Gaia ile Çok-Etmenli Konferans Yönetim Sistemi Analiz ve Tasarımı

Mahmut Tamersoy[1] Bekir Afşar[2] Ferhat Erata[3] Geylani Kardaş[4]

[1,2] Bilgisayar Mühendisliği Bölümü, Ege Üniversitesi, İzmir
[3,4] Uluslararası Bilgisayar Enstitüsü, Ege Üniversitesi, İzmir

[1]e-posta: mtamerso@faculty.ed.umuc.edu [2,4]e-posta: {bekir.afsar, geylani.kardas}@ege.edu.tr
[3]e-posta: ferhat.erata@unitbilisim.com

Özetçe

Çok-etmenli Sistemler'in (ÇES) geliştirilmesi amacıyla literatürde çeşitli ÇES yazılım geliştirme metodolojileri yer almaktadır. Ancak bu metodolojilerin ÇES'leri geliştirmede kullanılmasına ve etmen yazılımı geliştiricilerin faydalanabileceği deneyimlerin aktarılmasına yönelik çalışma pek azdır. Ulusal boyutta ise bu tip bir çalışmanın hemen hemen hiç yapılmadığı gözlenmektedir. Bu bildiride sunulan çalışma ile ulusal boyutta bulunmayan, uluslararası çalışmalarda ise az bulunan ÇES geliştirme deneyimleri ve metodoloji değerlendirme çalışmalarına bir katkıda bulunulması hedeflenmiştir. Çalışmada organizasyon metaforunu kullanan Gaia adlı ÇES yazılımı geliştirme metodolojisinin bir ÇES'in geliştirilmesi sırasında kullanılması ve bu metodolojinin uygulanmasına yönelik elde edilen deneyimler aktarılmıştır. Bu çalışma Gaia'nın, sistem gereksinimlerinin belirlenmesinde, analiz ve tasarım modellerinin gösteriminde bir takım eksiklikleri olsa da ÇES geliştirme için uygun bir metodoloji olduğunu ancak sistemin gerçekleştiriminde yetersiz olduğunu göstermiştir.

1. Giriş

Özerk, karşıt-eylemli ve proaktif yapıdaki yazılım etmenlerinin (*"software agents"*) ve bunların bir araya gelerek oluşturdukları Çok-Etmenli Sistem'lerin ("Multi-agent Systems") (ÇES) tasarımı ve hayata geçirilmesi için Etmen Tabanlı Yazılım Mühendisliği (ETYM) kapsamında çeşitli çalışmalar yürütülmektedir. Söz konusu çalışmaların sonucunda etmen araştırmacıları çeşitli ÇES yazılım mimarileri ve yazılım geliştirme metodolojileri ortaya koymaktadırlar.

Literatürde etmen tabanlı yazılım sistemlerinin gereksinim mühendisliğinin yerine getirilmesi için çeşitli yaklaşımlar sunan ve bir ÇES yazılım geliştirme sürecine ait geliştirme aktivitelerini, aktiviteler arasındaki ilişkileri ve aktivitelerin nasıl gerçekleştirileceğini tanımlayan çeşitli ÇES metodolojileri (örneğin Gaia [1], Tropos [2], Adelfe [3], MaSE [4], Prometheus [5] ve INGENIAS [6]) bulunmaktadır. Bu metodolojilerden bazıları (örneğin MAS-CommonKADS [7]) doğrudan yapay zeka tabanlı iken çoğunluğu ya doğrudan nesne tabanlı metodolojilerin direkt uzantısı halindedir (örneğin PASSI [8] ve Prometheus [5]) ya da hem yapay zeka hem de nesne tabanlı yaklaşımı da içeren bir yapıya sahiptir (örneğin Gaia [1], Adelfe [3], MaSE [4] ve INGENIAS [6]).

Endüstri güdümlü olan ve sistem geliştirme ve yönetmede endüstri tarafından sıklıkla kullanılan nesne yönelimli yazılım geliştirme metodolojilerinin aksine ÇES geliştirme metodolojileri [9]'da da belirtildiği gibi henüz oldukça yenidirler ve günümüzde sadece birkaç küçük endüstriyel uygulamada kullanılmışlardır. Söz konusu metodolojilerin ÇES'leri geliştirmede kullanılmasına yönelik literatürde az sayıda çalışmaya rastlanmaktadır (örneğin [10, 11, 12 ve 13]). Bu çalışmalarda da etmen yazılımı geliştiricilerin faydalanabileceği deneyimlerin aktarılması yerine sadece oldukça genel ve ayrıntıya inmeden metodoloji adımlarının nasıl uygulandığı aktarılmaktadır. Ülkemizde ise etmen tabanlı yazılım sistemlerinin geliştirilmesine yönelik çalışmalar günümüzde az sayıdadır ve bunların pek azı ulusal literatürümüzde yer almıştır (örneğin [14, 15 ve 16]). Bu bildiride sunulan çalışma ile ulusal boyutta hemen hemen hiç bulunmayan, uluslararası çalışmalarda ise az bulunan ÇES geliştirme deneyimleri ve metodoloji değerlendirme çalışmalarına bir katkıda bulunulması hedeflenmektedir.

Çalışmada Gaia ÇES yazılımı geliştirme metodolojisinin bir ÇES'in geliştirilmesi sırasında kullanılması ve bu metodolojinin uygulanmasına yönelik elde edilen deneyimler aktarılmıştır. Gaia [17] toplum organizasyonu metaforunu kullanan ve ÇES'lerin organizasyonel doğası gereği sosyal ilişkileri, etmenler arası bağımlılıkları ve etmenlerin oynadığı rolleri tanımlamak için etkileşim ve eşgüdüm modellerini ortaya koymayı hedefleyen bir ÇES metodolojisidir. Hem ETYM'de geniş kabul görmesi hem de etmen organizasyonel modellerinin tanımlanması ve sistem bileşenlerinin iç ilişkilerinin belirlenmesi amacıyla ÇES geliştiricilerine uygun bir dizi sistem geliştirme adımları sunmasından dolayı çalışmamız için aday metodolojiler arasından Gaia tercih edilmiştir. Üzerinde çalıştığımız sistem ise çok etmenli bir konferans yönetim sistemidir.

Bildirinin geriye kalan kısmı şu şekilde düzenlenmiştir. Bölüm 2'de Gaia metodolojisi kısaca tanıtılmıştır. Bölüm 3'te üzerinde çalışaılan çok-etmenli konferans yönetim sistemi anlatılmıştır. Bölüm 4'te Gaia ile söz konusu sistemin analiz ve tasarımının Gaia metodoloji adımları kullanılarak nasıl yerine getirildiği anlatılmıştır. Elde edilen deneyimler ve çalışmanın değerlendirilmesi Bölüm 5'te yer almaktadır. Bölüm 6'da konu ile ilgili diğer çalışmalar aktarılmıştır. Bölüm 7'de ise sonuç ve ileriye yönelik çalışmalar yer almaktadır.

2. Gaia Metodolojisi

Gaia metodolojisi [17] ÇES'lerin geliştirilmesinde *organizasyon* metaforunu göz önüne alır. Gaia'da bir yazılım sistemi, çeşitli amaçlarına ulaşmak için bir ya da birden fazla rol oynayan ve birbirleri ile etkileşim halinde olan bir dizi etmeni içeren organizasyonların bir bütünü olarak ele alınmaktadır. Buna ek olarak sistem etmenleri, amaçlarını elde etmek için etkileşimde bulundukları bir *ortam* içinde yer

almaktadırlar. Roller, etkileşimler ve ortam haricinde Gaia'nın göz önüne aldığı diğer iki soyutlama *organizasyonel kurallar* ve *organizasyonel yapılar*dır [18]. Bu soyut yapıların modellenmesini içeren Gaia metodolojisinin 3 ana safhası bulunmaktadır: Analiz Safhası (*"Analysis Phase"*), Mimari Tasarım Safhası (*"Architectural Design Phase"*) ve Detaylandırılmış Tasarım Safhası (*"Detailed Design Phase"*). Bu safhalar ve Gaia'nın içerdiği modeller hakkında aşağıdaki alt bölümlerde bilgi verilmiştir.

2.1. Analiz Safhası:

Analiz safhasının amacı organizasyon rolleri, bu rollerin birbirleri olan ilişkileri ve roller arası etkileşimler cinsinden organizasyonun öncül yapısının ortaya konmasıdır. Analiz safhasında sistem *ortam modeli* (*"enviromental model"*), *ön rol modeli* (*"preliminary rol model"*) ve *ön etkileşim modeli* (*"preliminary interaction model"*) ile bir dizi *organizasyonel kural* hazırlanır.

Ortam modelinde bir ÇES'in yerleştiği ortam, etmenlerin algılayabileceği ve onlara dayalı olarak harekette bulunabilecekleri *soyut kaynaklar*ın bir bütünü olarak kabul edilir. Söz konusu kaynaklar etmenin sahip olduğu bilgiler olabilir. Gaia'da ortam kaynaklarının gösterimi için basit bir grafiksel gösterim önerilmektedir.

Gaia'da etmen rolleri 4 özellik ile tanımlanır: *sorumluluklar*, *izinler*, *aktiviteler* ve *protokoller*. Bir rolün işlevselliğini tanımlayan sorumluluklar *canlılık* (*"liveness"*) ve *emniyet* (*"safety"*) özellikleri ile ifade edilir. Canlılık özellikleri rolü üstlenen etmenin sistem için yararını ve buna dayalı olarak canlılığını ifade eder. Emniyet özellikleri sisteme zarar gelmemesi ve her zaman için sistemin kabul edilebilir bir durumda olmasını ifade eder. Öte yandan izinler rollerin kullanabileceği ve kullanamayacağı bilgi kaynaklarını tanımlayarak rollerin haklarını ifade eder. Bir rolün aktiviteleri ise etmenlerin başka etmenlerle etkileşimi olmadan yürüttükleri atomik görevlerdir. Son olarak protokoller bir rolün başka rollerle olan etkileşimlerine ait özel desenler olarak tanımlanmıştır [19].

Gaia'da canlılık özellikleri *canlılık ifadeleri* (*"liveness expressions"*) ile belirtilir. Bu ifadelerde kullanılabilecek operatörler Tablo 1'de verilmiştir. Bir canlılık ifadesinin genel yapısı aşağıdaki gibidir:

ROL_ADI = *ifade*

ROL_ADI canlılık özelliklerinin tanımlandığı rolün adıdır. *ifade* bu ROL_ADI'nın canlılık özelliklerini tanımlar. Canlılık ifadeleri aktiviteler ve protokoller cinsinden verilir. Canlılık ifadelerindeki aktiviteler altıçizili olarak gösterilir. Güvenlik özellikleri bir hükümler (*"predicate"*) listesi ile belirtilir.

Tablo 1: Gaia canlılık ifadeleri için operatörler

Operatör	Açıklama
x.y	x, y tarafından takip edilir
x | y	x veya y meydana gelir
x*	x, 0 veya daha fazla meydana gelir
x+	x, 1 veya daha fazla meydana gelir
x^{w}	x sonsuz kez meydana gelir
[x]	x seçimliktir
x || y	x ve y girişiktir (*"interleaved"*)

Bir rol modeli bir dizi rol şemasından (*"role schema"*) oluşur. Her bir rol şemasında bir role ait yukarıda sözü edilen özellikler gösterilir.

Analiz safhasının ikinci çıktısı olan öncül etkileşim modelinde etkileşim protokolleri etkileşimde bulunan rollerin karakteristikleri ve dinamikleri cinsinden tanımlanır. Tanımlanan protokol özellikleri *protokol adı*, *protokol başlatıcı ("initiator"), protokol ortakları ("partner"), girdiler, çıktılar* ve *metinsel protokol tanımı*dır [17].

Öte yandan sistem organizasyonunun mimari tasarım safhasında düzgün tanımlanması için organizasyonda uyulması gereken kuralların da analiz safhasında belirlenmesi gerekmektedir. Organizasyonel kurallar zamanda bir anı ya da bir zaman dilimini dikkate alarak *anlık* (*"instantaneous"*) ya da *zamansal* (*"temporal"*) olabilirler [18]. Organizasyonel kuralların formal gösterimi için Gaia'da *birinci derece zamansal mantık* (*"first-order temporal logic"*) ifadelerinin kullanılması önerilmektedir. Tablo 2'de organizasyonel kuralların gösteriminde kullanılabilecek zamansal bağlaçlar (*"temporal connective"*) listelenmiştir. Zamansal modeller ○, ◊, □, U, □ ve □ sembollerini sırasıyla şu zaman durumlarını göstermek için kullanırlar: *sonraki*, *en sonunda*, *her zaman*, *olana dek*, *olmadan* ve *önceki*.

Tablo 2: Zamansal bağlaçlar

Formül	Açıklama
○γ	γ bir sonraki anda doğru
◊γ	γ en sonunda doğru
□γ	γ her zaman doğru
γUθ	θ doğru olana kadar γ doğru
γ□θ	θ doğru değilse γ doğru
γ□θ	θ doğru olmadan önce γ doğru

2.2. Mimari Tasarım Safhası:

Bu safhanın öncelikli amacı üzerinde çalışılan ÇES'in işlevsel karakteristiklerine ve ÇES ortamının analiz safhasında belirlenmiş olan operasyonel karakteristiklerine en uygun organizasyon yapısının belirlenmesidir.

Sistemin *organizasyonel yapısı* bir topoloji ve bir kontrol rejimi (*"control regime"*) doğrultusunda ortaya konur. Organizasyonel yapının bir topolojiye uygun olarak tanımlanması ile ÇES'e en uygun etmen organizasyon yapısının seçilmesi amaçlanmaktadır. Kontrol rejimi organizasyon üyeleri arasındaki etkileşimin nasıl olacağını ve kontrol mekanizmalarını belirler. Bir organizasyondaki üyelerin birim zamanda işleyebilecekleri bilgi sınırlı olduğundan üyeler arasındaki yük dağılımının makul olması ve aynı zamanda sistem canlılık ve güvenlik koşullarının sağlanması gerekmektedir. Buna uygun organizasyon yapısı belirlenmeye çalışılır. ÇES'lerin gerçek dünya için organizasyonu önemli olduğundan Gaia'daki bu düzenleme Simon'un [20]'de tanıttığı *Organizasyon Teorisi*'ne dayanmaktadır.

ÇES organizasyonel yapısı belirlendikten sonra bu safhada analiz safhasında ortaya konan başlangıç rol ve etkileşim modellerinin detaylandırılması ve tamamlanması yerine getirilir. Sistem tasarımcısı hangi rollerin hangi rollerle etkileşimde bulunacağını (topoloji) ve bu etkileşimler

sırasında hangi protokollerin işletileceğini (kontrol rejimi) belirler.

2.3. Detaylandırılmış Tasarım Safhası:

Sistemin genel mimarisi tüm rolleri ve etkileşimleri ile belirlendikten sonra bu safhada ÇES'in hayata geçirilmesi için gerekli olan etmen ve servis modelleri hazırlanır. *Etmen Modeli*'nde hangi etmen sınıflarının hangi rolleri oynayacağı ve hangi etmen sınıfından kaç tane (*"instance"*) başlatılacağı belirtilir. Etmen sınıfları ve roller arasında birebir eşleme olacağı gibi birden fazla rol sistemin etkinliği açısından bir etmen sınıfına atanabilir. Öte yandan *Servis Modeli*'nde etmen rolleri ile ilişkili tüm servisler tanımlanır. Servisler nesne tabanlı bakış açısından etmen sınıflarının fonksiyonel metotları olarak kabul edilebilir. Ancak etmenlerin özerk yapısı ve servislerini çalıştırıp çalıştırmama kontrollerinin tamamen etmenlerin kararı olmasından dolayı klasik nesne yaklaşımından farklı bir durum söz konusudur. Her bir servis için Gaia'da servisin girdileri, çıktıları, çalışma ön koşulları ve sonlanma koşulları belirtilir. Bu özellikler daha önceki Gaia geliştirim safhalarının çıktıları olan modellerde yer alan bilgilerden (örneğin protokoller, kurallar, vb.) türetilir.

3. Konferans Yönetim Sistemi

Bu çalışmada Gaia metodolojisi uygulanarak analiz ve tasarımı yerine getirilen yazılım sistemi etmen tabanlı bir konferans yönetim sistemidir (KYS). Sistem bilimsel konferansların yönetilmesini destekleyen bir ÇES'dir. Sistem 3 ayrı aşama ile tanımlanmıştır: *bildiri gönderimi*, *bildiri değerlendirme* ve *karar verme*. Bildiri yazarları gönderim aşamasında hazırladıkları bildirileri bildiri yöneticisine (*"paper manager"*) gönderirler. Bildiri yöneticisi bildiriyi aldığına dair yazarları bilgilendirdikten sonra bildirinin özetini ayrıştırarak hakemlere bildirileri atayacak olan kişilere bildirileri yollar. Bildiri atayıcıları (*"assigners"*) konferans program komitesi (PK) üyeleri olup gönderim aşaması tamamlandıktan sonra bildirileri uygun hakemlere yollamaktan sorumludur. Hakemler (*"reviewer"*) PK üyeleri olabileceği gibi dışardan birileri de olabilir. Hakemlerin değerlendirmesi tamamlandıktan sonra toplayıcı(lar) (*"collector"*) bildiri değerlendirmelerini toplar ve bir karar vericiye (*"decision maker"*) son kararın belirlenmesi için gönderir. Karar verici yazarlara bildirileri hakkında verilen kararları (kabul veya red) bildirir.

Doğal olarak KYS'de yer alacak yazılım etmenlerinin gerçek hayatta böyle bir sistemde yer alan insanların davranışlarını sergilemesi beklenmektedir. Yönetim sisteminde yer alan etmenler bencil veya kendi çıkarlarını güden davranışlarda bulunabilirler (örneğin bir hakemin kendi bildirisini değerlendirmek istemesi). Ayrıca yine etmenlerin özerkliklerinden kaynaklanan ve sistemin işleyişini aksatacak davranışlar olabilir (örneğin bir hakemin kendisine atanan bir bildiriyi değerlendirmek istememesi). Bir sonraki bölümde bu tip davranışları yazılım geliştiricilerin Gaia'da organizasyonel kurallar aracılığı ile kontrol etmesi örneklenmektedir.

4. Çok-etmenli Konferans Yönetim Sistemi'nin Analiz ve Tasarımı

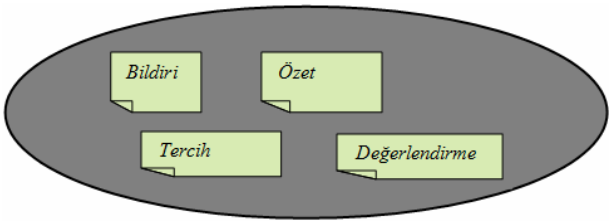
Bölüm 3'te iş akışı ve aşamaları verilen KYS'nin etmen tabanlı olarak geliştirilmesi için uyguladığımız Gaia temelli yazılım geliştirme süreci aşağıdaki alt bölümlerde anlatılmaktadır.

4.1. Analiz:

KYS'nin analiz safhasında Bölüm 2.1'de belirtildiği gibi çeşitli analiz modelleri ve sistem organizasyonel kuralları hazırlanmıştır.

4.1.1. Ortam Modeli:

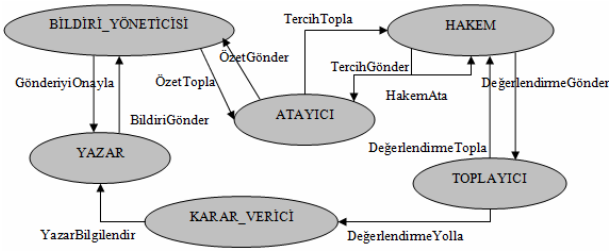
KYS'de yer alan etmenlerin kullandığı kaynaklar *bildiriler*, *özetler*, *hakem tercihleri* ve *değerlendirmeler*den oluşur. Şekil 1'de bu kaynakların yer aldığı etmen ortam modeli yer almaktadır. Yazarlar bildirileri oluşturur. Bildiri yöneticileri özetleri ayrıştırır. Hakemler kendilerine ait tercihleri oluştururlar ve bu tercihlere bağlı olarak kendilerine atanan bildirileri değerlendirirler. İleride anlatılacak olan rol modelindeki rol şemaları ortam modelindeki kaynakların roller tarafından nasıl kullanılacağına dair bilgi vermektedir.



Şekil 1: KYS ortam modeli

4.1.2. Ön Rol ve Etkileşim Modeli:

Şekil 2'deki modelde başlangıçta sistemde var olması düşünülen roller ve bu roller arasındaki etkileşimler yer almaktadır. Roller içi dolu elips şekilleri ile aradaki etkileşimler ise yönlü oklarla modelde gösterilmiştir. Rollere ait başlangıç rol şemaları da bu safhada çıkarılmıştır ancak bu şemalar mimari tasarım safhasında düşünülen organizasyonel yapıdan dolayı değişikliğe uğrayacağından en son halleri bildirinin takip eden ilgili bölümünde anlatılmaktadır.



Şekil 2: Roller ve etkileşimlere ait ön model

4.1.3. Organizasyonel Kurallar:

Zambonelli ve ark. [18]'de organizasyonel kuralları birinci derece zamansal mantık ile ifade edebilmek için üç tane hüküm ve bir tane fonksiyon önermişlerdir:

1. *oynar (i,r)*; *i* etmeni *r* rolünü oynar anlamına gelmektedir.
2. *başlat (r,p)*; *p* protokolünün işletilmesini *r* rolü başlatır anlamına gelmektedir.
3. *katılır (r,p)*; *p* protokolünün işletilmesinde *r* rolü görev alır anlamına gelmektedir.
4. *eleman_sayısı (r)*; *r* rolünü üstlenen kaç tane etmen olduğunu döndüren fonksiyondur.

Çalışmamızda bir bildirinin değerlendirilmesi için gereken hakem sayıları ve nihai kararı vermek için gerekli hakem değerlendirme sayısı yazılan kurallarda parametre olarak yer alırlar. Ayrıca bu iki parametre arasındaki ilişkiyi tanımlamak için de son bir kural eklenmiştir. Değerlendirme süreci için hazırlanan organizasyonel kurallardan bazıları şunlardır:

1. Her bir bildiri (***b***) en az ***n*** tane hakem tarafından değerlendirilmelidir.

$\forall b.eleman\_sayısı(HAKEM(p)) \geq n$

2. Bir hakeme (***i***) aynı bildiri birden fazla değerlendirilmek üzere gönderilmemelidir (emniyet özelliği).

$\forall i. \forall b.oynar(i, HAKEM(b)) \Rightarrow \Box \neg oynar(i, HAKEM(b))$

3. Bir yazar kendi bildirisini bir hakem olarak değerlendiremez .

$\forall i. \forall b.oynar(i, YAZAR(b)) \Rightarrow \Box \neg oynar(i, HAKEM(b))$

4. Bir bildiri toplayıcısı kendisinin yazar olduğu bir bildiriye ait değerlendirmeyi görmemelidir.

$\forall i. \forall b.oynar(i, YAZAR(b)) \Rightarrow \Box \neg oynar(i, TOPLAYICI(b))$

Protokollerin işletilmesi için hazırlanan organizasyonel kurallardan bazıları aşağıda verilmiştir:

1. Eğer bir bildiri alınmışsa mutlaka değerlendirilmelidir (canlılık özelliği).

$\forall i. \forall b.katılır(i, HakemAta(b)) \Rightarrow$
$\Diamond$ *başlat (i, DeğerlendirmeGönder(b))*

2. Hakem değerlendirmek üzere kendisine atanmamış bir bildiriye ait değerlendirme gönderemez (emniyet özelliği).

$\forall i. \forall b.katılır(i, HakemAta(b)) \Box$ *başlat (i, DeğerlendirmeGönder(b))*

3. Bir bildirinin kabulüne veya reddine ***m*** tane değerlendirme alınmadan karar verilemez.

$\forall b.gönderilenDeğerlendirmeler(b) >$
$m \Box$ *başlat (KARAR_VERİCİ,**karar**(b))*

4. Bir bildiri için alınan değerlendirme sayısı, o bildiriye atanan hakem sayısından büyük olamaz.

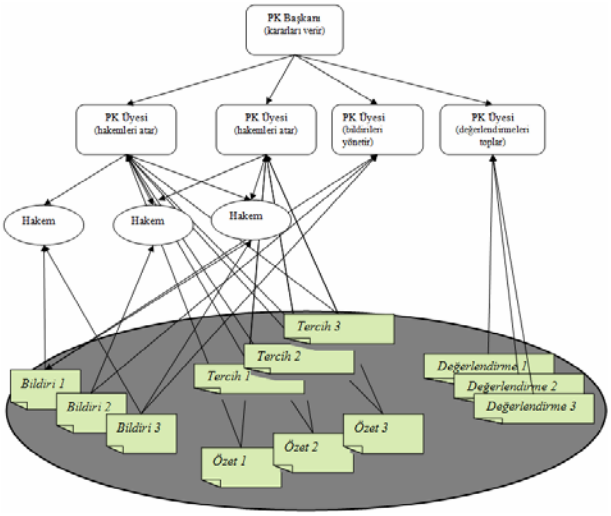
$\forall b.gönderilenDeğerlendirmeler(b) \leq$
eleman_sayısı(HAKEM(b))

4.2. Mimari Tasarım:

KYS için bu sistem geliştirme safhasında organizasyonel yapı tanımlanmıştır ve analiz safhasının rol ve etkileşim modelleri tamamlanmıştır.

4.2.1. Organizasyonel Yapı:

Bu çalışmada büyük ve orta ölçekli KYS'ler için çok-seviyeli hiyerarşik bir organizasyonel yapı hazırlanmıştır. Şekil 3'te gösterilen bu yapıda PK başkanı gelen bildirilerin sayısına göre bölümleme yaparak bir veya birden fazla atayıcıya bildirileri paylaştırabilir. Atayıcılar kendilerine düşen bildirileri uygun sayıda hakemlere gönderirler. Bu açıdan bakıldığında sistemde kullanılan organizasyonel yapının hiyerarşide en üst seviyede yapılan iş paylaşımı kontrol rejimine dayalı olduğu görülmektedir.

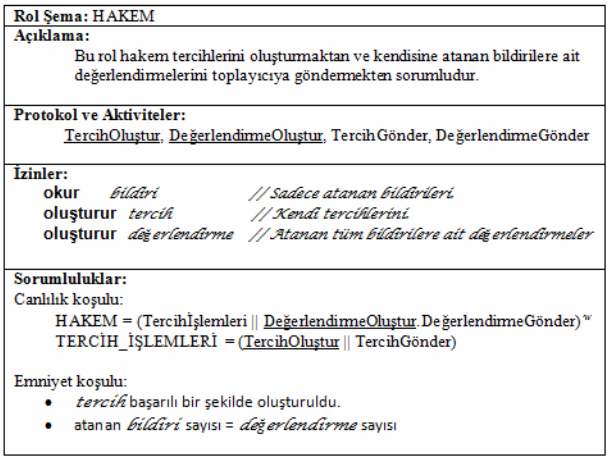


Şekil 3: KYS Organizasyonel Yapısı

4.2.2. Rol Modeli:

KYS için çok-seviyeli hiyerarşik bir ÇES organizasyon yapısı hazırlanırken analiz aşamasında düşünülen rol ve protokollere yeni bir rol ve protokol eklenmesine ihtiyaç olmadığı anlaşılmıştır. Bunun başlıca sebebi sistem tasarımcılarının bu tip konferans sistemlerine aşina olmasından dolayı ihtiyaç duyulan rollerin analiz safhasında eksiksiz belirlenmiş olmasıdır. Ancak mimari tasarım safhasında bu rollerin daha belirgin hale getirilmesi ve rol modellerinin rol şemaları ile ortaya konulması gerçekleştirilmiştir.

Organizasyonel yapı başlangıçta düşünülen rollerin kesinleşmesini sağlamıştır. Bu roller; karar verici, bildiri yöneticisi, atayıcı, hakem ve toplayıcı rolleridir. Organizasyonel yapıda bulunmamasına rağmen yazar rolü sistemde beklenen en önemli rollerden birisidir. Mimari tasarım safhasında her rol için uygun rol şemaları hazırlanmıştır. Rollere ait canlılık, emniyet vb. özellikler bu şemalarda Bölüm 2.1'de tarif edilen notasyona göre hazırlanmıştır. Ancak yer kısıtlarından ötürü bildiride sadece hakem rolüne ait şema verilmiştir (Şekil 4).



Şekil 4: HAKEM rolü için rol şeması

4.2.3. Etkileşim Modeli:

KYS'de etmenlerin rollerine bağlı olarak gerçekleştirdikleri etkileşimler Tablo 3'te gösterilen rol modelinde etkileşim özellikleri ile beraber listelenmiştir. Yer kısıtlarından ötürü burada sadece HAKEM rolüne ait etkileşimler anlatılacaktır.

HAKEM rolü ATAYICI ve TOPLAYICI rolleri ile etkileşime girmektedir. Bu role ait iki ana sorumluluk bulunmaktadır. Bunlar atayıcının bildirileri hakemlere eşlemede kullanacağı hakem tercihlerini oluşturmak ve kendisine atanan bildirinin değerlendirmesini oluşturmaktır. Bildiri gönderim süresinin bitiminde ATAYICI hakemden tercihlerini göndermesi için *TercihTopla* protokolü üzerinden istekte bulunur. Bunun için, HAKEM rolü *TercihGönder* protokolü üzerinden göndereceği kaynağın adresini atayıcıya gönderir. Bildiri değerlendirme süresi bitiminde TOPLAYICI, hakemlerden değerlendirmelerini *DeğerlendirmeTopla* protokolü ile ister. Daha sonra HAKEM, *DeğerlendirmeGönder* protokolü ile hazırlamış olduğu değerlendirmenin adresini toplayıcıya gönderir. HAKEM rolü gönderdiği bu kaynakları oluşturmak için iki tane aktivite gerçekleştirir: *TercihOluştur* ve *DeğerlendirmeOluştur*.

Tablo 3: KYS etkileşim modeli

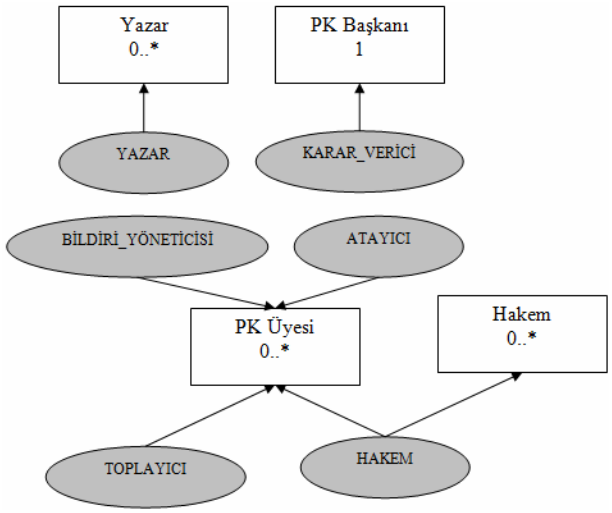
Protokol İsmi	**Başlatıcı**	**Eş**	**Girdiler**	**Çıktılar**	**Açıklama**
BildiriGönder	Yazar	BildiriYöneticisi	Bildiri bilgileri	Onay bilgileri	Bildirinin gönderimini sağlar.
GönderiyiOnayla	BildiriYöneticisi	Yazar	Yazar bilgileri	-	Bildiri gönderimini onaylar.
ÖzetTopla	Atayıcı	BildiriYöneticisi	Bildiri gönderim süresi bitimi	Özetlere ait URL adresleri	Bildirileri hakemlere atamak için özetleri ister.
ÖzetGönder	BildiriYöneticisi	Atayıcı	Atayıcı bilgileri	-	Atayıcıya düşen bildirilere ait özetleri gönderir.
TercihTopla	Atayıcı	Hakem	Bildiri gönderim süresi bitimi	Tercihlere ait URL adresleri	Bildirileri hakemlere atayabilmek için tercihleri ister.
TercihGönder	Hakem	Atayıcı	Atayıcı bilgileri	-	Hakem tercihlerini gönderir.
HakemAta	Atayıcı	Hakem	Tercih, özet	-	Bildirileri hakemlere atar.
DeğerlendirmeGönder	Hakem	Toplayıcı	Toplayıcı bilgileri	-	Bildirilere ait değerlendirmeleri gönderir.
DeğerlendirmeTopla	Toplayıcı	Hakem	Bildiri değerlendirme süresi bitimi	Değerlendirmelere ait URL adresleri	Hakem tarafından tamamlanan değerlendirmeleri ister.
DeğerlendirmeYolla	Toplayıcı	KararVerici	Değerlendirme toplama süresi bitimi	Değerlendirmelere ait URL adresleri	Hakemler tarafından tamamlanan değerlendirmeleri gönderir.
YazarBilgilendir	KararVerici	Yazar	Karar verme süresi bitimi	-	Bildirinin kabul veya reddedildiği bilgisini gönderir.

4.3. Detaylandırılmış Tasarım:

Bölüm 2.3'te anlatıldığı gibi Gaia ÇES geliştirme sürecine uygun olarak bu safhada etmen tabanlı KYS için etmen ve servis modelleri oluşturulmuştur.

4.3.1. Etmen Modeli:

Şekil 5'teki etmen modelinde görüldüğü gibi KYS'de 6 farklı rolü oynayacak 4 yazılım etmeni belirlenmiştir. YAZAR rolü doğal olarak *Yazar* etmenine verilmiştir. KARAR_VERİCİ rolü *PK Başkanı* etmeni tarafından, BİLDİRİ_YÖNETİCİSİ, ATAYICI ve TOPLAYICI rolleri ise *PK Üyesi* etmeni tarafından organizasyonel kurallar doğrultusunda oynanacaktır. Sistem program komitesi üyesi olmayan hakemlerin de hakemlik yapmasına izin verdiği için ayrı bir *Hakem* etmeni HAKEM rolünü oynamak üzere modelde yer almaktadır.



Şekil 5: Etmen modeli

4.3.2. Servis Modeli:

KYS'deki her rol için (YAZAR, ATAYICI, TOPLAYICI, vb.) bir servis modeli hazırlanmıştır. Her modelde ilgili servise ait girdi-çıktılar ve ön ve sonlanma koşulları belirtilmiştir. Yine yer kısıtlarından dolayı burada sadece HAKEM rolüne ait servis modelinden bahsedilmektedir.

Tablo 4'te HAKEM rolüne ait servis modeli görülmektedir. Bu role ait beş servis tanımlanmıştır. İlk iki servis *Tercih oluştur* ve *Değerlendirme oluştur* servisleridir. HAKEM rolünün aktiviteleri olan *TercihOluştur* ve *DeğerlendirmeOluştur*'dan türemiştir. *Tercih oluştur* servisi hakeme ait bilgiler ile hakem tercihlerini oluşturmak için kullanılır. Ön koşulu hakem bilgilerinin girilmiş olmasıdır. Sonlanma koşulu ise hakeme ait tercihlerin başarılı bir şekilde oluşturulmasıdır. *Değerlendirme oluştur* servisi hakeme atanan bildiriler için hakemin oluşturacağı değerlendirmelerin hazırlanmasını hedefler. Servisin girdisi bildiri bilgileri iken çıktısı ise bildiriye ait değerlendirme formudur. Servisin ön koşulu bildirinin HAKEM'e atanmış olmasıdır. *TercihGönder* protokolünden *Tercih gönder* servisi türemiştir. Bu servis ATAYICI ile etkileşim için kullanılır ve istendiğinde hakeme ait tercihler gönderilir. Servis zamana bağlı bir servistir ve bildiri gönderim süresinin bitiminde ATAYICI tarafından tetiklenir. Açıkça görüldüğü üzere, tercihlerin başarılı bir şekilde ATAYICI'ya gönderilmesi için bu servisten önce tercih bilgisinin girilmiş olması gerekmektedir. Benzer bir servis olan *Değerlendirme gönder* servisi, bildiri değerlendirme süresi bitiminde TOPLAYICI'ya değerlendirmeleri göndermek için geliştirilmiştir ve *DeğerlendirmeGönder* protokolünden türemiştir. Son servis ise *Tüm atanan bildirilerin değerlendirilmelerinin oluşturulduğundan emin ol* servisidir ve HAKEM'e atanan tüm bildirilerin sayısı kadar değerlendirme formunun oluşturulup oluşturulmadığının kontrolü için kullanılır.

Tablo 4: HAKEM rolü için servis modeli

Servis	**Girdi**	**Çıktı**	**Ön Koşul**	**Sonlanma Koşulu**
Tercih oluştur	Hakem bilgileri	`tercih`	`tercih` bilgisi girildi.	`tercih` başarılı bir şekilde oluşturuldu.
Değerlendirme oluştur	`bildiri` bilgileri	`değerlendirme`	`bildiri` hakeme atandı.	`değerlendirme` başarılı bir şekilde oluşturuldu.
Tercih gönder	-	`tercih`	Atayıcı tarafından `tercih` istendi.	Atayıcı `tercih`i aldı.
Değerlendirme gönder	-	`değerlendirme`	bildiri `değerlendirmesi` oluşturuldu.	Toplayıcı `değerlendirme`yi aldı.
Tüm atanan bildirilerin değerlendirmelerinin oluşturulduğundan emin ol	Atanan `bildiri` sayısı , `değerlendirme` sayısı	-	Doğru	Atanan `bildiri` sayısı = `değerlendirme` sayısı

5. Değerlendirme

Bu çalışmada kazandığımız deneyimler, bir kısmı literatürde de yer alan aşağıdaki gerçekler çerçevesinde şu şekilde özetlenebilir:

1. Gaia metodolojisi sistem gereksinimlerinin belirlenmesi ve gösterimi konularında herhangi bir destek içermemektedir. Gaia sistem gereksinimlerinin metodolojiye girdi olarak verileceğini varsayar. Hernekadar Gaia tasarımcıları geleneksel gereksinim mühendisliği yaklaşımlarının Gaia metodolojisine kolaylıkla eklenebileceği ve uyarlarlanabileceğini ifade etseler de [17], bu konuda bir çalışma söz konusu değildir. Çalışmamızda gereksinimlerin anlatımı için basit metinsel bir gösterim kullandık ve bu yaklaşımımızın analiz ve tasarım modellerinin oluşturulabilmesi için yeterli olduğunu gördük.

2. Hernekadar bu çalışmada tasarlanan sistemin gerçekleştirimi amaçlanmamış olsa da, Gaia'nin bu konu ile zaten ilgilenmediğini gözlemledik. Etmen sistemleri onlara gereksinim duyacakları çalışma ortamını sağlayacak platformlar üstünde var olabilirler. Bu platformlar etmen sistemi geliştiricilerine tasarımladıkları sistemin iş prensiplerini ilgilendirmeyen konular (etmenlerin nasıl haberleşecekleri, hangi etmenin hangi servisi verebileceğinin bilinmesi ve bulunması gibi) ile ilgilenmemelerini sağlar. Farklı etmen mimarileri hedeflenerek geliştirilmiş farklı etmen platformları mevcuttur (etmen davranışlarının Java sınıfları ile modellendiği JADE [1] ya da etmen kanı-istek-hedeflerinin Java sınıfları ve XML ile modellendiği JADEX [2] gibi). Etmen platformlarının heterojen yapıları Gaia'nın gerçekleştirim safhasını desteklememesini bir dereceye kadar haklı kılmaktadır. Böylelikle metodoloji herhangi bir etmen platformunda gerçekleştirilmesi planlanan ÇES'lerin tasarımı için kullanılabilir. Ancak, etmenlerin çalışma ortamının (etmen platformunun) tasarım modelleri üstüne kısıtlar getirebileceği ya da yeni ve farklı modellere gereksinim duyurabileceği ve tüm bunların gerçekleştirim sırasında kolaylıklar sağlayabileceği açıktır. Yakın gelecekte etmen platformlarının standartlaşması mümkün görülmemektedir. Bu neden ile Gaia gibi geliştirim safhasını desteklemeyen metodolojilerin farklı platformlar için farklı biçimlerde genişletilmesi gerekmektedir. Metodolojilere eklenecek geliştirim süreçlerinde tasarım modellerinden seçilen etmen platformunun bileşenlerine nasıl ulaşılacağı tanımlanmalıdır.

3. Gaia süreci herhangi bir tekrar içermemektedir. Süreç analizden tasarıma sıralı akar. Bu durum ile gerçek hayatta nadiren karşılaşılmaktadır. Çalışmamız sırasında süreç adımlarında geri dönümlere ve modellerimizi yeniden gözden geçirmeye gereksinim duyduk.

4. Gaia organizasyonel soyutlamalarının ÇES'lerin karakteristiklerine uygun, basit ve kolay kullanılabilir olduklarını ve Gaia metodoloji süreçleri içinde geliştirilen modeller ile uyum içinde olduklarını gözlemledik. Gaia ile tasarlanan bir ÇES'de etmenler iyi tanımlanmış rolleri bağımsız olarak oynarlar (rol soyutlama). Etmenler bu rollerin icrası sırasında birbirleri ile etkileşme gereksinimi duyarlar (etkileşim soyutlama). ÇES bir ortamda yerleşiktir (ortam soyutlama). Ek olarak organizayon güdümlü soyutlama etmen topluluklarının bir yapıları olacağını (yapısal soyutlama) ve bu yapının organizasyonel kuralları (organizasyon kuralları soyutlama) yerine getirebilmek için en iyi yapı olacağını öngörür. KYS'nin rol, etkileşim ve ortam soyutlamaları için, aynı adlar ile anılan rol, etkileşim ve ortam modellerini geliştirdik. KYS'de uyulması gerekli organizasyon kurallarını ve KYS için bu kuralların en iyi uygulanabileceğini düşündüğümüz bir organizasyonel yapı belirledik.

5. Gaia modelleri için herhangi bir gösterimi zorunlu kılmaz, tavsiyelerde bulunur. Bu yaklaşımın artıları ve eksileri olduğunu düşünüyoruz. Sistem geliştiricilerine gösterimde sağlanan mutlak özgürlük konunun en belirgin artısıdır. Örnek olarak biz organizasyonel kuralların gösterimi için önerildiği gibi zamansal maktık, emniyet özellikleri için daha rahat ifadeler, ve etmen modeli için grafik gösterim kullandık. Tüm bunları sadece metin kullanarak da yapabilirdik. Öte yandan, Gaia sürecini otomatikleştirecek araçların yapımı için formal gösterime ve standartlara gereksinim duyulduğu açıktır. Bugün için ne böyle standartlar, ne de araçlar mevcuttur.

6. Gaia rollerin ve servislerin etmenlere dinamik atanmalarına değiniyor olsa da, bu durumun modelleri nasıl etkileyeceğini açıklamamaktadır. Çalışmamızda rol ve servislerin statik atandıklarını var saydık.

6. İlgili Çalışmalar

Çeşitli ÇES metodolojileri kullanılarak etmen sistemlerinin geliştirilmesi ile ilgili literatürde çalışmalara rastlanmaktadır. Bu çalışmaların bir kısmında ÇES'lerine benzer organizasyonel hiyerarşik bir yapıda olması, sistem aktörlerinin etmenlerle birebir eşlenmesi, bu aktörlerin etmenlere ait karakteristik (özerk, karşıt-eylemli vb.) yapılara uygun aktivitelerinin bulunması ve akademisyenlerin devamlı kullandıkları bir sistem olması nedeniyle özellikle KYS tasarımı ele alınmıştır. Örneğin Morandi ve ark. [10]'da Tropos metodolojisinin ÇES'leri geliştirmek için sunmuş olduğu geliştirim aracını kullanarak bir KYS'nin geliştirme adımlarını örneklemişlerdir. Benzer bir çalışmada [11] ise Kanı-İstek-Hedef modeline uyan etmenlerin yer aldığı bir KYS'nin Prometheus metodolojisi kullanılarak gerçekleştirimi anlatılmıştır. Bir diğer çalışmada [12] çok-etmenli bir KYS'nin geliştirilmesi için organizasyon tabanlı ÇES mühendisliğine dayanan bir yaklaşım sunulmuş; etmen

[1] http://jade.tilab.com/, son erişim: Mayıs 2009
[2] http://jadex.informatik.uni-hamburg.de/, son erişim: Mayıs 2009

tasarımında amaç modellerinin (*"goal model"*) kullanılması üzerinde durulmuştur. Santos ve ark. ise [13]'te bizim çalışmamızda da çok önemli bir yer tutan organizasyonel kuralları temel alan bir etmen geliştirme sürecini bir KYS geliştiriminde kullanarak örneklemiştir. Zambonelli ve ark. ise [1]'de KYS'ni durum çalışması olarak seçmişler ve Gaia metodolojisini anlatmışlardır. Ancak ulusal boyutta böyle bir çalışmanın olmaması, uluslar arası çalışmalarda ise ÇES geliştirme deneyimlerinin ve metodoloji değerlendirmelerinin az olması bizi bu çalışmayı yapmaya yönlendirmiştir. Bu bildiride anlatılan Gaia temelli çok-etmenli KYS geliştirme çalışmasının hem Gaia metodolojisini değerlendirme hem de ETYM için bir geliştirme deneyimi sunması açısından yukarıdaki çalışmalarla birlikte alana katkı yaptığı düşünülmektedir.

7. Sonuç ve İleriye Yönelik Çalışmalar

Bu çalışmada organizasyon metaforunu kullanan bir ÇES tasarım metodolojisinin gerçek bir organizasyonun analiz ve tasarımında nasıl kullanıldığı örneklenmiştir. Gaia metodolojisinin ilgili safhaları sistem geliştirmede uygulanmış ve ÇES'lerin ortaya konması için bu sürecin uygulanmasına yönelik deneyimler aktarılmıştır.

Çalışmamızın devamında Gaia ile modellenen ÇES'lerin gerçekleştirilmesi için önerilen yaklaşımların incelenmesi ve bu çalışmada analizi ve tasarımını yaptığımız etmen tabanlı KYS'nin hayata geçirilmesi planlanmaktadır. Öncelikli olarak Moraitis ve Spanoudakis'in [21]'de tanıttığı, GAIA2JADE adı verilen ve Gaia ile modellenmiş bir ÇES'in JADE ÇES geliştirme çatısı ile gerçekleştirilmesine yönelik önerilen süreci uygulayarak bu çalışmada yer alan etmenlerin gerçekleştirimini planlıyoruz. Gerçekleştirilecek bu çalışmanın bir diğer amacı konu ile ilgili literatürde yeterli deneyim çalışmaları bulunmadığı için önerilen sürecin işlerliğini sınamaktır.

8. Kaynakça

[1] Zambonelli, F., Jennings, N.R. ve Wooldridge, M., "Developing multiagent systems: The Gaia methodology", *ACM Transactions on Software Engineering and Methodologies*, 12(3):317-370, 2003.

[2] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. ve Mylopoulos, J., "Tropos: An agent-oriented software development methodology", *Autonomous Agents and Multi-Agent Systems*, 8(3):203-236, 2004.

[3] Bernon, C., Gleizes, M-P., Peyruqueou, S. ve Picard , G., "ADELFE: A methodology for adaptive multi-agent systems engineering", *Lecture Notes in Artificial Intelligence*, 2577:70-81, 2003.

[4] DeLoach, S.A., Wood, M.F. ve Sparkman, C.H.,, "Multiagent systems engineering", *International Journal of Software Engineering and Knowledge Engineering*, 11(3):231-258, 2001.

[5] Padgham, L. ve Winikoff, M., "Prometheus: A methodology for developing intelligent agents", *Lecture Notes in Computer Science*, 2585:174-185, 2002.

[6] Pavon, J., Gomez, J. ve Fuentes, R., "The INGENIAS Methodology and Tools", (Kitap Bölümü) ss. 236-276, Agent-Oriented Methodologies, Idea Group Publishing, 2005.

[7] Iglesias, C.A., Garijo, M., Gonzalez, J.C., ve Velasco, J.R., "Analysis and design of multi-agent systems using MAS-CommonKADS", *Lecture Notes in Artificial Intelligence*, 1365:313-326, 1998.

[8] Cossentino M., "From Requirements to Code with the PASSI Methodology", (Kitap Bölümü) ss. 79-106, Agent-Oriented Methodologies, Idea Group Publishing, 2005.

[9] Henderson-Sellers, B. ve Giorgini, P., *Agent-Oriented Methodologies*, Idea Group Publishing, USA, 413p., 2005

[10] Morandini M., Nguyen D. C., Perini A., Siena A. ve Susi A., "Tool-Supported Development with Tropos: The Conference Management System Case Study", *Lecture Notes in Computer Science*, 4951:182–196, 2008.

[11] Padgham L., Thangarajah J. ve Winikoff M., "The Prometheus Design Tool – A Conference Management System Case Study", *Lecture Notes in Computer Science*, 4951:197–211, 2008.

[12] DeLoach S., "Developing a Multiagent Conference Management System Using the O-MaSE Process Framework", *Lecture Notes in Computer Science*, 4951:168–181, 2008.

[13] Santos D., Ribeiro M. R. ve Bastos R. M., "Developing a Conference Management System with the Multi-Agent Systems Unified Process: A Case Study", *Lecture Notes in Computer Science*, 4951:212–224, 2008.

[14] Durmus A., "*Dağıtık çoklu etmen toplantı planlama sistemi*", Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, 90s., 2002.

[15] Bozkur C. C., "*Güvenli etmen sistemi kullanarak e-borsa tasarımı ve gerçeklenmesi*", Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, 87s., 2007.

[16] Alaybeyoğlu, A., Kardaş, G., Erdur, R. C. ve Dikenelli, O., "SABPO Metodolojisi Kullanılarak FIPA Uyumlu Çok-Etmenli bir Otel Rezervasyon Sisteminin Tasarımı ve Gerçekleştirilmesi", Akademik Bilişim 2007, Kütahya, Türkiye, 2007.

[17] Zambonelli, F., Jennings, N.R. ve Wooldridge, M., "Multi-Agent Systems as Computational Organizations: The Gaia Methodology", (Kitap Bölümü) ss. 136-172, Agent-Oriented Methodologies, Idea Group Publishing, 2005.

[18] Zambonelli, F., Jennings, N.R. ve Wooldridge, M., "Organisational rules as an abstraction for the analysis and design of multi-agent systems", *International Journal of Software Engineering and Knowledge Engineering*, 11(3):303-328, 2001.

[19] Wooldridge, M., Jennings, N. R. ve Kinny, D. "The Gaia methodology for agent-oriented analysis and design", *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285-312, 2000.

[20] Simon, H. A., *Models of man: social and rational*, Wiley Publishing, New York, 287p., 1957.

[21] Moraitis, P. ve Spanoudakis, P., "The GAIA2JADE process for multi-agent systems development", *Applied Artificial Intelligence*, 19(8):251 – 273, 2006.