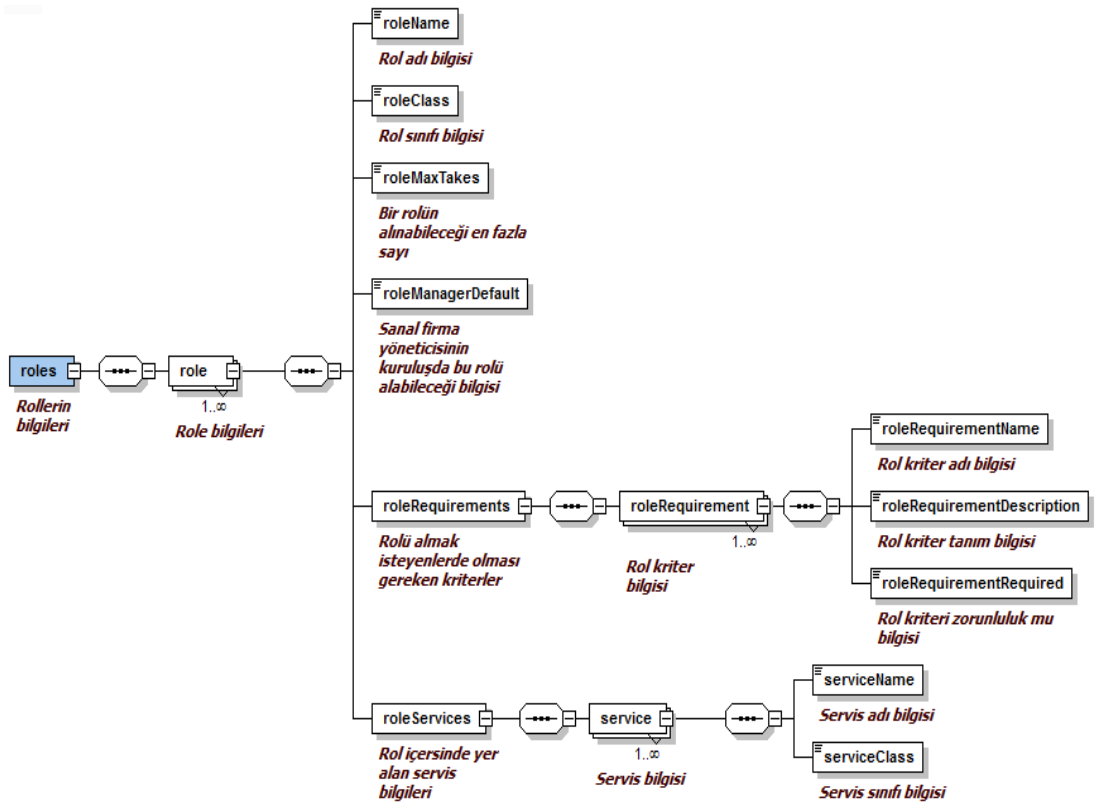
1)**roleName***:* Rol adını ifade eder.

2)**roleClass:** Rolün gerçeklendiği sınıfı ifade eder. Örneğin **com.ewsp.example.role.SatısTemsilcisi**.

3)**roleMaxTakes:** Rolü alabilecek maksimum sanal çalışan sayını ifade eder. Bir sanal firma içersinde bu rolü alabilecek maksimum sayıdaki sanal çalışanı gösterir. Örneğin bir firmada en fazla bir adet genel müdür bulunur, ancak on adet satış temsilcisi bulunabilir.

4)**roleManagerDefault***:* Sanal firma yöneticisinin varsayılan rolü olup olmadığını belirtir. Sistem yöneticinin sistem başlaması sırasında edinmesi gereken rolleri belirtir. Sistem yüklemesi sırasında bu değeri bir olarak belirlenmiş roller sistem yöneticisi tarafından edinilir.

5)**roleRequirements:** Rolün bir sanal çalışan tarafından edinilmesi için istenen kriterleri ifade eder. Bu kriterler sanal çalışan başvurusu sırasında başvuru ekranında çıkar. Sanal çalışan başvuru sırasında bu kriterlerin değerlerini sağlar. Sanal çalışan rol başvuru formu bu bilgilere göre oluşturulur. Bu eleman xml içinde birden fazla olabilir.



Şekil 5.6 : Rol tanım XSD.

a)**roleRequirementName:** Role kriterinin adını ifade eder. Örneğin iş tecrübesi süresi gibi.

b)**roleRequirementDescription:** Role kriterinin tanımını, uzun anlatımını ifade eder.

c)**roleRequirementRequired:** Bu kriterin başvuru formunda doldurulmasının ve başvurunun zorunluluğunu ifade eder.

6)**roleServices:** Role içersinde yer alan servisleri ifade eder. Rolün içersinde yer alan servisler tanımlanır. Bu eleman xml içinde birden fazla olabilir.

a)**serviceName:** Servisin adını ifade eder.

b)**serviceClass:** Servisin gerçeklendiği sınıfı ifade eder. Örneğin **com.EWSP.example.service.DomesticHotelReservation.**

Aşağıda şemaya uygun bir xml dosyası örnek olarak gösterilmiştir.

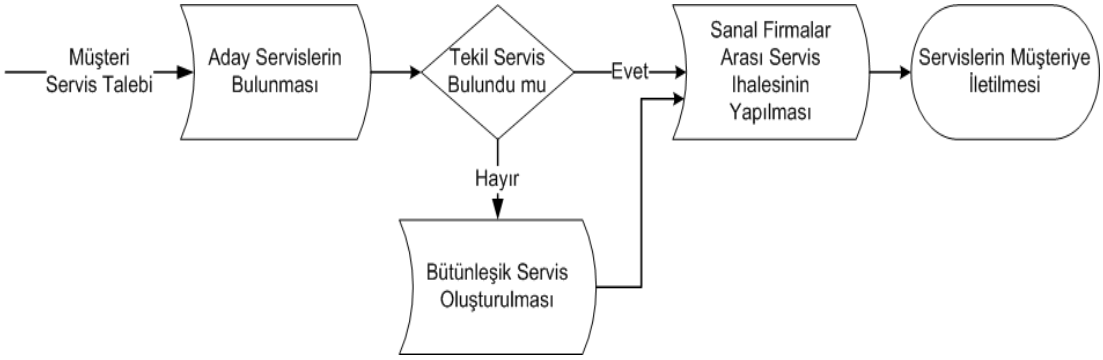
```
<role>
<roleName>Silver Satış Temsilcisi</roleName>
<roleClass>com.EWSP.example.role.SilverSaleRepresentative</roleClass>
<roleMaxTakes>2</roleMaxTakes>
<roleYöneticiDefault>1</roleYöneticiDefault>
<roleRequirements>
```

```
<roleRequirement>
<roleRequirementName>
        Üniversite Mezuniyeti
          </roleRequirementName>
<roleRequirementDescription>
        Üniversite Mezuniyeti işi için gerekli
         </roleRequirementDescription>
<roleRequirementRequired>
1
       </roleRequirementRequired>
</roleRequirement>
<roleRequirement>
<roleRequirementName>
        Tecrübe
          </roleRequirementName>
<roleRequirementDescription>
        İş için tecrübe gerekli-4 yıl
         </roleRequirementDescription>
<roleRequirementRequired>
1
       </roleRequirementRequired>
</roleRequirement>
<roleRequirement>
<roleRequirementName>
        Seyahat Engeli
          </roleRequirementName>
<roleRequirementDescription>
        İş için seyahat engeli olmamalı
         </roleRequirementDescription>
<roleRequirementRequired>
1
       </roleRequirementRequired>
</roleRequirement>
</roleRequirements>
<roleServices>
<service>
<serviceName>Ülke içi Otel Reservasyon Servisi</serviceName>
<serviceClass>com.EWSP.example.service.DomesticHotelReservation</serviceClass>
</service>
<service>
<serviceName>Ülke içi Tur Reservasyon Servisi</serviceName>
<serviceClass>com.EWSP.example.service.DomesticTourReservation</serviceClass>
</service>
<service>
<serviceName>Ülke içi Gezi Planlama Servisi</serviceName>
<serviceClass>com.EWSP.example.service.DomesticTripPlanning</serviceClass>
</service>
</roleServices>
</role>
```


5.4 EWSP'de İhale Yönetimi

Müşterinin servis talebi karşısında EWSP yönetici etmeni öncelikle uygun servisi daha önceki bölümlerde anlatıldığı şekilde bulmaya çalışır. Birden fazla aday servis bulunması halinde bir ihale işlemi ile seçim gerçekleştirilir. Müşterinin talebi öncelikle tekil servisler arasından karşılanmaya çalışılır. Bir tekil servis

bulunamaması durumunda bütünleşik servis oluşturulmaya çalışılır. Tekil adayın bulunması veya bütünleşik servis adayı oluşturulmasından sonra EWSP yönetici etmeni ihale sürecini başlatır ve gerçekleştirir; sonucu EWSP Web üzerinden talepte bulunan müşteriye iletir. Servis bulma ve ihale başlatma süreci Şekil 5.7'de gösterilmiştir. EWSP'de yapılan ihaleler modüler bir ihale yöneticisi tarafından yürütülmektedir. Bu bölümde ihale süreci detayları aktarılacaktır.



Şekil 5.7 : İhale süreci.

5.4.1 Servis istek kriterleri

Müşteri istediği servisleri elde edebilmek için aşağıda belirtilen bilgileri servis bulma servisine sağlamak durumundadır. Sağlanan bilgiler kullanılarak servis bulma ve ihale işlemi gerçekleşmektedir. Sağlanması gereken bilgiler:

serviceName: İstemcinin talepte bulunduğu servisin adı. Bu bilgi sistemdeki aday servisleri bulmak için kullanılmaktadır. Servis adına uyan servis sistemde olmayabilir bu durumda servis tanımı ve anahtar kelimelere bakılarak aday servisler bulunur.

serviceDescription: İstemcinin talepte bulunduğu servisin tanımı. Servis ifade eden servis açıklama/tanım bilgisidir. EWSP Yönetici etmen bu bilgileri servis deposundaki bilgiler ile eşleştirerek aday servisleri bulmak için kullanır. Örneğin **otel rezervasyonu yapan servis** gibi

keywords: Uygun web servisini bulmak için kullanılan anahtar kelimeler. EWSP Yönetici etmen bu bilgileri servis deposundaki bilgiler ile eşleştirerek aday servisleri bulmak için kullanır. Örneğin **otel rezervasyon, otel, rezervasyon** gibi.

commisionRate: Servis çağrısı için istemcinin ödemeyi kabul edebileceği en yüksek komisyon oranı. Maksimum değer (**commisionRate**$_{max}$) belirtilmelidir.

fixedCommisionValue: Servis çağrısı için istemcinin ödemeyi Kabul ettiği sabit komisyon değeridir. **commisionRate** değeri kullanılarak hesaplanan komisyon değeri **fixedCommisionValue** değerini aşarsa, istemci **fixedCommisionValue** değerini öder. **fixedCommisionValue** maksimum değeri sağlanmalıdır.

Yukarıdaki bilgilere ek olarak müşteri talep ettiği servise ait servis seviyesi bilgilerini de sağlamak durumundadır. Bulunan servisin talep edilen özellikleri (servis seviyesi) karşılıyor olması gerekmektedir. Aşağıdaki parametreler servis seviyesini belirtir:

invocationCount: Servisin bir ayda kaç kez çağrılacağını gösteren değerdir. $\mathbf{invocationCount_{max}}$ ve $\mathbf{invocationCount_{min}}$ değerlerinin belirlediği aralıkta veya eşit olmalıdır.

availabilityRate: Servisin mevcut olma oranını ifade eder. $\mathbf{availabilityRate_{min}}$ değeri sağlanmalıdır.

responseTime: Servisin yanıt süresidir. $\mathbf{responseTime_{max}}$ değeri sağlanmalıdır.

averageResponseTime: Servisin ortalama yanıt süresidir. $\mathbf{averageResponseTime_{max}}$ değeri sağlanmalıdır.

Aşağıdaki parametreler, ihale sürecinde kriterlerin hangi ağırlıkta hesaba katılacağını belirlemek üzere istemci tarafından tanımlanır:

avgResponseTimePerformanceWeight: Tarihsel yanıt süresinin ağırlığı

avgAvailabilityPerformanceWeight: Tarihsel mevcudiyet değerinin ağırlığı

commissionRateWeigh: Komisyon oranının ağırlığı

fixedCommissionValueWeight: Sabit komisyon değerinin ağırlığı

invocationCountWeight: Çağırma adedinin ağırlığı

availabilityRateWeight: Bulunma değerinin ağırlığı

responseTimeWeight: Yanıt süresinin ağırlığı

averageResponseTimeWeight: Ortalama yanıt değerinin ağırlığı

5.4.2 İhale süreci

EWSP Yönetici etmeni yukarda belirtilen parametreleri aldıktan sonra ihale sürecini başlatır. EWSP'de yapılan ihaleler modüler bir ihale yöneticisi tarafından yürütülmektedir. İhale işlemi şu adımlardan oluşur:

1. İstemcinin beyan ettiği servis kriterlerine uygun olan aday servislerin bulunur.
    a. Öncelikle Tekil servis adayların bulunur
    b. Tekil aday servisleri bulunamıyorsa, bütünleşik servis adaylarının belirlenir
2. Adayların performans tarihçesi elde edilir ve performans değerleri oluşturulur.
3. Adaylardan teklif isteyerek ihale süreci başlatılır.
4. İhale katılımcılarından teklifler toplanır.
5. Teklifler, katılımcıların tarihi(önceki) performanslarına ve mevcut tekliflerine bakılarak değerlendirilir. En iyi n adet (n:bir sistem parametresi) teklif seçilir. Eğer uygun teklif bulunamaz ise, ihale süreci başarısızlıkla sonlandırılır.
6. Kazanan etmene ihale sonucunu belli eden ve üretilen tekil **awardId** ile birlikte kazanma mesajı gönderilir. **awardId** istemci tarafından servis çağrısında kullanılır. **awardId** değeri etmenin kazandığı ihaleyi belirtir.
7. İhale işini kazanan bütünleşik servis ise bütünleşik servisi sunmak üzere EWSP Bütünleşik Servis Yöneticisi Etmenini yaratılır.
8. Web servisi istemcisine ilgili bilgiler ile birlikte mesaj gönderilir. Bu bilgiler arasında servis detay bilgileri ve **awardId** bilgisi yer almaktadır.

5.4.2.1 Servis sağlayıcı adayların belirlenmesi

İhale yöneticisi, istemci tarafından belirlenen servis bulma parametrelerine göre servis sağlayıcı etmenleri belirler. Aday olarak tekil bir servis sağlayıcı etmen bulabileceği gibi birden fazla servis sağlayıcı etmenin servislerini bir araya getirerek bütünleşik servis sağlayıcı adayları bulabilir. Daha önce kazandığı ihale taahhütlerini yerine getirmeyen servis sağlayıcıları yeni ihaleye katılamaz. Kaç adet başarısız

taahhüt yerine getirme sonrasında servis sağlayıcının yeni ihaleye katılmayacağı bir sistem parametresi olarak belirlenir.

Tekil Servis Sağlayıcı Adayların Belirlenmesi

EWSP tekil servis, servis sağlayıcısı etmen adaylarını bulmak için aşağıda belirtilen iki yöntemi kullanır ve servis sağlayıcı adayları içeren etmen kümesi C aşağıdaki şekilde oluşturulur:

1. c_a: **serviceName** parametresi kullanarak servisleri eşleştirme.

2. c_b: **serviceDescription** ve **keywords** parametrelerini kullanarak servis deposu araştırma.

$C = \{c_a, c_b\}$

Bütünleşik Servis Sağlayıcı Adayların Belirlenmesi

EWSP bütünleşik servis sağlayıcısı, etmen adaylarını bulmak için aşağıdaki yöntemi kullanır:

Aranan servis tekil servisler arasında bulunamamış ise servis bütünleşik servis olarak aranır. Bütünleşik servisler arasında yapılan arama anahtar kelimeler üzerinden yapılmaktadır.

AK_{aranan}= Aranan anahtar kelime. Servis isteğinde bulunan müşteri tarafından sağlanır.

AK_{aday1}= Aday servisin sağladığı anahtar kelimeler (S_1 servisinin sağladığı aday kelimeler).

AK_{aday1} sistemde tanımlanmış belirli bir oranda AK_{aranan} içeriyorsa, AK_{aday1} bütünleşik servis adayı olarak belirlenir.

$AK_{aranan\ yeni} = AK_{aranan} - AK_{aday1}$

$AK_{aranan\ yeni}$ diğer adaylar arasında tekrar aranır. Eğer sistemde tanımlanmış oranda uyum sağlayan bir aday servis S_2 bulunursa, S_1 ve S_2 talep edilen servisi sağlayabileceği sonucuna varılır ve ihaleye katılırlar. Böylece aday servis şu şekilde oluşur;

$S_{aday1} = S_1 + S_2$ olarak belirlenir.

$c_1 = S_{aday1}$ servisini sağlayan etmen.

Bu işlem amaca ulaşılana kadar tekrarlanır ve belirlenen adaylar aday listesini oluştururlar.

C= $\{c_1, c_2 \ldots c_n\}$

5.4.2.2 Servis sağlayıcı adayların tarihsel performansının hesaplanması

Eğer servis sağlayıcı adaylar ihaleyi kazanırlarsa, etmenin ihaledeki işi yerine getirmesi EWSP Yönetici etmeni tarafından kayıt altına alınır. Dolayısıyla servis sağlayıcıların tarihsel performans değerleri depolanır ve yeni adayları belirlemek için bu değerler kullanılır. Her bir adayın $avgResponseTimec_i$ ve $avgAvailabiliytRatec_i$ değerleri aşağıda belirtilen formül kullanılarak hesaplanır:

$$\mathbf{avgResponseTimec_i} = \sum \frac{\text{avgResponseTime}_1 + \text{avgResponseTime}_2 + \ldots + \text{avgResponseTime}_n}{\text{n}}$$

$$\mathbf{avgAvailabilityRatec_i} = \sum \frac{avgAvailabilityRate_1 + avgAvailabilityRate_2 + \ldots + avgAvailabilityRate_n}{n}$$

Tarihsel değerlerin toplam sayısı n ile ifade edilmiştir. Bu değerlerin hesaplanmasından sonra istemci tarafından sağlanan parametre değerleri ile tarihsel olarak hesaplanan değerler arasındaki fark oranı hesaplanır. 100'e yakın olan değerlere sahip olan servisler talebe en uygun adaydır.

$$\mathbf{diffAvgResponseTimec_i} = 100 - \left(\frac{avg\,\mathrm{Re}sponseTime_{ci}}{avg\,\mathrm{Re}sponseTime_{\min}} \right) * 100$$

$$\mathbf{diffAvgAvailabilityRatec_i} = \left(\frac{avgAvailabilityRate_{ci}}{avgAvailabilityRate_{\min}} \right) * 100$$

Bütünleşik servisler için değerler bütünleşik servisi oluşturan servislerin değerlerinin toplamı olarak alınmaktadır.

EWSP Yönetici Etmeni "teklif çağrısı" mesajını muhtemel aday etmenlere gönderir. Aday etmenlerden teklifler toplar.

5.4.2.3 Teklif değerlendirme aşaması

Teklifler aşağıdaki kriterlere uygun olarak değerlendirilir:

- Teklif edilen değerler ve parametre değerleri kullanılarak, her bir parametre için aşağıdaki fark oranları hesaplanır:

  $\mathbf{diffCommisionRatec_x}$,

diffFixedCommisionValuec$_x$,

diffInvocationCountc$_x$,

diffAvailabilityRatec$_x$,

diffResponseTimec$_x$,

diffAverageResponseTimec$_x$

- Her teklifin istemcinin sağladığı kriterler, aday etmenlerin verdiği teklif bilgileri ve etmenin tarihsel bilgileri kullanılarak hesaplanan teklif toplam değeri bulunmaktadır. Bu değerin yüksek olması iyi teklifi belirtir. Teklif toplam değeri hesaplanır T_x:

  T_x=**avgResponseTimePerformanceWeight***
  diffAvgResponseTimec$_x$

  + **avgAvailabilityPerformanceWeight * diffAvgAvailabilityRatec$_x$**

  + **commissionRateWeight * diffCommisionRatec$_x$**

  + **fixedCommissionValueWeight * diffFixedCommisionValuec$_x$**

  + **invocationCountWeight * diffInvocationCountc$_x$**

  + **availabilityRateWeight * diffAvailabilityRatec$_x$**

  + **responseTimeWeight * diffResponseTimec$_x$**

  + **averageResponseTimeWeight * diffAverageResponseTimec$_x$.**

- T_x değeri en yüksek olan teklif en iyi tekliftir. T_x değerine göre en iyi k değer seçilir.

Bütünleşik servisler için değerler bütünleşik servisi oluşturan servislerin değerlerinin toplamı olarak alınmaktadır.

Kazanan teklif maksimum ve minimum parametre değerleri arasında olmalıdır.

5.4.2.4 İhalenin sonuçlandırılma aşaması

T_x değerine göre seçilen en iyi servis ihaleyi kazanır. Kazanan servisi sunan etmene servis ihalesini kazandığına dair bilgi ve **awardId** değeri gönderilir. İhale ile ilgili detay bilgiler sisteme kayıt edilir. Servis talebinde bulunan müşteriye EWSP Web etmeni aracılığı ile servis detay bilgileri ve **awardId** bilgisi iletilir. Servis talebi

bulunan müşteri daha sonraki servis çağrılarında **awardId** değerini sağlamak durumundadır.

İhale işini kazanan bir bütünleşik servis ise bütünleşik servisi sunmak üzere EWSP Bütünleşik Servis Yöneticisi Etmeni yaratılır. EWSP Bütünleşik Servis Yöneticisi Etmeni kendisini sisteme kayıt ettirir. EWSP Bütünleşik Servis Yöneticisi Etmeni ,kendisine gelen servis çağrılarını kendini oluşturan alt servislere aktarma yeteneğine sahiptir.

5.4.2.5 İhale sonrası takip

Sistemden servis talebinde bulunarak servis alan müşteriler kendilerine bildirilen **awardId** ile servis çağrılarını yapmak durumundadır. Sistemde kayıtlı olmayan bir **awardId** ile servis çağrısı yapıldığı zaman servis çağrısı doğrudan reddedilir. Sistemde yapılan bütün çağrılar **awardId** üzerinden kayıt edilmektedir. Çağrılar kayıt altına alınarak daha sonraki ihalelere girdi sunmaktadır. Servis sağlayıcıların yerine getirdikleri ve getirmedikleri taahhütler sistem üzerinde kayıt edilmektedir.

6. BAŞARIM DEĞERLENDİRMESİ

Bu bölümde sistem başarım değerleri belirtilecektir. EWSP sistemi üzerinde etmen servisleri web servisi olarak sunulmaktadır. EWSP, istemci sistemlerinden gelen servis talepleri doğrultusunda web servis ihalesi başlatmaktadır. İhale'nin mümkün olduğunca hızlı sonuçlandırılması ve sonucun istemciye iletilmesi gerekmektedir. İhale işlemi, servis bulma web servisi çağrısı sonucunda başlamaktadır. İstemci servis bulma işlemini bir web servisi olarak çağırmaktadır. Servis bulma işleminin anlık gelen taleplere yanıt verme süresi bu çalışma için önemli bir başarım kriteridir. Ayrıca, istemcinin talep ettiği servislerin işlenme süreleri de bu çalışma için sistemin ölçülebilir başarım kriterlerinden biridir.

Servis bulma işleminin başarımını etkileyen faktörler şunlardır:

- Servis bulma işlemine yapılan anlık çağrı sayısı: anlık çağrı sayısının artması sistem üzerinde yük yaratacaktır. Yük artışı EWSP Web bileşeni, EWSP yönetici etmen üzerinde ve ihaleye katılan etmenler üzerinde olacaktır.
- Servis bulma ihalesine katılan sanal firma sayısı: aday etmen sayısı arttıkça mesaj trafiği artacaktır ve sistemi yavaşlatabilir.
- İhale yürüten EWSP Yönetici Etmen sayısı: çok sayıda gelen anlık çağrılar tek bir EWSP Yöneticisi etmeni üzerinde yük oluşturabilir.

Tez çalışması için yapılan başarım değerlendirmesi çalışmalarında, istemcilerin servis bulma taleplerinin karşılanma süresi ve servislerin yerine getirilme süreleri değerlendirme kriterleri olarak kabul edilmişlerdir. Yukarıda yer alan faktörlerin bu kriterleri ne şekilde etkilediklerini belirlemek üzere Windows ve Unix işletim sistemlerinin yer aldığı ortamlarda testler yapılmıştır. Test değerleri değişen sayıda kullanıcının aynı anda ihale bulma servisini ve diğer servisleri çağırması ile oluşturulmuştur. Çağrı süreleri için aşağıdaki değerler ölçülmüştür;

- minimum işlem süresi: Çağrılar içindeki taleplerden en kısa sürede tamamlanan talep çağrısının süresidir.

- maksimum işlem süresi: Çağrılar içindeki taleplerden en uzun sürede tamamlanan talep çağrısının süresidir.
- ortalama işlem süresi: : Çağrılar içindeki taleplerin ortalama tamamlanma süresidir. Bütün taleplerin tamamlanma süresinin toplam çağrı adedine bölünmesiyle elde edilir.

Ölçümler sırasında ortaya çıkan değerler (ortalama, minimum ve maksimum), yapılan beşer deneme sonrasında elde edilen sonuçların ortalaması alınarak bulunmuştur.

Test koşulları ve elde edilen sonuçlar bu bölümde ayrıntılı olarak anlatılacaktır.

6.1 Windows Test Ortamı

Windows işletim sisteminin yer aldığı test ortamında EWSP uygulamasının yüklü bulunduğu bir bilgisayar, EWSP sanal firmaların (servis sunmaya aday etmenler) çalıştığı bir bilgisayar ve servis taleplerinin ve servis çağrılarının yapıldığı bir bilgisayar olmak üzere üç adet bilgisayar bulunmaktadır. Üç bilgisayar birbirlerine kablosuz ağ ile bağlanmıştır. Servis çağrıları ve taleplerinin sayısı *thread* kullanılarak artırılmaktadır.

EWSP Bilgisayar:

EWSP Web Sunucusu, EWSP Yöneticisi, EWSP Etmen'lerinin çalıştığı bilgisayardır ve şu özelliklere sahiptir:

- İşlemci: Intel Core 2 Duo P860 2.4 GHz
- Bellek: 3GB
- İşletim Sistemi: Windows Vista Service Pack 1

Sanal Firmaların Çalıştığı Bilgisayar:

Sanal firmaların, sanal çalışanların ve sanal aday çalışanların yer aldığı bilgisayardır ve şu özelliklere sahiptir:

- İşlemci: Intel Core i5 CPU M430 2.27 GHz
- Bellek: 4GB
- İşletim Sistemi: Windows 7

İstemcinin Çalıştığı Bilgisayar:

Servisler için talepte bulunan ve servis çağrılarının yapıldığı istemcilerin çalıştığı bilgisayardır ve şu özelliklere sahiptir:

- İşlemci: Genuine Intel (R) CPU T2400 1.83 GHz
- Bellek: 2GB
- İşletim Sistemi: Windows XP

Ağ Bilgileri

- Kablosuz LAN, IEEE 802.11 54Mbit/s

6.1.1 Test senaryo 1

İlk test senaryosunun amacı aynı anda yapılan çağrı sayısının servis bulma süresi üzerindeki etkisini belirlemektir. Belirli bir kriter kümesine sahip olan bir servisin bulunması için istemcilerden gelen anlık çağrıların sayısı değiştirilerek, sistemin talebi karşılama-aranan servisin bulunması- süresi ölçülmüştür. Deney ortamında, aranan servis kriterlerine uyan bir adet servis bulunmaktadır. Bu testin amacı bir adet servis sağlayıcısı ile servis bulma süresi ölçmek ve en az tek katılımcı ile ihale sürecini değerlendirmektir . İhale yürütücüsü olarak da bir adet EWSP Yönetici etmeni kullanılmıştır.

Dışarıdan gelen her bir servis talebi için yürütülen işlem adımları şunlardır:

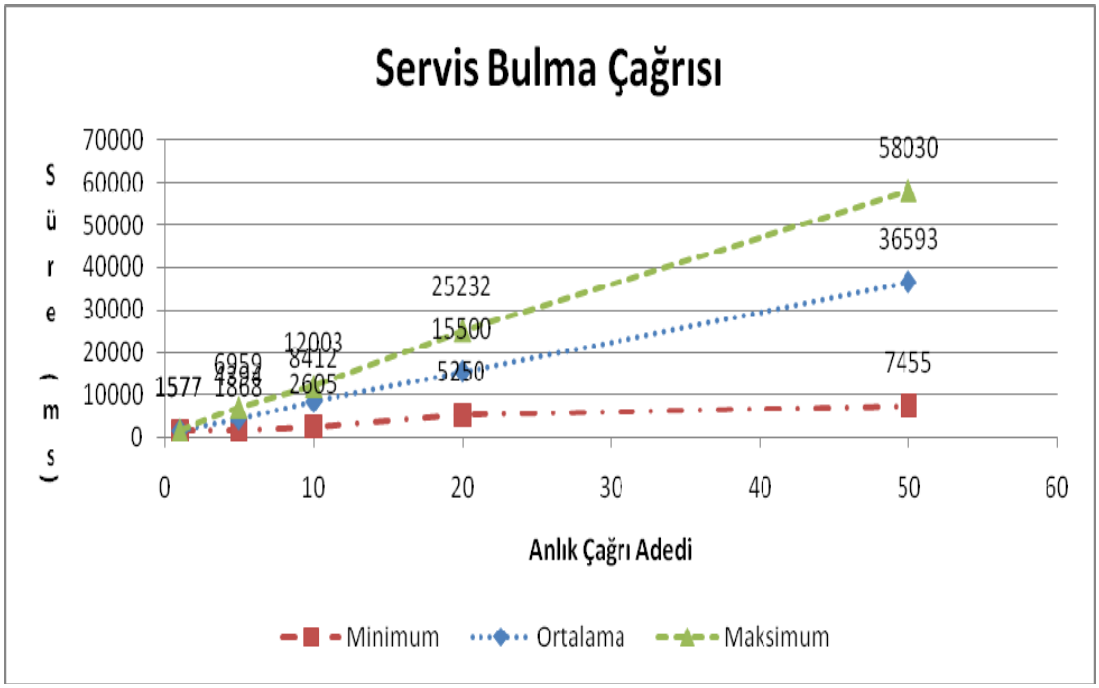
1. EWSP Web gelen talebi SOAP mesajından etmenler arası kullanılan XML mesajına çevirerek EWSP Yönetici etmenine iletir.
2. EWSP Yönetici Etmeni aday servisleri bularak servis ihalesi başlatır.
3. Her bir servis ihalesi için EWSP etmeni üzerinde bir ihale davranışı çalışır ve ihale işlemini sonuçlandırır.

İhale süreci 3 kritik nokta üzerinden geçmektedir; EWSP Web bileşeni, EWSP Yönetici Etmeni ve aday (ihaleye katılan) etmen. EWSP Web bileşeni bir http sunucusu içinde çalışan bir **servlet** olduğu için çoklu istekleri ön tanımlı değere kadar kabul etmekte ve kendisine gelen talepleri kuyruğa sokmaktadır. EWSP Yönetici etmeni ve aday etmen kendilerine gelen talepleri sıralı ancak iş kuyruğu üzerinden işlemektedir. Çağrılar istemci bilgisayar üzerinden iş iplikçileri

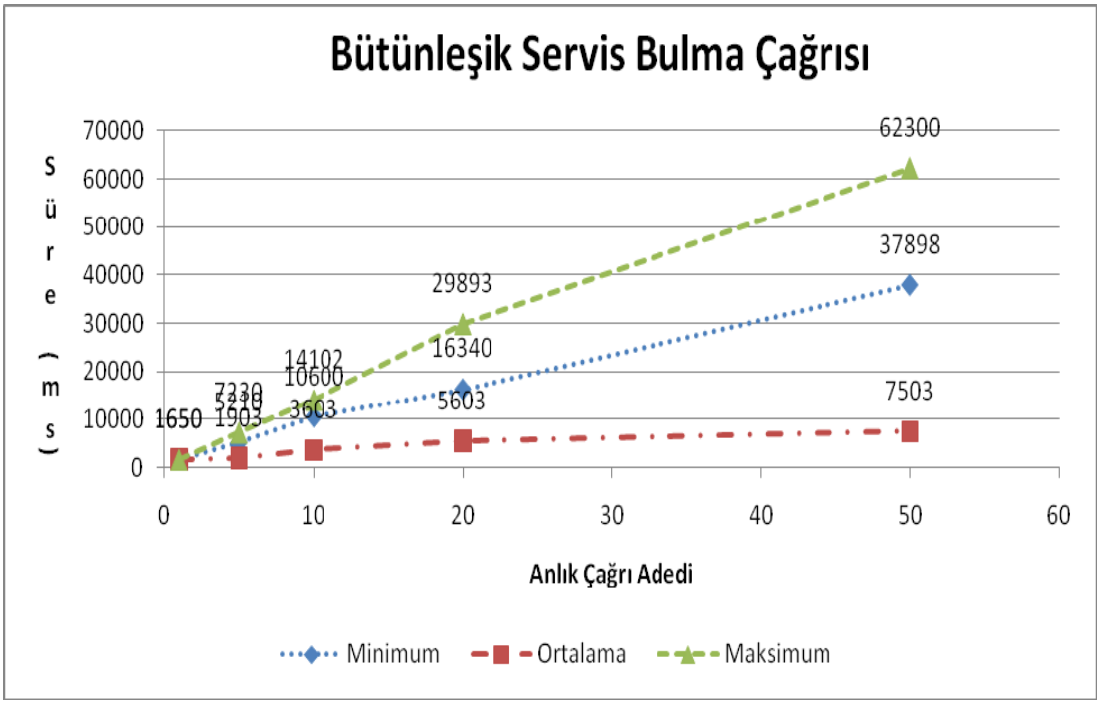
kullanılarak üretilmektedir. Geçen süre istemciden talebin çıkması ile sonucun alınmasına kadar geçen zaman baz alınarak ölçülmüştür. Her çağrı sayısı için beş adet ölçüm yapılmış ve değerler hesaplanırken bunların ortalamaları dikkate alınmıştır. Her bir çağrı adedi için minimum, maksimum ve ortalama süre değerleri ölçülmüştür.

Şekil 6.1'de görüldüğü gibi, anlık çağrı sayısı arttıkça ihale sürecini sonlandırma süresi de artmaktadır. EWSP yönetici etmeni ve aday etmeni sayıca aynı kaldığından gelen çağrıları karşılama süreleri artmıştır. Maksimum işlem süresi hızlı bir şekilde artarken minimum süre çok fazla artmamıştır. Zaman geçtikçe gelen çağrıların birikmesi sonucu sistemin yavaşladığını göstermektedir.

Aynı şekilde bütünleşik servis adayının bulunduğu servis arama çağrıları grafiği Şekil 6.2'de görülmektedir. İki durum arasında önemli bir fark oluşmadığı görülmektedir. Bu nedenle bütünleşik servis arama ile tekil servis arama arasında kayda değer bir farklılık olmadığı ifade edilebilir. Aradaki ufak farklar birden fazla aday etmenin olmasıdır. Bu yorumu Şekil 6.3'de adayla yapılan tekil servis bulma ölçümünden de görebiliriz. Oradaki değerler iki adaylı tekil servis bulma ölçüm değerleri bütünleşik servis bulma ölçüm değerlerine yakın sürelerdir.



Şekil 6.1 : Tek aday ile servis bulma.



Şekil 6.2 : Tek aday ile bütünleşik servis bulma.

6.1.1.1 Servis bulma adımlarının değerlendirilmesi

Servis bulma işleminde zaman alan temel adımları şu şekilde sıralayabiliriz. Servis arama mesajları EWSP web bileşenine (http sunucu) SOAP mesajı şeklinde gelmektedir. Bu mesajın EWSP yönetici etmen ile haberleşmede kullanılan xml formatına dönüştürülmesi gerekmektedir. Bu adıma gelen SOAP mesajından EWSP yönetici etmen XML'e dönüşüm diyebiliriz. Dönüştürülen mesaj EWSP yönetici etmene gönderildikten sonra EWSP yönetici etmen Bölüm 5.4'te anlatılan şekliyle aday etmenlerin bulunması adımını işletmektedir. Daha sonra EWSP yönetici etmen, aday etmenleri kullanarak ihale işlemi gerçekleştirmektedir. İhale işlemi sonrasında EWSP yönetici etmen sonucu EWSP web bileşenine bildirmektedir. EWSP web bileşeni EWSP yönetici etmenin gönderdiği XML mesajının SOAP mesajına çevirir.

Çizelge 6.1 : İhale süreci temel adım süreleri.

Adım	Açıklama	Süre(ms)
EWSP Web bileşeninde (Http sunucuda) geçen süre	İstemciden gelen SOAP mesajının EWSP yönetici etmen XML mesajına dönüştürülmesi	74
	EWSP yönetici Etmenden gelen XML mesajının SOAP mesajına dönüştürülmesi	62
	Diğer (EWSP Yönetici etmene mesaj gönderme/alma ve sunucu içindeki diğer işlemler)	89
EWSP Yönetici etmende geçen süre	Aday bulma	574
	İhale Süresi	395
	Diğer (EWSP Yönetici etmen içinde geçen süreler)	130
İstemci işlemleri-istemci http sunucu haberleşme Süreleri	İstemcide yapılan işlemler, istemci sunucu haberleşme süresi.	97
Toplam Süre		1421

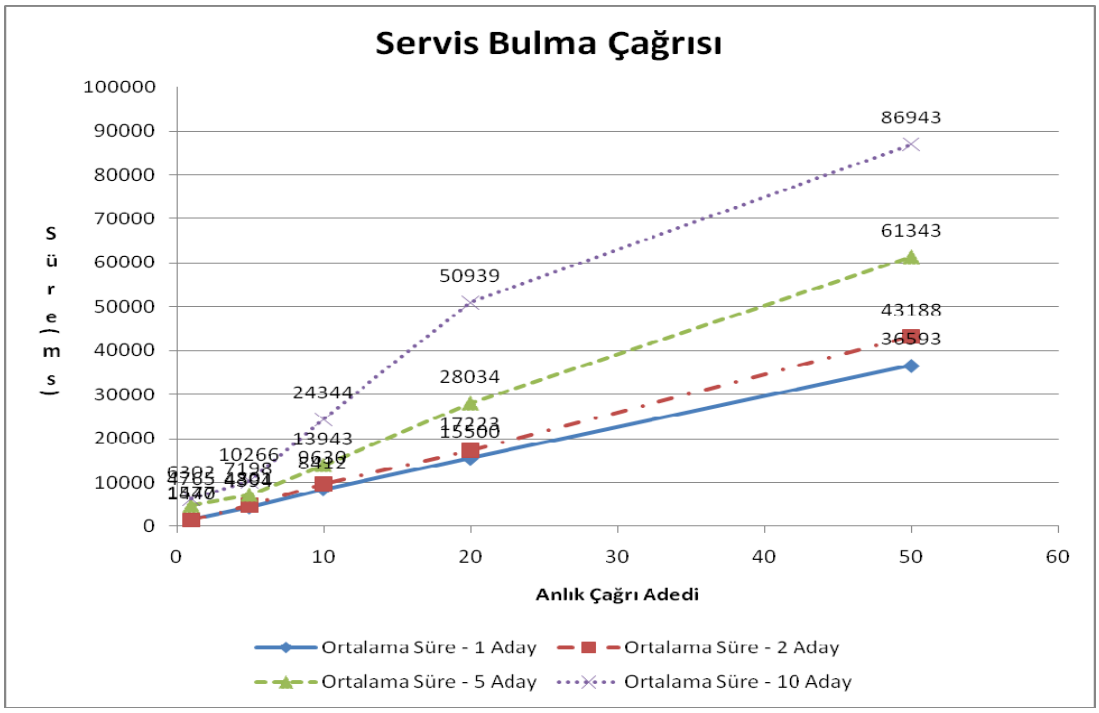
Çizelge 6.1'de bu adımların bir çağrı için süreleri gösterilmiştir. Süreler tek bir aday var olduğu bir ihale çağrısı için ölçülmüştür. Ölçüm sırasında sadece tek bir çağrı yapılmıştır. En uzun süre alan adım aday etmenlerin bulunması adımı olarak görülmüştür. Bu adımda yoğun olarak gerçekleşen veritabanı işlemleri süreyi uzatmaktadır. Daha sonra en fazla zaman alan aşama ise ihale sürecinin kendisidir.

6.1.2 Test senaryo 2

İkinci test senaryosunda, yine bir öncekinde olduğu gibi, aynı anda yapılan çağrı sayısı arttırılarak servis bulma süreleri ölçülmüştür. Ancak ilk senaryodan farklı

olarak, aranan servis kriterlerine uyan servis sayısı da artırılmıştır. İhale yürütücüsü olarak yine bir adet EWSP Yönetici etmeni kullanılmıştır.

Her bir servis talebi için yürütülen işlem adımları ilk test için sıralananlar ile aynıdır, tek farklılık ihaleye katılacak olan aday servis sayısının birden fazla olmasıdır.

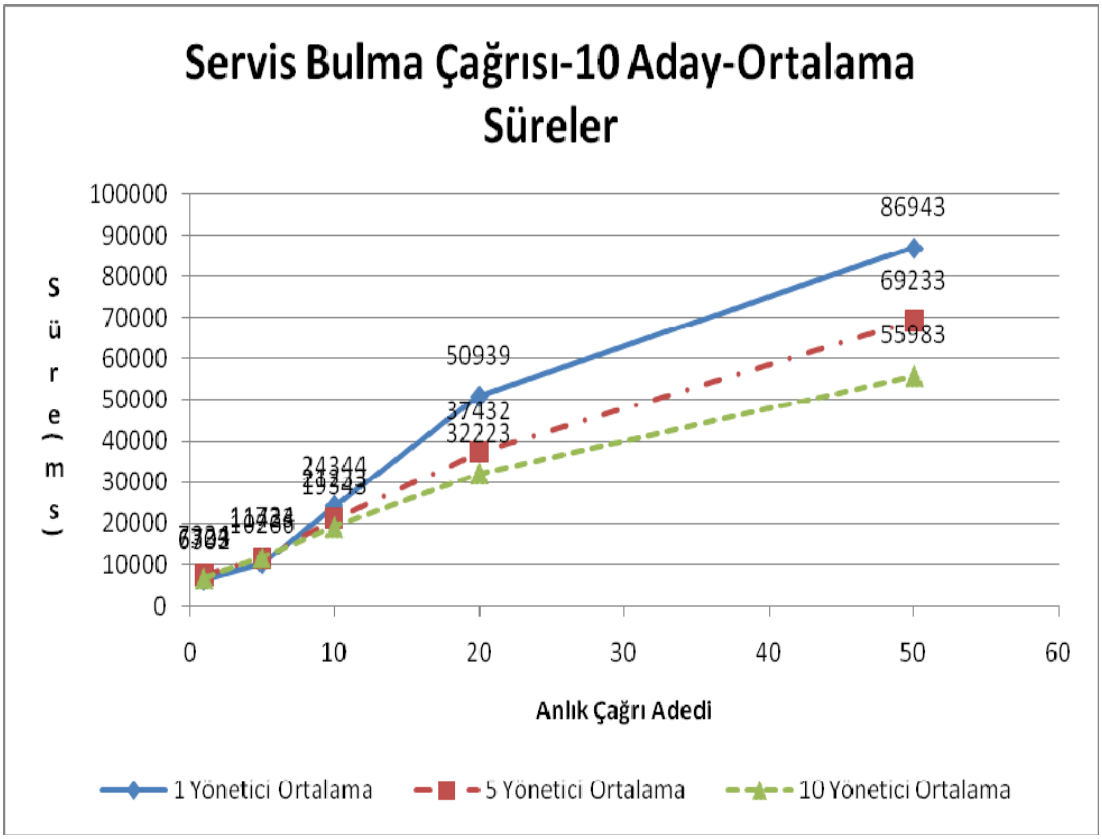


Şekil 6.3 : Birden fazla aday ile servis bulma.

Şekil 6.3'de görüldüğü gibi anlık çağrı sayısı arttıkça ihale sürecini sonlandırma süresi de artmaktadır. EWSP yönetici etmeni aynı kaldığından dolayı gelen çağrıları karşılama süresi artmıştır. Aday sayısı arttıkça adaylar ile EWSP yönetici etmen arasında var olan mesajlaşma trafiği artmıştır. Bu ise ihale süresini arttıran bir etken olmuştur.

6.1.3 Test senaryo 3

Üçüncü test senaryosunda ihale yürütücüsü olarak kullanılan EWSP Yönetici Etmeni sayısındaki değişikliklerin servis bulma süreleri üzerindeki etkisi ölçülmüştür. Ölçümler, aynı anda yapılan farklı sayıdaki servis bulma çağrıları için tekrarlanmıştır. Aranan servis kriterlerine uyan servis sayısı sabit bir değerde, on servis, tutulmuştur. İhale yürütücüsü olarak kullanılan EWSP Yönetici Etmeni sayısı arttırılarak sonuçları gözlenmiştir.



Şekil 6.4 : Birden fazla EWSP yöneticisi ile servis bulma ortalama süre.

Şekil 6.4'te görüldüğü gibi, anlık çağrı sayısı arttıkça ihale sürecini sonlandırma süresi de artmaktadır. Ortalama süreler yönetici etmen sayısının artması ile birlikte düşüş göstermiştir. Anlık çağrı sayısı 10'a kadar olan değerlerde küçük farklar oluşmaktadır. Anlık çağrı sayısının 10'a geçmesi ile birlikte oluşan ortalama yanıt süresi düşmeye başlamıştır. Anlık çağrı sayısı 50'ye ulaştığında fark daha belirgin hale gelmiştir. Çağrı sayısı elli iken eğer sistemde bir adet yönetici etmen varsa bütün çağrıları (ihaleleri) bir adet yönetici etmen yürütmekte bu durum yönetici etmenin yanıt süresini yükseltmektedir. Oysa sistemde on adet yönetici etmen olduğu durumda her bir yönetici etmen için beş adet çağrı düşmekte ve bu durumda yönetici etmenler çağrıları (ihaleleri) daha hızlı sonuçlandırabilmektedir. On adet yönetici etmen ile çağrıların ölçülenden daha hızlı sonuçlanması beklenir., Ancak, web sunucusunun tek olması ve tüm ihalelere aynı etmenlerin katılması nedeniyle, servis bulma süresinde azalma yönetici etmen sayısındaki artışla aynı oranda olmamıştır. Bu duruma rağmen, yeni yönetici etmenlerin sisteme katılması yüksek çağrı sayılarında başarımı arttırdığı gözlenmiştir.

6.1.4 Test senaryo 4

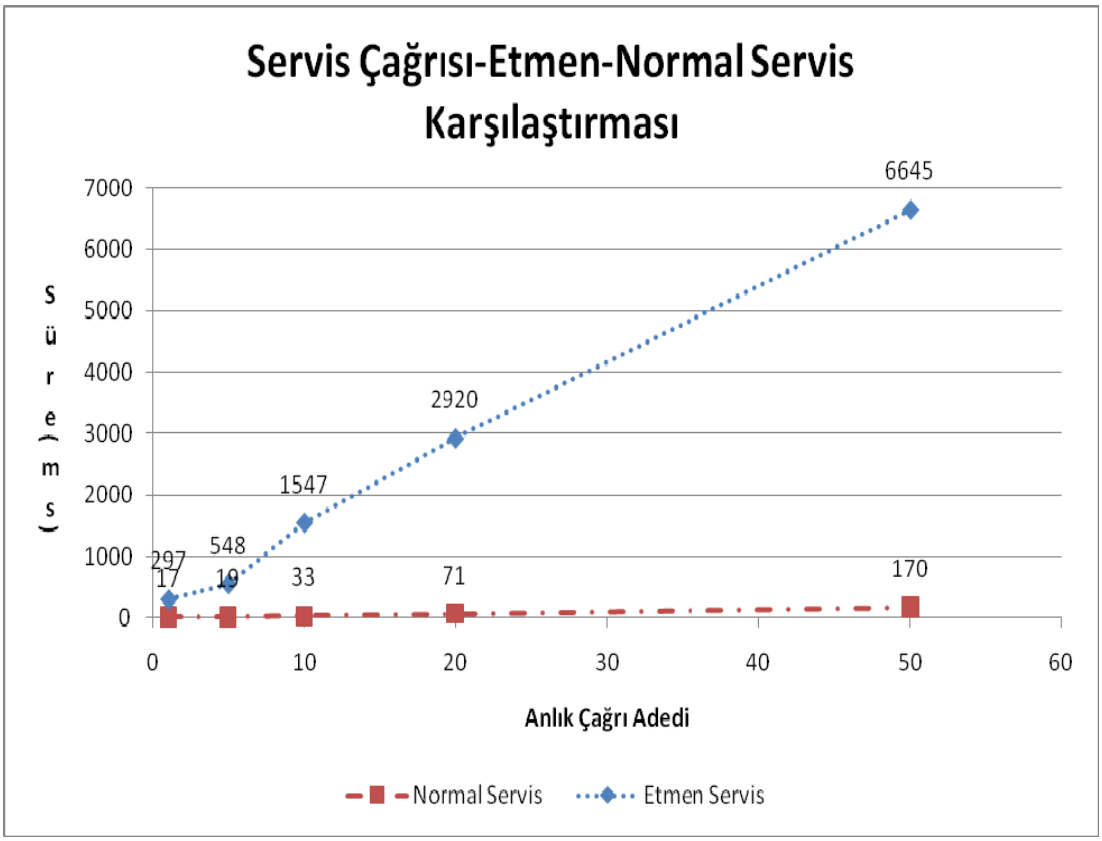
Dördüncü test senaryosunda ihale sonrası belirlenen bir servis için yürütülen çağrının tamamlanma süresi ölçülmüştür. Daha sonra klasik web servisi tanımlama yöntemi ile tanımlanmış bir web servisinin çağrı tamamlama süreleri ölçülmüştür. Etmen servisi ile web servisi çağrı tamamlanma sürelerini karşılaştırmak amaçlanmıştır. Etmen web servisi ile normal web servisi çağrıları arasında oluşan zaman farkları belirlenmeye çalışılmıştır. Bu senaryoda kullanılan servisler hiçbir işlem yapmamakta sadece basit bir mesaj dönmektedir. Yani servis içinde zaman kaybına yol açacak bir işlem yoktur. Deneyler, etmenlerin servislerine yapılan çağrı sayısı arttırılarak tekrarlanmış ve servis yanıt süreleri ölçülmüştür. Yapılan servis çağrılarında tek bir etmen servisi kullanılmıştır.

Dışarıdan gelen her bir servis çağrısı karşısında, EWSP Web gelen talebi SOAP mesajından etmenler arası kullanılan XML mesajına çevirerek ilgili servis etmenine iletmektedir. Servis etmeni gerekli işlemleri yaptıktan sonra sonucu geriye döner ve çağrı işlemi tamamlanır.

Çizelge 6.2 : Servis çağrısı temel adımlar ve süreler.

Adım	Açıklama	Süre (ms)
EWSP web bileşeni (Http sunucuda) geçen süre	İstemciden gelen SOAP mesajının servis sağlayıcı etmenin mesaj formatı olan XML mesajına dönüştürülmesi	78
	Servis sağlayıcı etmenden gelen XML mesajının istemciye göndermek için SOAP mesajına dönüştürülmesi	65
	Diğer (Servis sağlayıcı etmene mesaj gönderme/alma ve sunucu için diğer işlemler)	86
Servis sağlayan etmende geçen süre		19
Toplam Süre		248

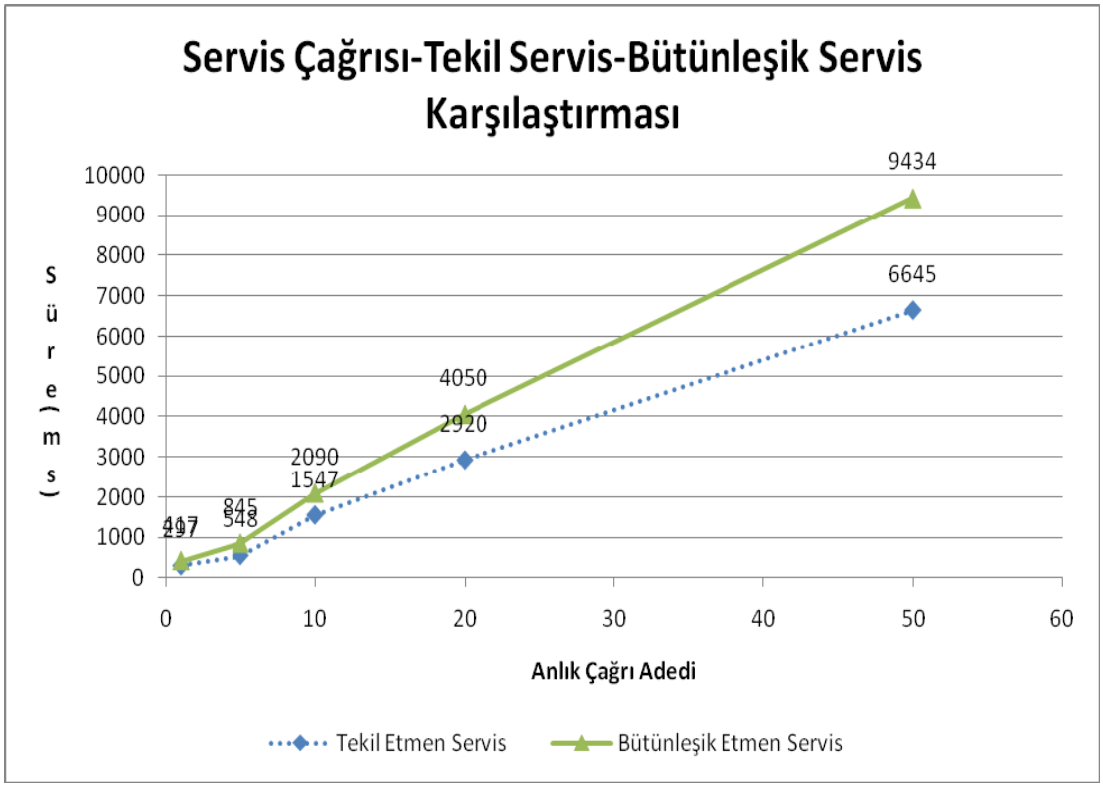
Çizelge 6.2'de dışarıdan gelen bir servis çağrısının sonuçlanana kadar yürütülen temel adımlar ve her adımın aldığı süre gösterilmiştir. Çizelgeden de görüleceği üzere, etmene gelen servis çağrısı, EWSP web bileşeni üzerinde mesaj dönüşümleri ve EWSP web bileşeni ile etmen servis haberleşmelerinde zaman harcamaktadır. Ayrıca etmen servisi içersinde java *reflection* için zaman harcanmaktadır. Daha önce de belirtildiği gibi servis içinde bir işlem yapılmadığından, ölçülen süreye bir katkısı olmamaktadır.



Şekil 6.5 : Normal web servis - etmen servis çağrısı karşılaştırması.

Normal web servisi(etmen kullanılmadan gerçeklenen web servisi) çağrısı ile etmen servisi çağrısı karşılaştırılmış ve normal web servisinin etmen servisine oranla, Şekil 6.5'da görüldüğü gibi, çok hızlı sonuç aldığı görülmüştür. Etmen servisi için, çağrı ve sonuç iletme noktasında gerçekleştirilen xml dönüşümleri ve java *reflection* kullanılması bu sonuçta etkilidir. Ayrıca, etmen servisi tek bir etmen üzerinden çağrılmaktadır. Dolayısıyla gelen talepler tek bir etmen tarafından karşılanmaktadır. Bu yüzden sistem üzerinde mesajlar, mesaj kuyruğunda beklemektedir. Bu durum çağrı sayısı artıkça etmen servisi çağrı süresinin de daha fazla artmasına neden olmaktadır. Oysa normal web servisinde her bir servis çağrısı için sadece yeni bir

java nesnesi çağrılmaktadır. Etmen servisi çağrısında yer alan xml dönüşümleri, java *reflection* işlemleri bulunmamaktadır. Bu durum normal web servisinin gelen çağrı sayısı artmasına rağmen etmen web servisi çağrısı süresi kadar artmaması sonucunu doğal olarak yaratmaktadır sağlamaktadır.



Şekil 6.6 : Tekil etmen servis – bütünleşik etmen servis çağrı karşılaştırması.

Bütünleşik etmen servisi çağrıları ile tekil servis çağrısını sürelerini karşılaştırdığımızda ise Şekil 6.6'da gibi bir grafik ortaya çıkmaktadır. Bu senaryoda kullanılan bütünleşik servis iki adet tekil servisten oluşmaktadır. Şekilde de görüldüğü gibi bütünleşik servis çağrısı daha fazla zaman almaktadır. Bu durum bütünleşik servis çağrısı yapılırken EWSP içersinde çağrının ikiye bölünerek ilgili etmen servislerinin ayrı ayrı çağrılmasından kaynaklanmaktadır.

6.2 Test Ortamı Linux

Linux işletim Linux işletim sisteminin yer aldığı test ortamında EWSP uygulamasının yüklü bulunduğu bir bilgisayar, EWSP sanal firmaların (servis sunmaya aday etmenler) çalıştığı bir bilgisayar ve servis taleplerinin ve servis çağrılarının yapıldığı bir bilgisayar olmak üzere üç adet bilgisayar bulunmaktadır. Üç

bilgisayar birbirlerine kablosuz ağ ile bağlanmıştır. Servis çağrıları ve taleplerinin sayısı thread kullanılarak artırılmaktadır.

EWSP Bilgisayar:

EWSP Web Sunucusu, EWSP Yöneticisi, EWSP Etmen'lerinin çalıştığı bilgisayardır ve şu özelliklere sahiptir:

- İşlemci: Intel Core 2 Duo P860 2.4 GHz
- Bellek: 3GB
- İşletim Sistemi: Ubuntu 10.04

Sanal Firmaların Çalıştığı Bilgisayar:

Sanal firmaların, sanal çalışanların ve sanal aday çalışanların yer aldığı bilgisayardır ve şu özelliklere sahiptir:

- İşlemci: : Intel Core i5 CPU M430 2.27 GHz
- Bellek: 4GB
- İşletim Sistemi: Ubuntu 10.04

İstemcinin Çalıştığı Bilgisayar:

Servisler için talepte bulunan ve servis çağrılarının yapıldığı istemcilerin çalıştığı bilgisayardır ve şu özelliklere sahiptir:

- İşlemci: Genuine Intel (R) CPU T2400 1.83 GHz
- Bellek: 2GB
- İşletim Sistemi: Windows XP

Ağ Bilgileri

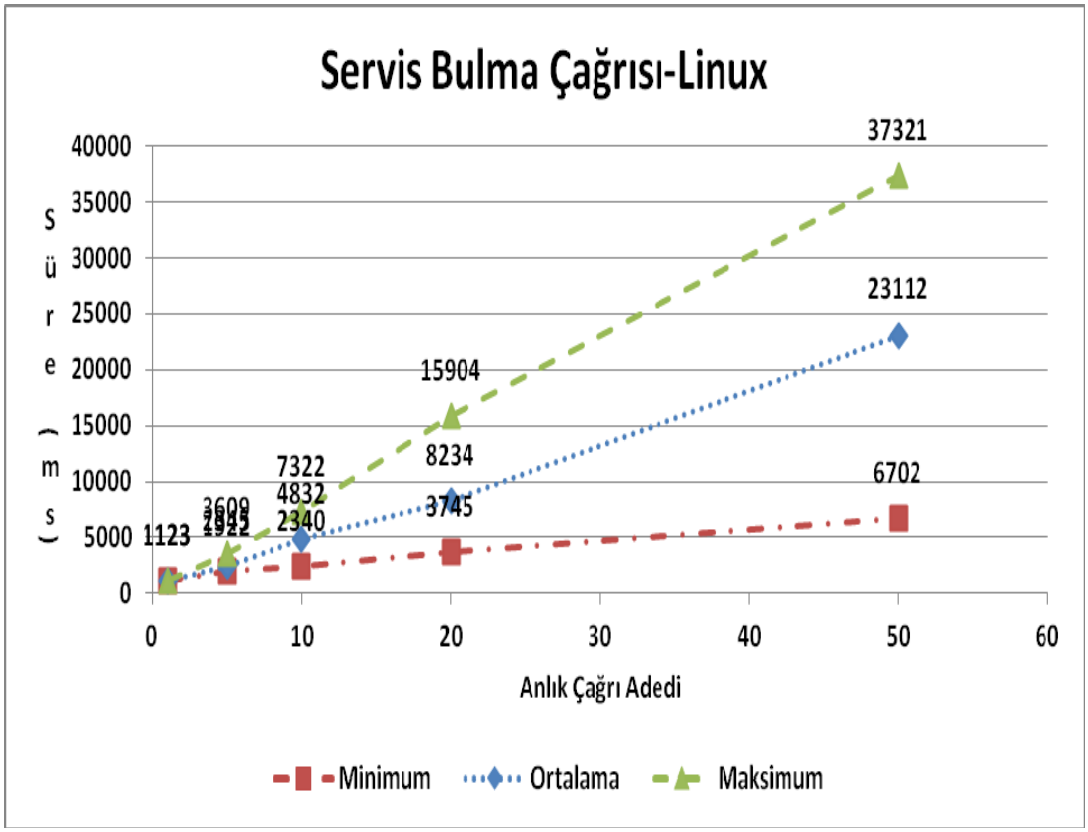
- Kablosuz LAN, IEEE 802.11 54Mbit/s

6.2.1 Linux ile test senaryo 1

Bu test senaryosunda aynı anda yapılan çağrı sayısı arttırılarak servis bulma istekleri yapılmıştır. Servis bulma çağrılarının kriterlerine uyan tek bir servis sistemde bulunmaktadır. İhale yürütücüsü olarak tek EWSP Yönetici etmeni kullanılmıştır.

Bu test senaryosunda turizm firmaları sanallaştırılmıştır. Turizm firmalarının vermiş olduğu servisler servis ihalesine konu edilmiştir.

Dışarıdan gelen her bir servis talebi karşısında EWSP Web gelen talebi SOAP mesajından etmenler arası kullanılan XML mesajına çevirerek EWSP Yönetici etmenine iletmektedir. EWSP Yönetici Etmeni aday servisleri bularak servis ihalesi başlatmaktadır. Her bir servis ihalesi için EWSP etmeni üzerinde bir ihale davranışı çalışmakta ve ihale işlemini sonuçlandırmaktadır.



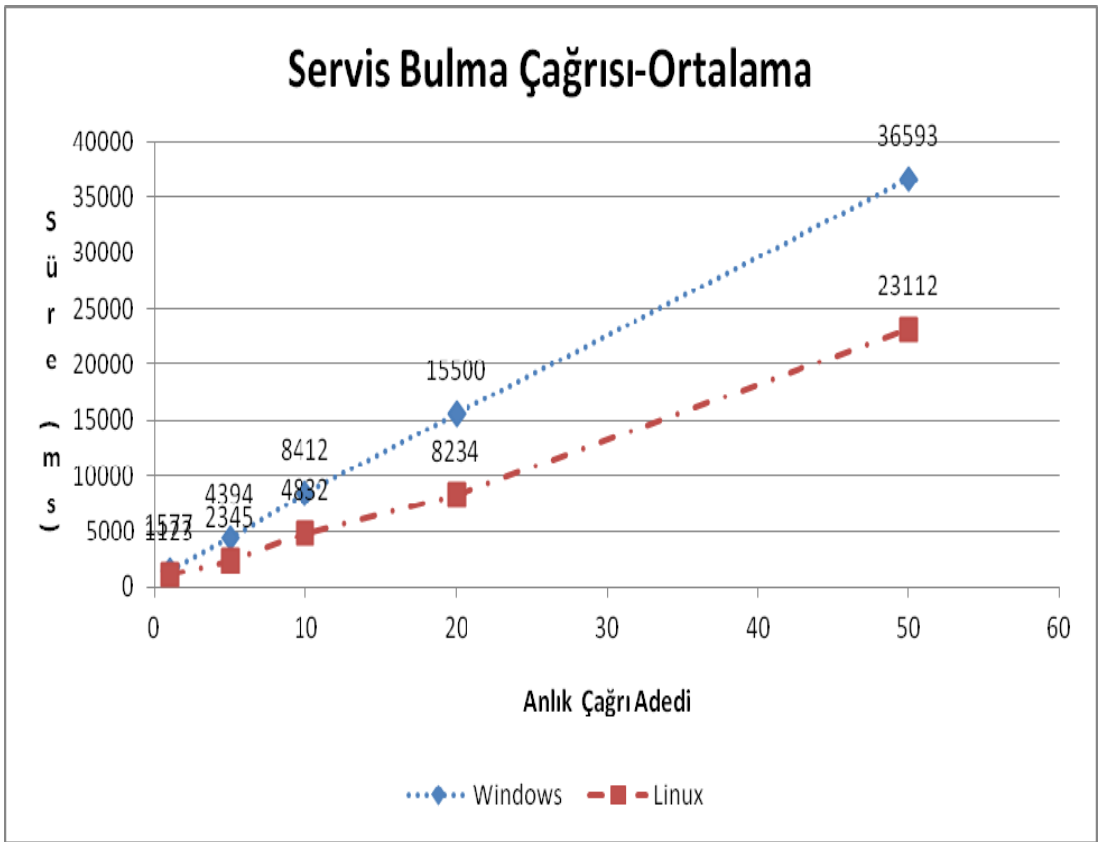
Şekil 6.7 : Tek aday ile servis bulma.

İhale süreci 3 kritik nokta üzerinden geçmektedir; EWSP Web bileşeni, EWSP Yönetici Etmeni ve aday (ihaleye katılan) etmen. EWSP Web bileşeni belirli bir noktaya kadar çoklu istek karşılayabilmektedir. EWSP Yönetici etmeni ve aday etmen kendilerine gelen talepleri sıralı ancak iş kuyruğu şeklinde işlemektedir.

Şekil 6.7'de görüldüğü gibi anlık çağrı sayısı arttıkça ihale sürecini sonlandırma süresi de artmaktadır. EWSP yönetici etmeni ve aday etmeni sayıca aynı kaldığından dolayı gelen çağrıları karşılama süreleri artmıştır. Maksimum işlem süresi hızlı bir şekilde artarken minimum süre çok fazla artmamıştır. Bu zaman geçtikçe gelen çağrıların birikmesi sonucu sistemin yavaşladığını göstermektedir.

6.3 Windows Ve Ubuntu Çalışmaları Karşılaştırılması

Şekil 6.8'de Ubuntu üzerinde çalışma sonuçları ile windows sistem çalışma sonuçları gösterilmiş ve bunlar karşılaştırılmıştır. Luni(x) işletim sistemi tabanlı sistemler java uygulamalarda windows işletim sistemli tabanlı sistemlere göre daha iyi performans göstermektedir. Burada da ubuntu sistemi üzerindeki çalışma windows sistemine göre çok daha iyi bir performans göstermiştir.



Şekil 6.8 : Windows-ubuntu karşılaştırması tek aday ile servis bulma.

6.4 Değerlendirme ve Sonuçlar

Başarım değerlendirmeleri sonucunda anlık çağrı yapan kullanıcı sayısı artıkça işlem süresinin arttığı görülmüştür. İhaleye katılan etmen sayısı mesajlaşmayı artırdığı için ihale süresini etkilemektedir. Talep edilen servisi verebilecek etmenlerin yani ihaleye katılan etmenlerin sayısı ile ihale süresi arasında ilişki bulunmaktadır. Yüksek anlık çağrı sayılarında ihale süresini düşürmek için ihale yürütmekle yükümlü olan EWSP Yönetici etmen sayısı arttırılmıştır. Arttırılan ihale yürütücüsü EWSP etmeni sayısı yüksek miktarda anlık çağrı karşısında ortalama ihale süresini düşürmüştür.

Uygulama windows ve ubuntu işletim sistemleri üzerinde test edilmiştir. Ubuntu sistemi windows sistemine göre çok daha iyi işlem süreleri vermiştir. Genel olarak java uygulamalarının Linu(x) sistemlerinde daha etkin çalışması burada da etkisini göstermiştir. Etmen kullanmayan web servisleri etmen kullananlara göre daha iyi performans göstermiştir. Ancak önerilen sistemin sahip olduğu ve tez çalışmasında ifade edilen özellikleri performanstan kaybın kabullenilmesi için yeterlidir.

7. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında sanal firmaların yönetimi, firmaların servislerini pazarda satabilmesi için rol ve etmen tabanlı web servis pazarı sistemi geliştirmiştir [51,52]. Tez boyunca etmenler, roller ve web servisleri ve bunların arasındaki etkileşim incelenmiş ve sistemin oluşturulması için kullanılmıştır.

Farklı ortamlardan çağrılma özelliğine sahip ve heterojen sistemlerde etkin bir şekilde çalışan web servisleri, kendi başına çalışabilen etmen sistemleri ve bu sistemlere içsel dinamiklik katan roller sanal firmaların oluşumu ve yaşam döngüsü için kullanılmıştır. Etmenler ile web servislerinin birlikte kullanımı web servislerini daha esnek ve dinamik bir hale getirmiştir.

EWSP'de roller servislerin bir bütünü olarak tasarlanmış ve bu yönde geliştirilmiştir. Rollerin sistem içinde kullanımı sisteme bir esneklik getirmiştir. Sanal firma fonksiyonlarının rollere dağıtılması ve roller aracılığı ile yönetilmesi sanal firmalara bir dinamiklik kazanmıştır. Roller çalışma anında yönetilebilmektedir. EWSP sisteminde roller sanal çalışanlara/yöneticilere çalışma zamanında verilip alınabilmektedir. Bir sanal çalışan birden fazla rol sahibi olabilmektedir.

Roller sistemin analiz, tasarım ve gerçekleme adımlarında kullanılmıştır. Sanal firmalar ortaya çıkarılırken(analiz) roller ve bu rollerin fonksiyonları(servisleri) da belirlenmektedir. Tasarım ve gerçeklemede adımında rol tanım dili(EWSPX) kullanılarak roller ve bu rollerin servislerle olan ilişkileri tanımlanabilmektedir.

Sanal firmalar sanal çalışanlardan oluşmaktadır. Roller kullanılarak sanal çalışanlar yetkilendirilmekte ve sanal çalışanlara sorumluluk verilmektedir. Bu şekilde sanal firmalara dinamizm kazandırılmıştır. Sanal firmalar yeni elemanlarını ve servislerini yönetebilmektedir.

Sanal firmalara servislerini satabilecekleri bir ortam sağlanmaktadır. Sanal firmalar kendi servislerini pazarda satabilmekte, servis ihalelerine katılabilmektedir. Web servislerinin kullanılması sanal firmaların servislerinin çok farklı ortamlar tarafından kullanılmasını sağlamakta ve sanal firmaların pazarını büyütmektedir.

Müşteriler için servis satın almak için bir sistem sağlanmıştır. Müşterilerin istekleri tek servis tarafından karşılanamıyorsa birden fazla servisin tek bir servis haline gelmesine olanak verilmiştir. Bu sayede bütünleşik servislerin oluşmasına olanak sağlanmıştır.

Müşteriler servis talepleri için sadece servis kriterlerini değil, aynı zamanda kendi kriterlerinin ağırlıklarını da vererek servis bulabilirler.

Müşteri servis talepleri sonrasında servis sağlama ihalesini kazanan etmen ile ilgili olarak taahhüt takibi yapılmış ve müşteri haklarının korunmasını sağlanmıştır.

EWSP'de sanal firma tanımlamalarının dinamik şekilde yapılması için bir rol tanımlama dili olan EWSPX'i geliştirilmiştir. EWSPX kullanılarak rol tanımları kolay bir şekilde geliştirilebilir. Geliştiriciler kendi sanal firmalarını kolay bir şekilde sisteme entegre edebilirler.

EWSP ölçeklenebilir yapıda kurgulanmıştır. Farklı makinelerde çalışabilmektedir. Servis talebinde ortaya çıkan performans sıkıntılarını çözebilmek için ihtiyaç dahilinde birden fazla EWSP Yönetici etmeni yaratılabilir.

EWSP sistemi içinde geliştirmeye açık noktalar yer almaktadır. Sistem içinde sanal firmalar arası ilişkilerin yönetilmesi için çalışma yapılabilir. Günümüz koşullarında firmalar ortaklıklar kurabilir veya grup/holding çatıları altında birleşebilir. Bu konuda da EWSP sistemi içinde bir çalışma yapılabilir. Yine EWSP içindeki ihale süreci ile ilgili olarak ve özellikle aday servislerin seçilmesi/bulunması yönünde çalışmalar yapılabilir.

KAYNAKLAR

[1] **Huhns, M.N.,** 2002." Agents as Web Services " IEEE Internet Computing (6:4), pp. 93-95.

[2] **Wooldridge, M.,** 1999. Multiagent Systems A Modern Approach to Distributed Artificial Intelligence edited by Gerhard Weiss. p. 41

[3] **Franklin, S. and Graesser, A.,** 1996. "Is it an Agent, or just a program? A taxonomy for Autonous Agents" Proc. Third International Workshop on Agent Theories, Architecures,and Languages, Springer-Verlag.

[4] **Gao, S., Wang, H., Wang, Y., Shen, W., and Yeung, S.,** 2005. "Web-service-agents-based Family Wealth Management System" Expert Systems with Applications Volume **29**, Issue 1, July, Pages 219-228.

[5] **Papazoglou, M. P.,** 2003. ”Service-Oriented Computing: Concepts, Characteristics and Directions”, Proceedings of Fourth International Conference on Web Information Systems Engineering (WISE’03).

[6] **Url** <*http://uddi.xml.orgl*>, alındığı tarih 10.06.2010.

[7] **Url** <*http://www.w3.org/tr/soap/*>, alındığı tarih 10.01.2010.

[8] **Url** <*http://www.w3.org/tr/wsdl*>, alındığı tarih 10.06.2010.

[9] **Lee, Y., Ma, C., Chou, S.,** 2005. ”A Service-Oriented Architecture for Design and Development of Middleware”, Proceedings of 12th Asia-Pasific Software Engineering Conference(ASPEC’05).

[10] **Url** <*http://www.w3.org/TR/2004/NOTE-ws-arch-20040211*>, Web Services Architecture, W3C Working Group Note. Alındığı tarih 11.02.2010.

[11] **Coyle, F.P.,** 2002. XML, Web Services, and the Data Revolution, Information Technology Series, ed. Addison-Wesley.

[12] **Url** <*http://www.w3.org/TR/2008/REC-xml-20081126*>, alındığı tarih 08.04.2009.

[13] **Erl, T.,** 2004. “Service-Oriented Architecture : A Field Guide to Integrating XML and Web Services”, Prentice Hall PTR, ISBN: **9780131428980**, New Jersey, Nisan.

[14] **Url** <*http://www.w3.org/XML/Schema.html*>, alındığı tarih 10.06.2010.

[15] **Url** <http://www.oasis-open.org/home/index.php>, alındığı tarih 10.06.2010.

[16] **Huhns, M. P., Singh, M.P., Burnstein, M., Decker, K., Durfee, E., Finin, T., Gasser, L., Goradia, H., Jennings, N., Lakkaraju, K., Nakashima, H., Parunak, H. V. D., Rosenschein, J. S., Ruvinsky, A., Sukthankar, G., Swarup, S., Sycara, K., Tambe, M., Wagner, T., Zavala, L.**, 2005. ”Research Directions for Service-Oriented Multiagent Systems”, IEEE INTERNET COMPUTING, November-December, p. 65-70.

[17] **Blake, M. B.,** 2003. ”Forming Agents for Workflow-Oriented Process Orchestration”, Workshop on Electronic Commerce, Agents, and Semantic Web Services in conjunction with the International Conference on Electronic Commerce (ICEC2003), October.

[18] **Li, Y., Ghenniwa, H., Shen, W.,** 2003. ”Integrated Description for Web Service-Oriented Agents in e-Marketplaces”, 16th Canadian Conference on Artificial Intelligence AI.

[19] **Rykowski, J., Cellary, C.,** 2004. ”Virtual Web Services-Application of Software Agents to Personalization of Web Services”, ICEC’04, Sixth International Conference on Electronic Commerce.

[20] **Wang, Y., Wang, H., Deng, J., Zhao, X., Yung, K., Gao, S.,** 2005. “Web-Service-Agents-Based Securities Trading Simulation System”, PACIS 2005 Proceedings. Paper 32.

[21] **Valle, M., Ramparany, F., Vercouter, L.,** 2005. “A Multi-Agent System for Dynamic Service Composition in Ambient Intelligence Environments”, The 3rd International Conference on Pervasive Computing (PERVASIVE 2005).

[22] **Hawkins, D. I., Best, R.J., and Coney, K. A.,** 1983. *Consumer Behavior, Business Publications*, Inc., Plano, Texas.

[23] **Genilloud, G., and Wegmann, A.,** 2000. “A Foundation for the Concept of Role in Object Modeling”, *Proceedings of the 4th International Enterprise Distributed Object Computing Conference(EDOC2000)*, Makuhari, Japan, September, pp. 76-85.

[24] **Sandhu, R. S., Coynek, E. J., Feinsteink, H. L., Youmank, C. E.,** 1996. ”Role-Based Access Control Models”, IEEE Computer, February, Volume **29**, p. 38-47

[25] **Selçuk, Y.E., Erdoğan, N.,** 2003. ”How to Solve the Inefficiencies of Object Oriented Programming: A Survey Biased on Role-Based Programming”, 7th Wordl Multiconf. On Systemics, Cybernetics and Informatics (SCI 2003), July. Vol. **XIII**, pp. 160-165.

[26] **Selçuk, Y.E., Erdoğan, N.,** 2005. ”Using Roles with JAWIRO”, Proc. AAAI 2005 Fall Symposium Series:” Roles: An Interdisciplinary Perspective”, November.

[27] **Schrefl, M., Thalhammer, T.,** 2004. ”Using roles in Java”, Software-Practice & Experience, April, p. 449-464

[28] **Rykowski, J., Juszkiewicz, A.,** 2003. “Personalization of Information Delivery by the Use of Agents”, In Proceedings of the IADIS Int. Conference WWW/Internet 2003, pp. 1056-1059.

[29] **Rykowski, J.,** 2005. “ACE Agents – Mass Personalized Software Assistance”, Lecture Notes in Computer Science, Volume **3690**, pp. 587-591.

[30] **Url** <*http://www.agent-software.com/products/index.html*>, alındığı tarih 10.06.2010.

[31] **Zinnikus, I., Elguezabal, G. B., Elvesaeter, B., Fischer, K., Vayssiere, J. A.,** 2006. “Model Driven Approach to Agent-Based Service-Oriented Architectures”, MATES, pp. 110-122.

[32] **Lu, H., Chhabra, M.,** 2006. “Methodology for Agent Oriented Web Service Engineering”, PRIMA 2006, pp. 650-655.

[33] **Bellavista, P., Corradi, A., Monit, S.,** 2005. “Integrating Web Services and Mobile Agent Systems”, 25th IEEE International Conference on Distributed Computing Systems Workshops(ICDCSW’05), pp. 283-290.

[34] **Huhns, N. M.,** 2006. “A Research Agenda for Agent-Based Service-Oriented Architectures”, Cooperative Information Agents CIA, pp. 8-22.

[35] **Cabri, G., Ferrari, L., Leonardi, L.,** 2004. ”Agent Role-based Collaboration and Coordination: a Survey About Existing Approaches”, Systems, Man and Cybernetics, 2004 IEEE International Conference on Volume **6**, 10-12 Oct., pp. 5473-5478.

[36] **Wooldridge, M., Jennings, N. R., and Kinny, D.,** 2000. "The Gaia Methodology for Agent-Oriented Analysis and Design", Journal of Autonomous Agents and Multi-Agent Systems, Vol. **3**, No. 3, pp. 285-312.

[37] **Ferber, J., and Gutknecht, O.,** 1998. ”AALAADIN: A meta-model for the analysis and design of organizations in multi-agent systems", Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98).

[38] **Becht, M., Gurzki, T., Klarmann, J., Muscholl, M.,** 1999. ROPE: Role Oriented Programming Environment for Multiagent Systems", Proceedings of the Fourth IFCIS Conference on Cooperative Information Systems (CoopIS'99), Edinburgh,Scotland, September.

[39] **Jamison, W., Lea, D.,** 1999. "TRUCE: Agent coordination through concurrent interpretation of role-based protocols", in Proceedings of Coordination 99, Amsterdam, The Netherlands, April.

[40] **Url** <*http://polaris.ing.unimo.it/MOON/BRAIN/index.html* >, alındığı tarih 10.06.2010.

[41] **Yan, Q., Shan, L., Mao, X., Qi, Z.,** 2003. ”RoMAS: A Role-based Modeling Method for Multi-Agent System”, Porc. of International Conference on Active Media Technology.

[42] **Cabri, G., Ferrari, L., Leonardi, L.,** 2004. ”Agent Role-based Collaboration and Coordination: a Survey About Existing Approaches”, The 2004 IEEE Systems, Man and Cybernetics Conference, Ocotber.

[43] **Url** <*http://jade.tilab.com/*>, alındığı tarih 10.06.2010.

[44] **Url** <*http://java.sun.com/developer/technicalArticles/ALT/Reflection/>,* alındığı tarih 10.06.2010.

[45] **Url** <*http://www.javabeat.net/articles/30-annotations-in-java-50-1.html*>, alındığı tarih 10.06.2010.

[46] **Bellifemine, F. L., Caire, G.,, and Greenwood, D.,** 2007. Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology). Wiley, April.

[47] **Url** <*http*://*jade.tilab.com/papers/WhitePaperJADEEXP.pdf*>, alındığı tarih 10.06.2010.

[48] **Petrie, C., Bussler, C.,** 2003. "Service Agents and Agents as Web Services" IEEE Internet Computing (7:4), pp. 68-78.

[49] **Ripeanu, M., Singh, M.P., Vazhkudai, S. S.,** 2008. "Virtual Organizations " IEEE Internet Computing (12:2), pp. 10-12.

[50] **Foster, I., Kesselman, C., Tuecke, S.,** 2001. "The anatomy of the grid: Enabling scalable virtual organizations. Internationa" International Journal of Supercomputer Applications, 15(3).

[51] **Erdogan, N., Durmus, A.,** 2009. "A Web Services Market Framework with Role Based Agents” IEEE Int. Conference on Systems, Man, and Cybernetics (IEEE SMC 2009), pp. 4722-4727,Texas, USA.

[52] **Erdogan, N., Durmus, A.,** 2009. "An Agent Based Web Service Market" Studies in Computational Intelligence, Vol **253/2009**, 145-157, Springer Berlin/Heidelberg.

[53] **Grefen, P., Eshuis, R., Mehandjiev, N., Kouvas, G., Weichhart, G.,** 2009. "Internet-Based Support for Process-Oriented Instant Virtual Enterprises" IEEE Internet Computing, Vol **13**, No. 6; IEEE, pp. 65-73.

[54] **Sun, X., Huan, M.,** 2011. "Web and multi-agent based virtual enterprise risk management system" Control and Decision Conference (CCDC),May.

[55] **Rodriguez, S., Perez-Lancho, B., De Paz, J.F., Bajo, J., Corchado, J.M.,** 2009. "Ovamah: Multiagent-based Adaptive Virtual Organizations" 12th International Conference on Information Fusion, July.

[56] **Xue, Y.Q., Chen, A.X., Zhao, Z.K., Wang, X.W.,** 2010. "Web Services-Based and Multi-Agent Information Platform" International Conference on Machine Learning and Cybernetics (ICMLC), July.

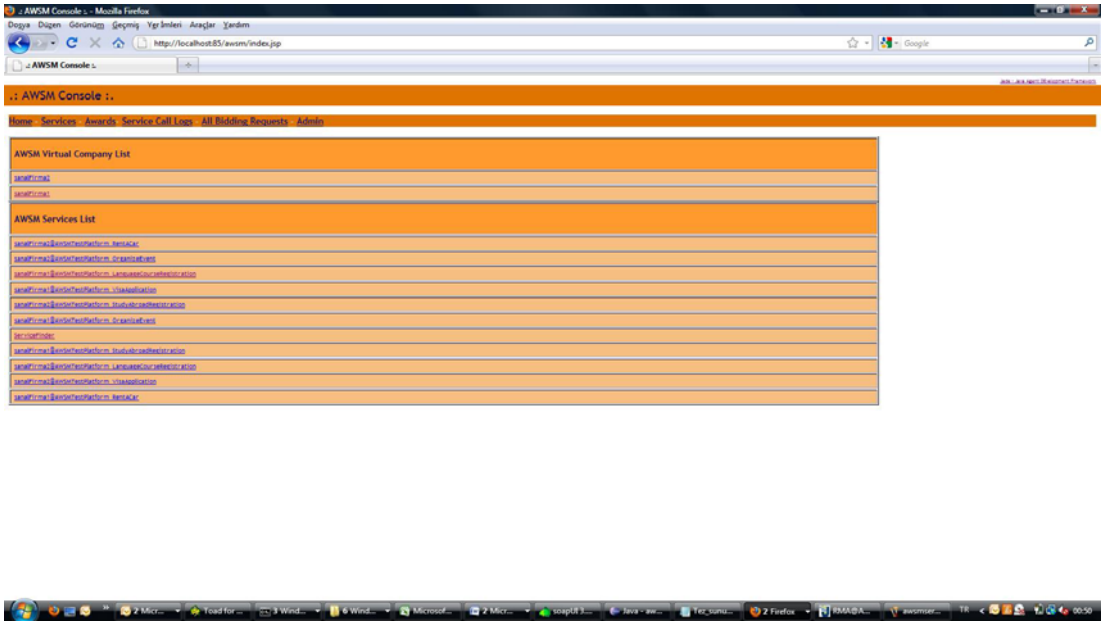
[57] **Brazier, F.M.T., Kephart, J.O., Van Dyke Parunak, H., Huhns, M.N.,** 2009. "Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda" IEEE Internet Computing (13:2), pp. 82-87.

[58] **Herrera, V.V., Bepperling, A., Lobov, A., Smit, H., Colombo, A.W., Lastra, J.,** 2008. "Integration of Multi-Agent Systems and Service-Oriented Architecture for industrial automation" 6th IEEE International Conference on Industrial Informatics, INDIN 2008, July.

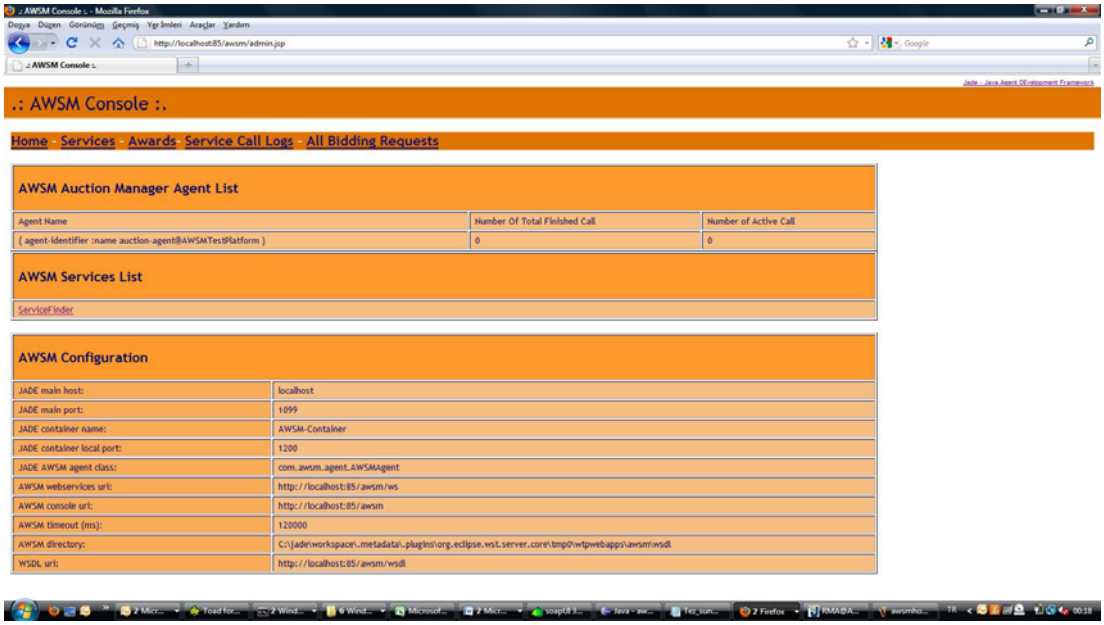
EKLER

EK A: EWSP sistemi yönetici ve izleme ekranları. Sisteme dahil olan etmenlerin ekranları.

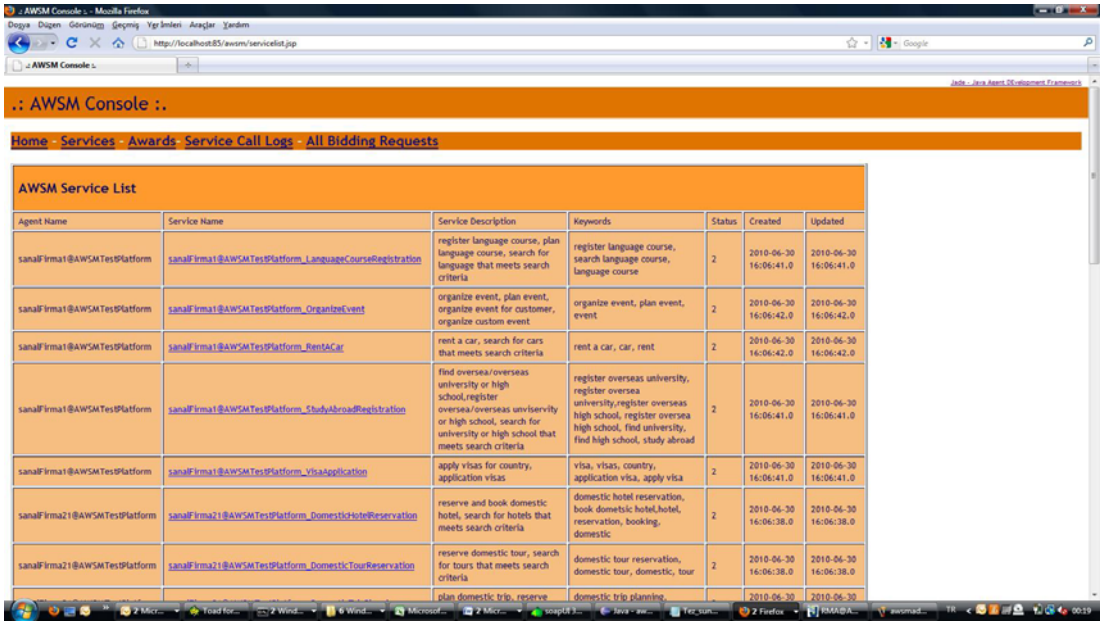
EK A



Şekil A.1 : Uygulama ana sayfa.

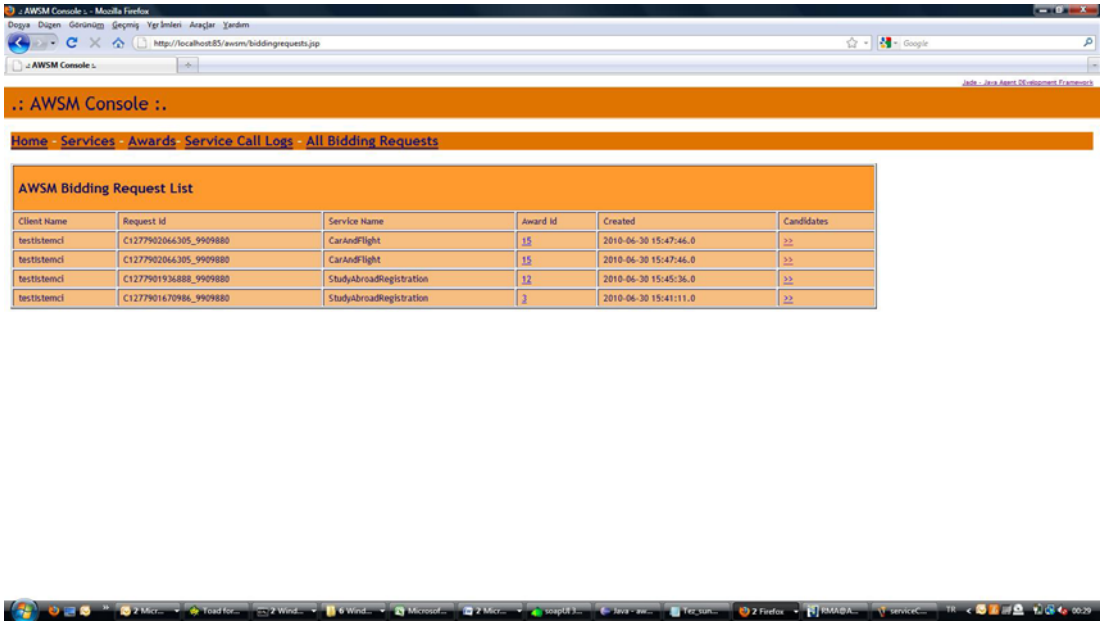


Şekil A.2 : Uygulama yönetici ekranı.



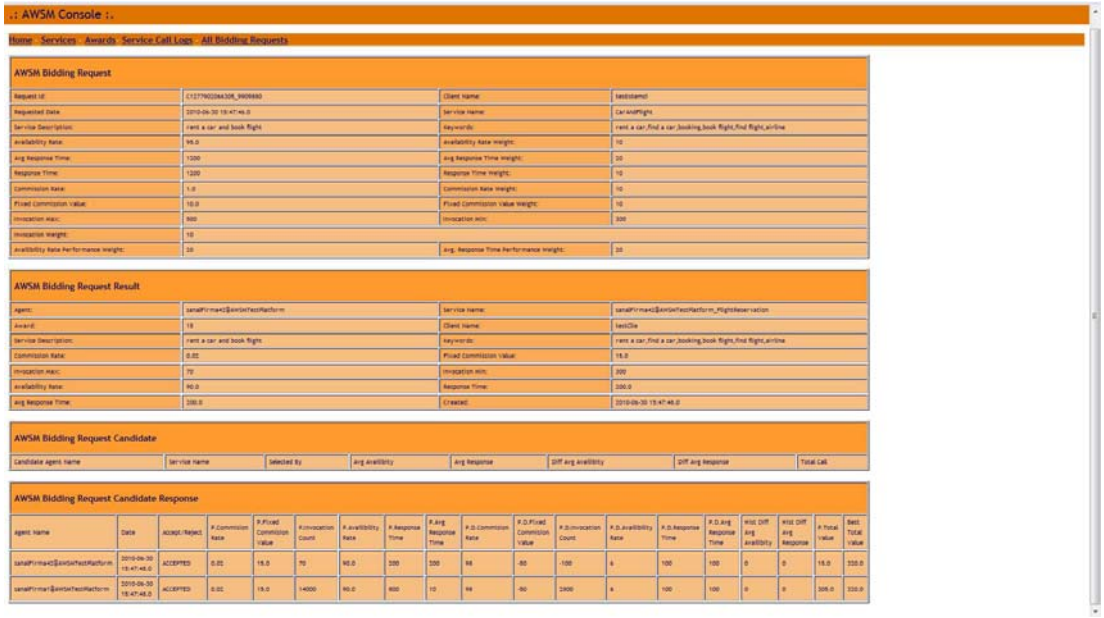
Agent Name	Service Name	Service Description	Keywords	Status	Created	Updated
sanalFirma1@AWSMTestPlatform	sanalFirma1@AWSMTestPlatform_LanguageCourseRegistration	register language course, plan language course, search for language that meets search criteria	register language course, search language course, language course	2	2010-06-30 16:06:41.0	2010-06-30 16:06:41.0
sanalFirma1@AWSMTestPlatform	sanalFirma1@AWSMTestPlatform_OrganizeEvent	organize event, plan event, organize event for customer, organize custom event	organize event, plan event, event	2	2010-06-30 16:06:42.0	2010-06-30 16:06:42.0
sanalFirma1@AWSMTestPlatform	sanalFirma1@AWSMTestPlatform_RentACar	rent a car, search for cars that meets search criteria	rent a car, car, rent	2	2010-06-30 16:06:42.0	2010-06-30 16:06:42.0
sanalFirma1@AWSMTestPlatform	sanalFirma1@AWSMTestPlatform_StudyAbroadRegistration	find oversea/overseas university or high school,register oversea/overseas unviservity or high school, search for university or high school that meets search criteria	register overseas university, register oversea university,register overseas high school, register oversea high school, find university, find high school, study abroad	2	2010-06-30 16:06:41.0	2010-06-30 16:06:41.0
sanalFirma1@AWSMTestPlatform	sanalFirma1@AWSMTestPlatform_VisaApplication	apply visas for country, application visas	visa, visas, country, application visa, apply visa	2	2010-06-30 16:06:41.0	2010-06-30 16:06:41.0
sanalFirma21@AWSMTestPlatform	sanalFirma21@AWSMTestPlatform_DomestickHotelReservation	reserve and book domestic hotel, search for hotels that meets search criteria	domestic hotel reservation, book dometsic hotel,hotel, reservation, booking, domestic	2	2010-06-30 16:06:38.0	2010-06-30 16:06:38.0
sanalFirma21@AWSMTestPlatform	sanalFirma21@AWSMTestPlatform_DomesticTourReservation	reserve domestic tour, search for tours that meets search criteria	domestic tour reservation, domestic tour, domestic, tour	2	2010-06-30 16:06:38.0	2010-06-30 16:06:38.0
		plan domestic trip, reserve	domestic trip planning,		2010-06-30	2010-06-30

Şekil A.3 : Servis listesi ekranı.

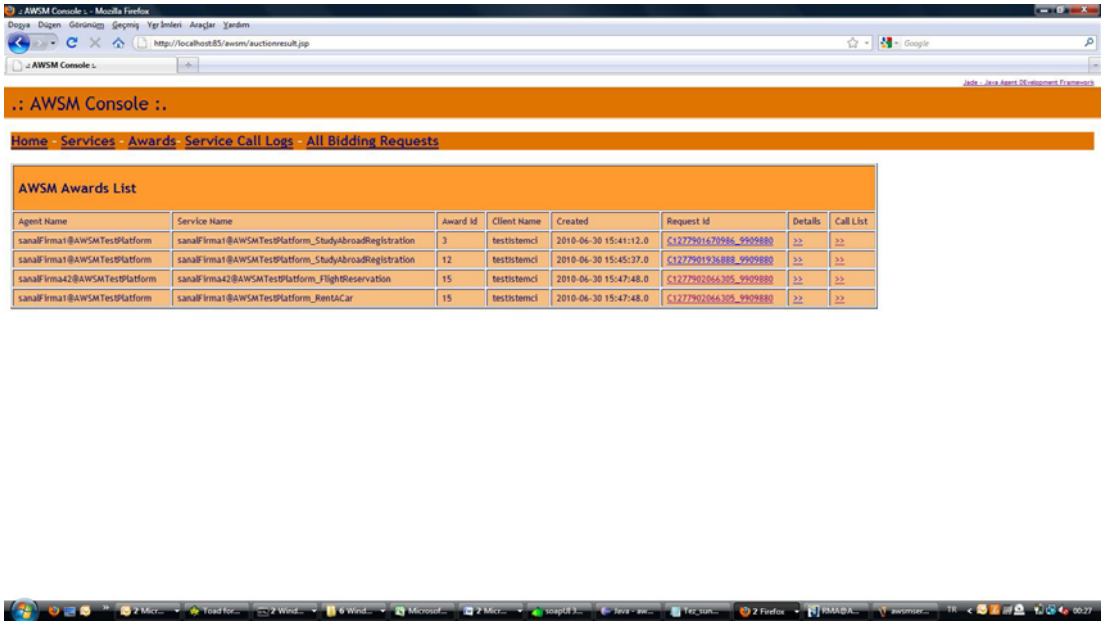


Client Name	Request id	Service Name	Award id	Created	Candidates
testistemci	C1277902066305_9909880	CarAndFlight	15	2010-06-30 15:47:46.0	>>
testistemci	C1277902066305_9909880	CarAndFlight	15	2010-06-30 15:47:46.0	>>
testistemci	C1277901936888_9909880	StudyAbroadRegistration	12	2010-06-30 15:45:36.0	>>
testistemci	C1277901670986_9909880	StudyAbroadRegistration	3	2010-06-30 15:41:11.0	>>

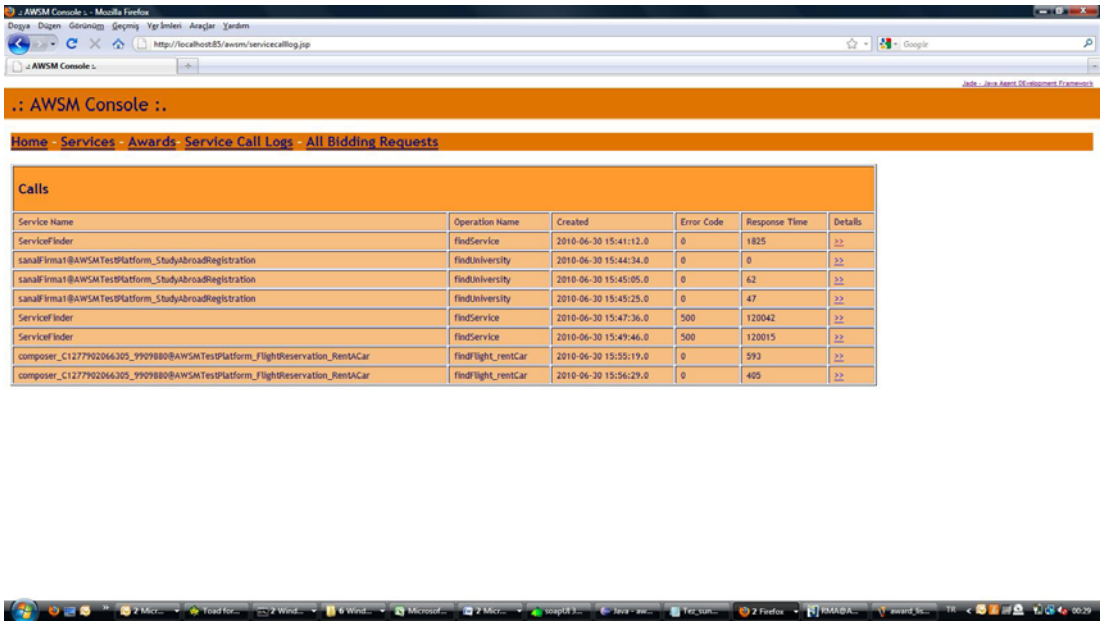
Şekil A.4 : İhale talepleri ekranı.



Şekil A.5 : İhale adayları değerlendirme ekranı.

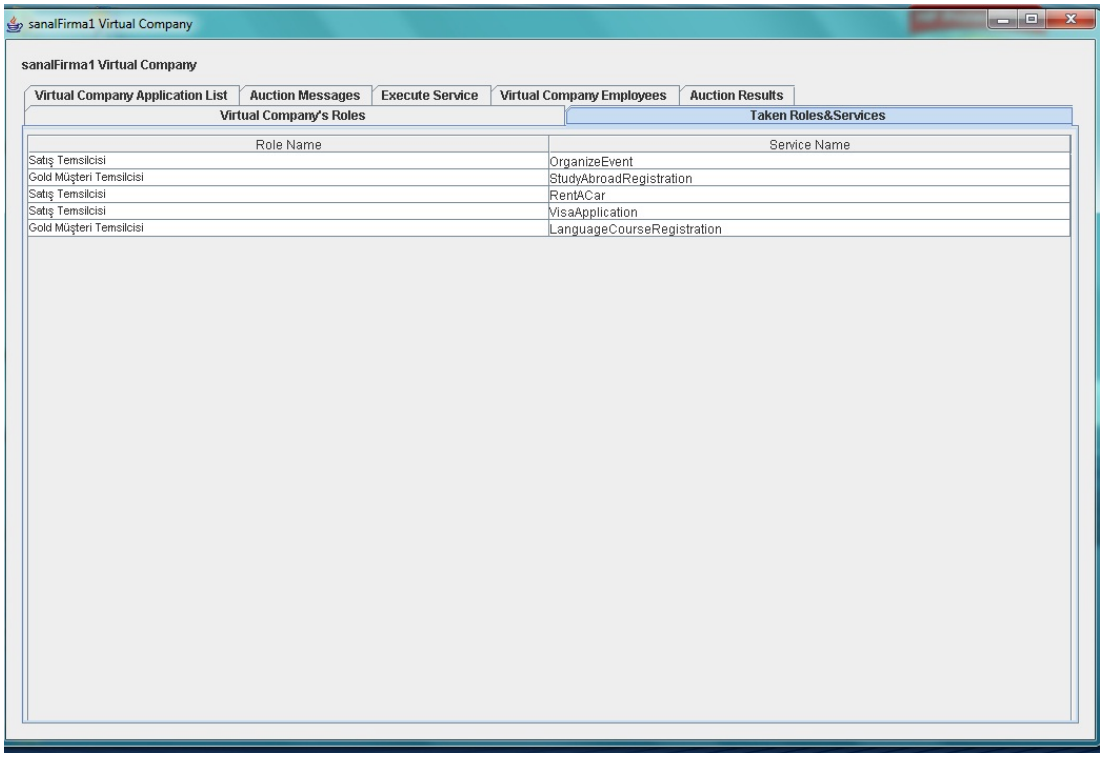


Şekil A.6 : İhale kazananların liste ekranı.



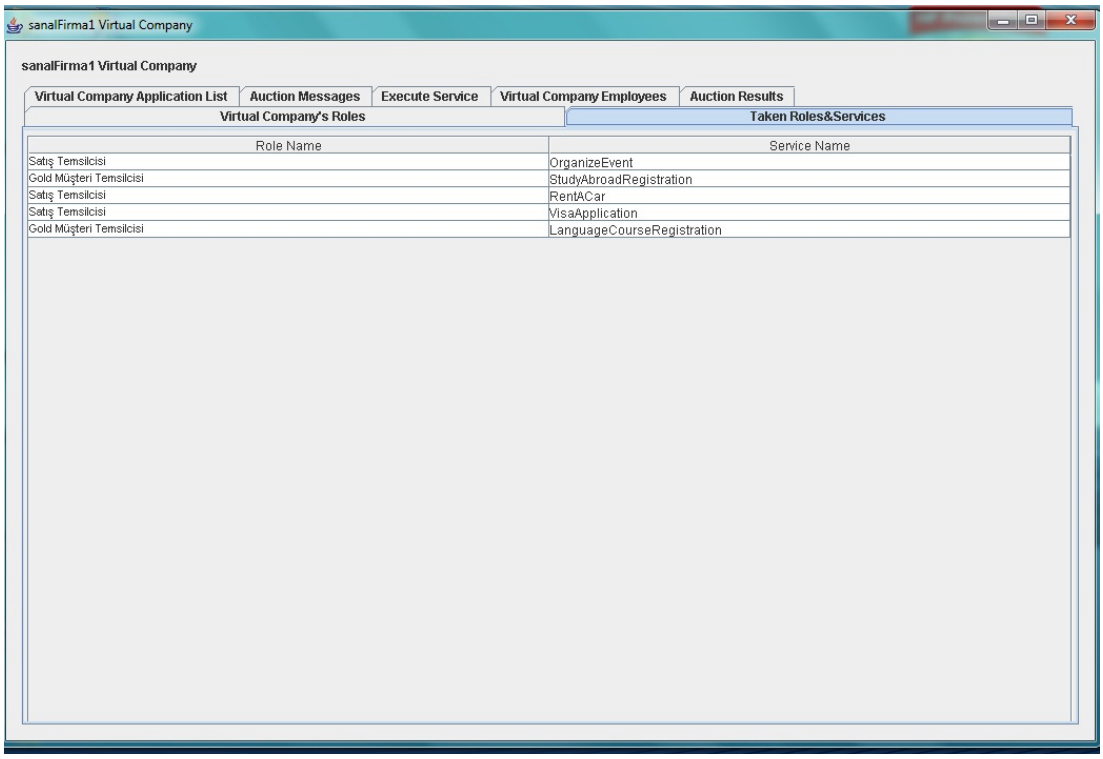
Service Name	Operation Name	Created	Error Code	Response Time	Details
ServiceFinder	findService	2010-06-30 15:41:12.0	0	1825	>>
sanalFirma1@AWSMTestPlatform_StudyAbroadRegistration	findUniversity	2010-06-30 15:44:34.0	0	0	>>
sanalFirma1@AWSMTestPlatform_StudyAbroadRegistration	findUniversity	2010-06-30 15:45:05.0	0	62	>>
sanalFirma1@AWSMTestPlatform_StudyAbroadRegistration	findUniversity	2010-06-30 15:45:25.0	0	47	>>
ServiceFinder	findService	2010-06-30 15:47:36.0	500	120042	>>
ServiceFinder	findService	2010-06-30 15:49:46.0	500	120015	>>
composer_C1277902066305_9909880@AWSMTestPlatform_FlightReservation_RentACar	findFlight_rentCar	2010-06-30 15:55:19.0	0	593	>>
composer_C1277902066305_9909880@AWSMTestPlatform_FlightReservation_RentACar	findFlight_rentCar	2010-06-30 15:56:29.0	0	405	>>

Şekil A.7 : Servis çağrıları liste ekranı.

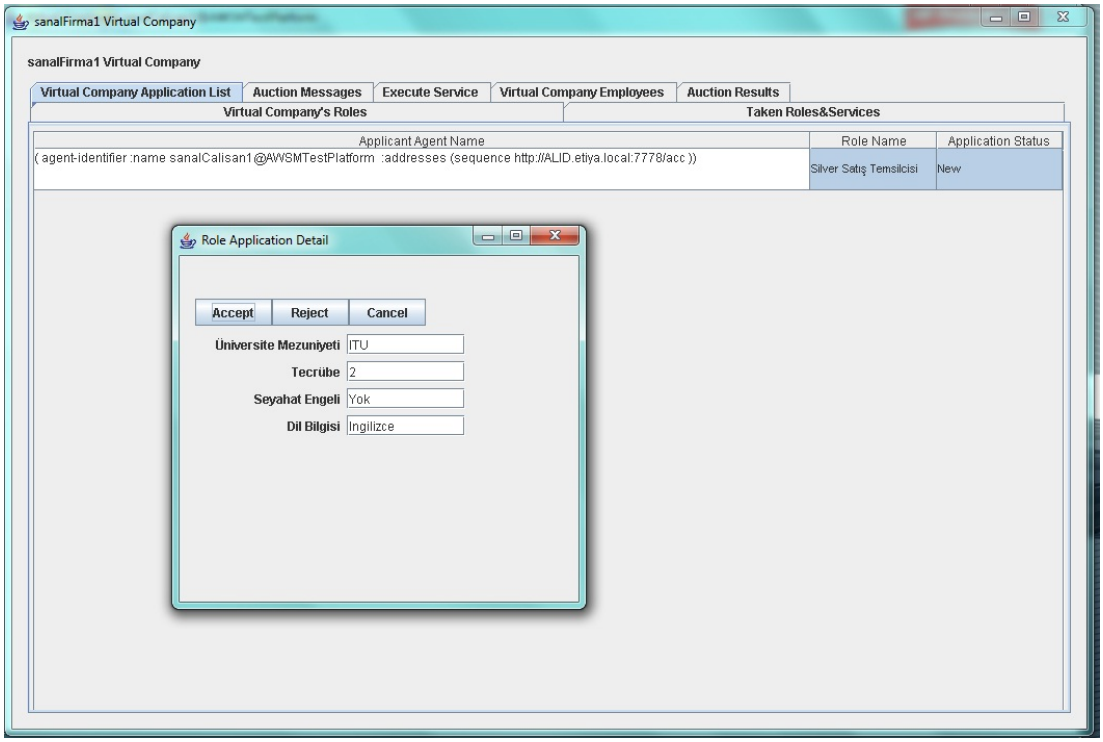


Role Name	Service Name
Satış Temsilcisi	OrganizeEvent
Gold Müşteri Temsilcisi	StudyAbroadRegistration
Satış Temsilcisi	RentACar
Satış Temsilcisi	VisaApplication
Gold Müşteri Temsilcisi	LanguageCourseRegistration

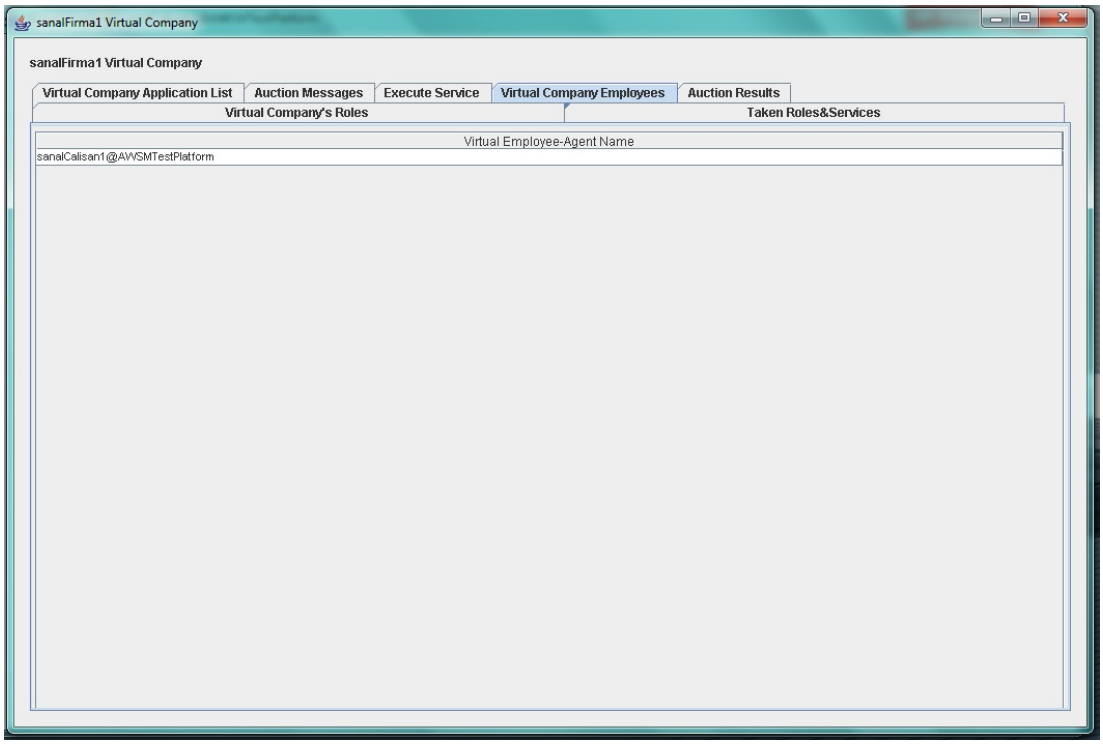
Şekil A.8 : Sanal firma rol listesi.



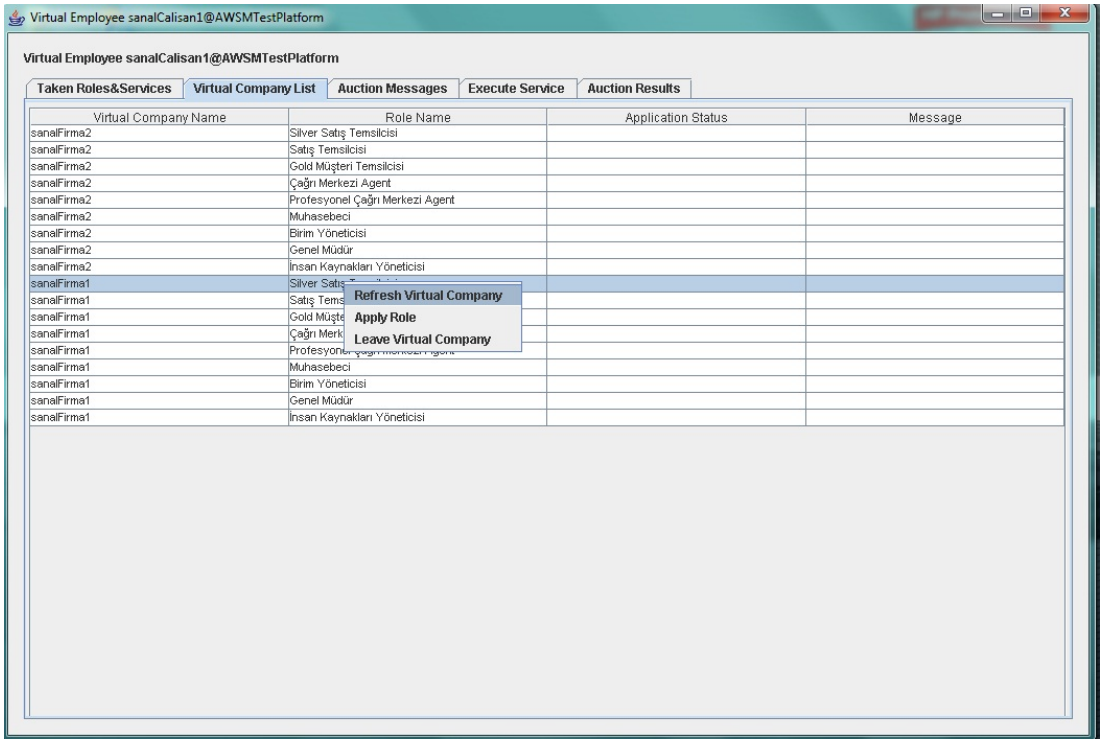
Şekil A.9 : Sanal firma servis listesi.



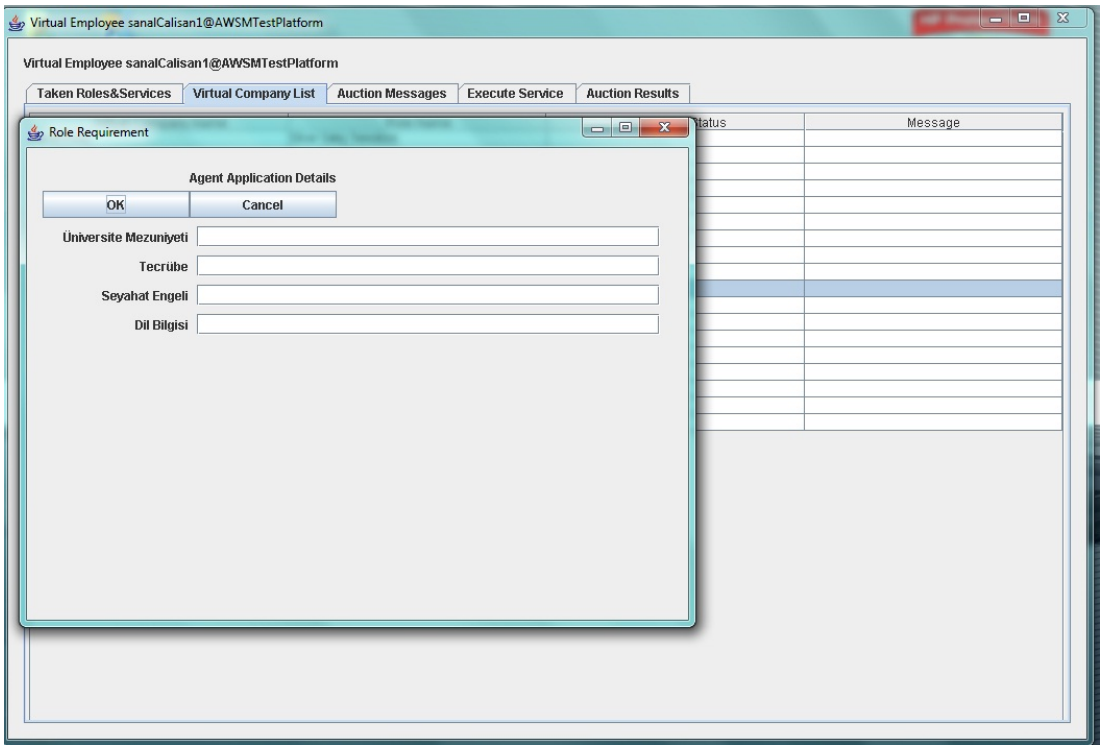
Şekil A.10 : Sanal firma başvuru listesi.



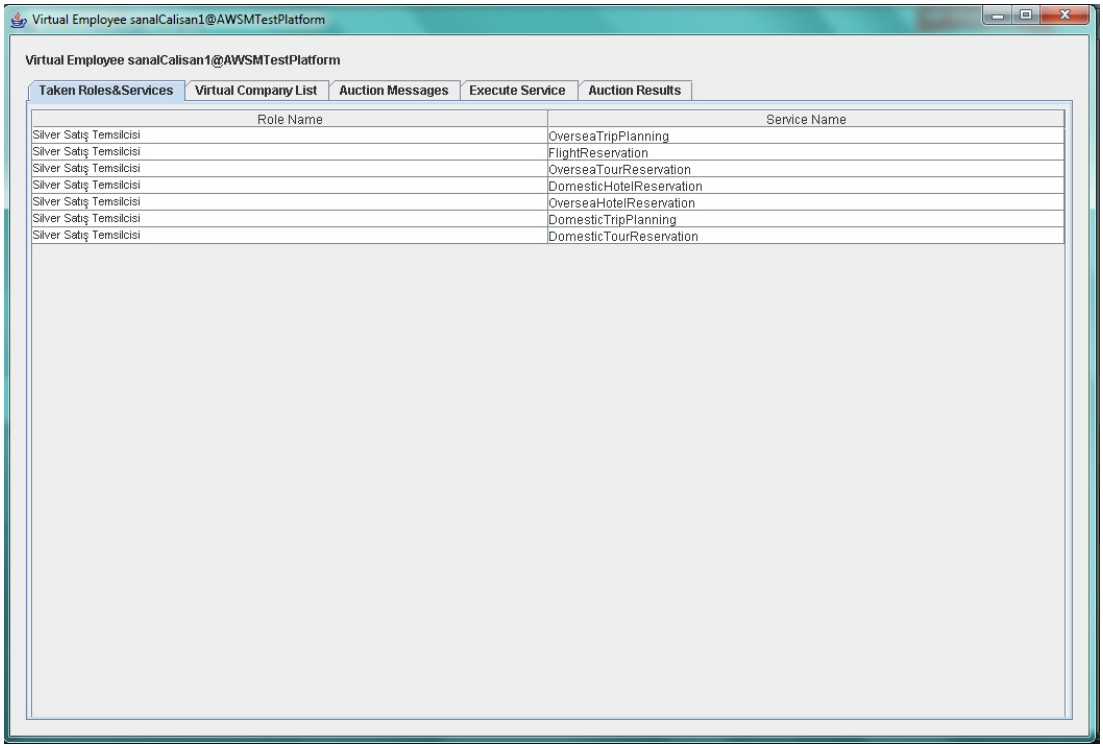
Şekil A.11 : Sanal firma çalışan listesi.



Şekil A.12 : Sanal çalışan firma listesi.



Şekil A.13 : Sanal çalışan sanal firma başvuru ekranı.



Şekil A.14 : Sanal çalışan rol listesi.

sanalFirma1 Virtual Company

sanalFirma1 Virtual Company

Virtual Company Application List | Auction Messages | Execute Service | Virtual Company Employees | Auction Results

Virtual Company's Roles | Taken Roles&Services

Send/Received	Conv Id	Message
Received	C1283120258351_974...	(CFP :sender (agent-identifier :name auction-agent-auction-agent@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc)) :receiver (set (agent-identifier :name sanalFirma1@AWSMTestPlatform)) :content "<auction><serviceName>StudyAbroadRegistration</serviceName></auction>" :reply-by 20100829T221741640Z :protocol service-auction :conversation-id C1283120258351_9745176)
Send	C1283120258351_974...	(PROPOSE :receiver (set (agent-identifier :name auction-agent-auction-agent@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc))) :content "<auction><service><serviceName>StudyAbroadRegistration</serviceName><availabilityRate>90.0</availabilityRate><averageRespons eTime>10</averageResponseTime><commissionRate>0.02</commissionRate><fixedCommissionValue>15.0</fixedCommissionValu e><invocationCount>14000</invocationCount><responseTime>800</responseTime></service></auction>" :reply-with auction-agent-auction-agent@AWSMTestPlatform1283120338857 :protocol service-auction :conversation-id C1283120258351_9745176)
Received	C1283120258351_974...	(ACCEPT-PROPOSAL :sender (agent-identifier :name auction-agent-auction-agent@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc)) :receiver (set (agent-identifier :name sanalFirma1@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc))) :content "9" :protocol service-auction :conversation-id C1283120258351_9745176)
Send	C1283120258351_974...	(INFORM :receiver (set (agent-identifier :name auction-agent-auction-agent@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc))) :reply-with auction-agent-auction-agent@AWSMTestPlatform1283120340043 :protocol service-auction :conversation-id C1283120258351_9745176)

Şekil A.15 : Sanal firma ihale mesajları.

sanalFirma1 Virtual Company

sanalFirma1 Virtual Company

Virtual Company Application List | Auction Messages | Execute Service | Virtual Company Employees | Auction Results

Virtual Company's Roles | Taken Roles&Services

Service Name	Conv Id	AwardId	Result
StudyAbroadRegistration	C1283120258351_9745176	9	GAINED

Şekil A.16 : Sanal firma kazanılan ihale listesi.

sanalFirma1 Virtual Company

sanalFirma1 Virtual Company

Virtual Company Application List | Auction Messages | Execute Service | Virtual Company Employees | Auction Results

Virtual Company's Roles | Taken Roles&Services

Send/Received	awardId	Message
Received	9	(REQUEST :sender (agent-identifier :name ControlAWSM-Container@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc)) :receiver (set (agent-identifier :name sanalFirma1@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc))) :content "<service><serviceName>StudyAbroadRegistration</serviceName><operation><operationName>findUniversity</operationName><operationParameters><countryName>US</countryName><majorName>Computer</majorName></operationParameters></operation></service>" :reply-with R1283120773792_0 :language fipa-sl :reply-by 20100829T222813792Z :protocol fipa-request :conversation-id 9)
Send	9	(INFORM :receiver (set (agent-identifier :name ControlAWSM-Container@AWSMTestPlatform :addresses (sequence http://ALID.etiya.local:7778/acc))) :content "<service><serviceName>StudyAbroadRegistration</serviceName><operation><operationName>findUniversity</operationName><operationReturns><studyId>null</studyId><countryName>null</countryName><message>I couldnt find university for; countryName(US); majorName(Computer)</message><reservationId>null</reservationId><majorName>null</majorName><returnCode>101</returnCode></operationReturns></operation></service>" :reply-with ControlAWSM-Container@AWSMTestPlatform1283120852852 :in-reply-to R1283120773792_0 :language fipa-sl :protocol fipa-request :conversation-id 9)

Şekil A.17 : Sanal firma çağrı listesi.

ÖZGEÇMİŞ

Ad Soyad: Ali DURMUŞ

Doğum Yeri ve Tarihi: Bigadiç, 1976

E-posta: alidur@gmail.com

Lisans Üniversite: Orta Doğu Teknik Üniversitesi Bilgisayar Mühendisliği – 1998

Yüksek Lisans Üniversite: İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği - 2002

Yayın Listesi:

- **Durmus, A.** Erdogan, N.,: An Agent Based Web Service Market. *Studies in Computational Intelligence, Vol* ***253/2009****,145-157, Springer Berlin / Heidelberg, 2009*

- **Durmus, A.** Erdogan, N.,: A Web Services Market Framework with Role Based Agents. *IEEE Int. Conference on Systems, Man, and Cybernetics (IEEE SMC 2009), pp. 4722-4727,Texas, USA, 2009.*

Durmus, A. Erdogan, N.,: A Distributed Multi-Agent Meeting Scheduler System. *Proc. 3rd Asia Pacific Int. Symp. on Information Technology (APIS 2004), Istanbul, Turkey, 2004, pp.179-186.*