

Ontoloji Tabanlı Çok-etmenli Sistemlerin Model GÜdümlü Geliştirilmesi

Geylani Kardaş¹ Erdem Eser Ekinci² Bekir Afşar³ Oğuz Dikenelli³ N. Yasemin Topaloğlu³

¹Uluslararası Bilgisayar Enstitüsü, Ege Üniversitesi, İzmir

^{2,3}Bilgisayar Mühendisliği Bölümü, Ege Üniversitesi, İzmir

¹e-posta: geylani.kardas@ege.edu.tr

²e-posta: erdemeserekinci@gmail.com

³e-posta: {bekir.afsar, oguz.dikenelli, yasemin.topaloglu}@ege.edu.tr

Özetçe

Çok-etmenli Sistemler'in (ÇES) geliştirilmesinde mümkün olan en yüksek soyutlama seviyesinde çalışmak ÇES'lerin iç karmaşıklıklarından, dağıtık yapılarından ve açık bilgi modellerine (ontolojiler) dayalı geliştirilmelerinden dolayı gün geçtikçe önem kazanmaktadır. Yazılım geliştirme odağını koddan modellere çeviren ve farklı soyutlama seviyelerindeki modelleri kullanarak yazılım geliştirmedeki karmaşıklığı azaltmayı hedefleyen Model GÜdümlü Geliştirme (MGG) yaklaşımının ÇES yazılımlarının geliştirilmesinde de uygun bir alternatif olduğu görülmektedir. Bu durumdan hareketle bu bildiri ontoloji tabanlı ÇES'lerin model güdümlü geliştirilmesi için gereken sürece ait adımlar tanımlanmıştır ve uygulanacak yöntem üzerinde durulmuştur. ÇES sistemlerinin platforma özgü seviyede modellenmesi için gereken süreç adımı somutlaştırılarak yazılım araçları ile nasıl desteklendiği yine bu bildiri anlatılmıştır.

1. Giriş

Tasarım amaçlarını karşılamak amacıyla bir ortamda özerk ("autonomous") eylemlerde bulunma yeteneğine sahip sistemler olarak tanımlayabileceğimiz yazılım etmenlerinin ("software agents") tasarımı ve hayata geçirilmesi için günümüzde Etmen Tabanlı Yazılım Mühendisliği (ETYM) kapsamında birçok çalışma yürütülmektedir. Bu özerk, reaktif ve proaktif yapıdaki etmenlerin bir araya gelerek oluşturdukları Çok-etmenli Sistem'lerin (ÇES) geliştirilme altyapısını oluşturacak çeşitli iletişim dilleri, etkileşim protokolleri, etmen mimarileri ve etmen yazılım geliştirme metodolojileri ETYM araştırmacıları tarafından tanımlanmıştır. ÇES araştırma alanı uzun bir süreç sonunda dağıtık, karmaşık ve hedef yönelimli yazılımların geliştirimi için önemli bir teorik altyapı sağlamıştır. Ancak bu teorik altyapıya rağmen, ÇES uygulama altyapısının karmaşıklığı (dağıtık, açık, özerk vb.) bu sistemlerinin geliştirilmesi için yoğun bir bilgi kümesini gerektirmektedir. Buna dayalı olarak [1]'de de belirtildiği gibi ETYM'nin karakteristikleri şu anki yazılım metodolojilerinin sınırlarını zorlamaktadır.

İnternet ortamında dağıtık olarak çalışan ÇES'lerin bilgi paylaşımı problemi Anlamsal Web araştırma alanı ile kesişmektedir. Bu İkinci Jenerasyon Web, Dünya Geneli Ağ'ın (WWW), web sayfası içeriklerinin ontolojiler kullanılarak yorumlanabileceği bir seviyeye taşınmasını hedeflemektedir. Bu bağlamda ontoloji dil standartları oluşturulmuş ve bilgi modelleme amacıyla yoğun olarak kullanılmaya başlanmıştır. Etmenler arasındaki bilgi paylaşımı için de ontolojilerin kullanımı ÇES araştırmalarına yeni bir yön vermiştir [2]. Ontoloji kavramlarıyla ÇES kavramlarının bir arada kullanılmaya çalışılması ÇES geliştirilmesini daha da karmaşık hale getirmiştir.

Öte yandan yazılım gereksinmelerinin sürekli karmaşıklaşması, yazılım mühendisliği araştırmacılarını yeni yazılım geliştirme yaklaşımları, metodları ve teknikleri geliştirmeye zorlamaktadır. Bunlardan biri olan Model GÜdümlü Geliştirme (MGG) yazılım geliştirme odağını koddan modellere çevirmekte ve farklı soyutlama seviyelerindeki modelleri kullanarak yazılım geliştirmedeki karmaşıklığı azaltmayı hedeflemektedir.

ÇES'lerin geliştirilmesinde mümkün olan en yüksek soyutlama seviyesinde çalışmak kritik bir öneme sahiptir. Çünkü iç karmaşıklıklarından, dağıtık yapılarından ve açık bilgi modellerine (ontolojiler) dayalı geliştirilmelerinden dolayı ÇES'lerin kod seviyesi ayrıntılarını ortaya çıkarmak bu alanlarda uzmanlaşmamış geliştiriciler için neredeyse imkansızdır. Bu durumdan hareketle bu bildiri ontoloji tabanlı ÇES'lerin MGG yaklaşımı ile geliştirilmesine yönelik bir sürecin tanımlanması ve bunun hayata geçirilmesine yönelik devam eden çalışmamız tanıtılmakta ve çalışmada şu ana kadar elde edilen ürünler aktarılmaktadır.

Bu çalışmada tanıtılan ÇES'ler için özelleştirilmiş model tabanlı yazılım geliştirme süreci, ontolojileri ÇES mimarisinin temel yapıtaşları olarak ele almaktadır. Böylelikle görsel modelleme ile hazırlanan ÇES'ler Anlamsal Web yetenekli olacak, yazılım etmenleri kullanıcıları adına Web içeriğini farklı kaynaklardan elde edebilecek, bilgiyi işleyebilecek ve sonuçları değiş tokuş edebileceklerdir.

Bildirinin geriye kalan kısmı şu şekilde düzenlenmiştir. Bölüm 2'de genel MGG süreci çıktıları ile beraber tanıtılmaktadır. Bölüm 3'te ÇES'lerin geliştirilmesi için söz konusu sürecin çalışmamız kapsamında nasıl uygulandığı yer almaktadır. Bölüm 4'te çalışmanın şu ana kadar elde edilen ürünleri anlatılmıştır. Bölüm 5'te model güdümlü ÇES geliştirme ile ilgili diğer çalışmalar aktarılmıştır. Bölüm 6'da ise sonuç ve ileriye yönelik çalışmalar yer almaktadır.

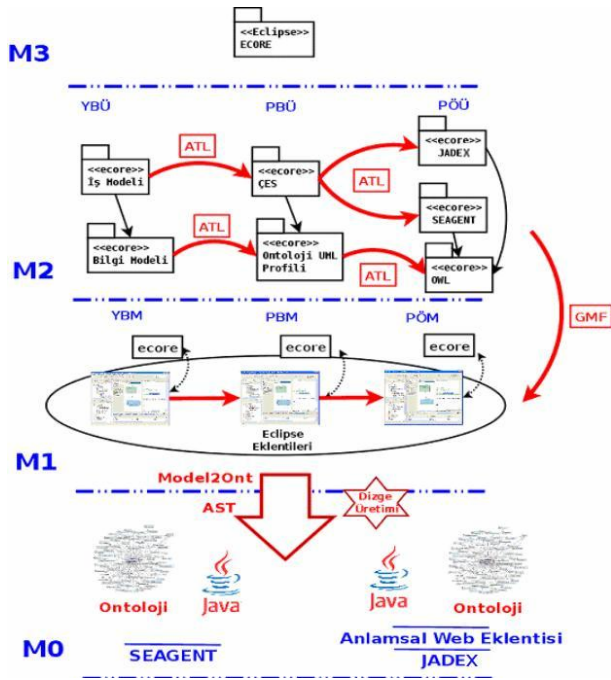
2. Model GÜdümlü ÇES Geliştirme

MGG süreci, bu süreci kullanan yazılım geliştiricileri ve bu sürecin yazılım mimarisini tanımlayan tasarımcılar açısından farklı farklı değerlendirilmelidir. Yazılım geliştiriciler açısından bakıldığında MGG'nin hedefi geliştiricileri teknolojik ayrıntılardan soyutlamak ve görsel olarak problemi ve çözümü tanımlayacakları bir ortam sunmaktır. MGG sürecini destekleyen yazılım mimarisinin ise bu üst düzey problem tanımını model dönüşümleri yoluyla farklı farklı platformlarda çalışabilir kod düzeyine kadar indirmesi gerekmektedir.

Model güdümlü ÇES geliştirme sürecimiz OMG'nin ("Object Management Group"), MGG yaklaşımının bir gerçekleştirimi olarak önerdiği Model GÜdümlü Mimari'ye

(MGM) dayanmaktadır. MGM¹ ile yazılım modellerinin çalıştırılabilir bileşenlere ve uygulamalara dönüştürülerek yazılım sistemlerinin geliştirilmesi amaçlanmaktadır. Bu nedenle MGM, MOF² ("Meta-Object Facility") çerçevesine dayanan birçok model dönüşümünü tanımlamaktadır.

Şekil 1'de MGM'nin dört katmanlı temel yapısının bizim çalışmamızdaki izdüşümü görülmektedir. Katmanlı mimarinin en üst seviyesinde (M3) üst-üstmodel ("meta-metamodel") yer almaktadır. Üst-üst-model, üstmodellerin ("metamodel") tanımlanması için bir zemin oluşturmaktadır. MOF standardı üst-üstmodeldeki yapıları tanımlamaktadır. Mimarinin M2 seviyesinde M3'e dayalı olarak tanımlanan üstmodeller yer almaktadır. Bu seviyedeki üstmodeller model güdümlü geliştirim aracının uygulama alanına ait modellerin oluşturulabilmesi için şablon sağlamaktadır. M1 seviyesinde M2'de yer alan üstmodellerle tanımlanabilecek model örnekleri yer almaktadır. En alt seviye olan M0'da ise örnek modellerin çeşitli platformlar üzerinde çalışabilir kaynak kodları bulunmaktadır.



Şekil 1: MGM'ye dayalı ÇES yazılımı geliştirme süreci ve model dönüşüm altyapısı

MGM kapsamında model güdümlü geliştirme yapabilmek için öncelikle M2 seviyesindeki üstmodellerin tanımlanması gerekmektedir. MGM bu seviye için üç adet üstmodel tanımlamıştır: Yazılım Bağımsız Üstmodel ("Computation Independent Metamodel") (YBÜ), Platform Bağımsız Üstmodel ("Platform Independent Metamodel") (PBÜ) ve Platforma Özgü Üstmodel ("Platform Specific Metamodel") (PÖÜ).

Alan içerisinde yer alan varlıklar ve bu varlıklar arasındaki ilişkilerin programlama kavramları yer almadan modellenebilmesi için YBÜ oluşturulur. Bu soyutlama

seviyesindeki bir modelde ÇES'in uygulanacağı alanı tanımlayan kavramlar (örneğin rol, hedef gibi) kullanılabilir.

PBÜ'ler ise herhangi bir platformu temel almamalarına rağmen programlanabilir yapıları içerirler. Örneğin Anlamsal Web ortamında çalışan ÇES'ler için tanımlanan bir PBÜ, özel bir etmen çerçevesine ait olmayan etmen varlıklarını, ilişkilerini ve etmenlerin kullandığı ontolojileri tanımlamaktadır.

Öte yandan bir PÖÜ, gerçek bir platform uygulamasının kavramlarını barındırmaktadır. PBÜ'lerle tanımlanan varlıkların ve ilişkilerinin model dönüşümleri sonrasında farklı PÖÜ'lerle ifade edilmesi mümkün olur. Örneğin yukarıda değinilen Anlamsal Web ortamında çalışan ÇES PBÜ'nin varlıkları ve ilişkileri model dönüşümleri sonrasında gerçek etmen çerçevelerinin PÖÜ'lerine dayalı olarak ifade edilebilir. Böylece aynı ÇES modelinin farklı platformlarda hayata geçirilmesi mümkün olmaktadır.

M1 seviyesinde yukarıda sözü edilen üstmodellerle (YBÜ, PBÜ ve PÖÜ) uyumlu uygulama örnek modelleri yer almaktadır. Bunlar da sırasıyla Yazılım Bağımsız Model (YBM), Platform Bağımsız Model (PBM) ve Platforma Özgü Model (PÖM) adını almaktadır.

MGM'ye dayalı yazılım geliştirme sürecinde temel etkinlik model dönüşümleridir. Bir model dönüşümü, bir dizi modelin başka bir dizi modele veya kendilerine eşlenmesi olarak tarif edilebilir. Bu model dönüşümleri üstmodel seviyesinde (M2) tanımlanarak, model seviyesindeki (M1) örneklerin YBM'den kaynak koda kadar dönüşmesini sağlar.

Çalışmamız kapsamında Anlamsal Web ortamında çalışan ÇES'ler için birer YBÜ, PBÜ ve çeşitli etmen geliştirme çerçeveleri (SEAGENT³ ve JADEX⁴) için PÖÜ'lerin hazırlanması yer almaktadır. Söz konusu bu üstmodelleri kullanan modelden modele ve modelden koda dönüşüm adımları da önerilen MGG süreci için tanımlanmakta ve uygulamaya geçirilmektedir.

ÇES geliştiricilerinin yukarıda sözü edilen modelleri oluşturup güncelleyebilmeleri için MGM tabanlı görsel geliştirim araçlarına ihtiyaç vardır. Çalışmamızın temel hedeflerinden biri de bu ihtiyaca cevap verecek yazılım araçlarının üretilmesidir. Üretilen araçlar ile bir ÇES geliştiricisi etmen geliştirme teknolojilerinden bağımsız olarak tanımladığı YBM'i, MGG sürecine girdi olarak verecektir. Daha sonra model dönüşümü sonucunda oluşan PBM ve PÖM modellerinin güncellenmesi ve genişletilmesi yine araç desteğiyle sağlanacaktır.

Aşağıdaki alt bölümlerde çalışmanın ÇES'lerin yazılım bağımsız ve platform bağımsız modellemesi açısından kapsamı detaylandırılmıştır. Platforma özgü seviyede çalışmanın somut çıktıları hazır olduğundan bildiriye ayrı bir bölümde (Bölüm 4) detaylandırılmıştır.

2.1. Model Güdümlü ÇES Geliştirmede Kullanılacak bir YBÜ:

Etmen sistemleri için tanımlanan MGG yaklaşımlarında PBÜ'den tamamen soyutlanmış bağımsız bir YBÜ ortaya konulmamıştır. Bu çalışmalarda (örneğin [3, 4, 5 ve 6]) alan kavramları ve ÇES'e özgü kavramlar PBÜ içinde birlikte kullanılmaktadır. Bu durum zaten oldukça büyük olan ÇES

¹ <http://www.omg.org/mda>, son erişim: Mayıs 2009

² <http://www.omg.org/docs/ad/97-08-14.pdf>, son erişim: Mayıs 2009

³ <http://seagent.ege.edu.tr/>, son erişim: Mayıs 2009

⁴ <http://jadex.informatik.uni-hamburg.de/>, son erişim: Mayıs 2009

üstmodelinin içine, alan üstmodelinden gelen kavramların da eklenmesiyle anlaşılması zor bir PBÜ oluşturmaktadır.

Literatürde yayınlanmış önemli ÇES geliştirme metodolojilerinde [1] organizasyon, rol, hedef, davranış ve roller arası etkileşim gibi ÇES alanından bağımsız düşünülebilecek kavramların tanımlandığı gözlenmektedir. Bu kavramlar bazı ÇES metodolojilerinde gereksinimlerin modellenmesi için kullanılmakta ve daha sonraki etkinliklerde ÇES alanına özgü kavramlarla eşleştirilmektedir. Dolayısıyla bağımsız bir YBÜ tanımlanması için teorik altyapı, etmen temelli yazılım geliştirme araştırma alanında bulunmaktadır. Fakat bu çalışmaları toparlayan genel amaçlı bir YBÜ henüz tanımlanmamıştır. Bizim çalışmamızda genel amaçlı YBÜ oluşturulurken, sistem bilgi modellerinin (alan kavramları ve kuralları) YBÜ seviyesinde tanımlanması hedeflenmektedir. Söz konusu YBÜ, etmen araştırma alanında gereksinim seviyesinde tanımlanmış kavramları derleyecek ve bu kavramlarla bilgi modelleri arasındaki ilişkileri tanımlayacaktır.

2.2. Anlamsal Web Ortamında Çalışacak ÇES'ler için bir PBÜ:

Şekil 1'de de görüldüğü gibi YBÜ'in ve bilgi modelinin PBÜ'ye dönüştürülmesi gerekmektedir. Bu bağlamda, yürüttüğümüz çalışmada ÇES'lerin uygulama ortamlarından bağımsız modellerinin hazırlanmasını sağlayacak bir PBÜ'in ve bir ontoloji üstmodelinin oluşturulması hedeflenmektedir.

Aslında Anlamsal Web ortamında çalışan etmen sistemlerini platform bağımsız olarak modellemek amacıyla bir etmen üstmodeli [7] elimizde bulunmaktadır. Üstmodel, IEEE FIPA (*"Foundation of Intelligent Physical Agents"*) Modelleme Teknik Komitesi'nin ACSM¹ (*"Agent Class Superstructure Metamodel"*) adı verilen üstmodelini genişletmekte ve ayrıca UML (*"Unified Modeling Language"*) 2.0'a dayalı olarak Anlamsal Web desteğini sağlayacak yeni yapıları ve ilişkileri tanımlamaktadır. Bu üstmodel içerisinde 43 adet ÇES üstvarlığı (*"meta-entity"*) ve bunların arasındaki 75'ten fazla varlık ilişkisi modellenmiştir. Söz konusu bu üstmodelin model dönüşümlerinde kullanılması ve değişik ÇES geliştirme çerçeveleri için platforma özgü etmen modellerinin ve yazılım kodlarının otomatik dönüşümler sonrası elde edilmesi yine [7]'de yer almaktadır. Ancak geliştirilen bu PBÜ de YBÜ'de yer alması gereken kavramları içermektedir. Bu üstmodelin kullanımı sonucunda elde ettiğimiz deneyim, gereksinim kavramlarının YBÜ seviyesine çıkarılması gerektiği konusundaki yaklaşımımızı doğrulamıştır. Bu bağlamda, [7]'deki PBÜ'in yeniden mühendisliğinin gerçekleştirilerek öncelikle YBÜ kavramlarının üstmodelden ayrıştırılması gerekmektedir. Ayrıca PBÜ kavramları ile ontoloji kavramları arasındaki ilişkilerin de tanımlanması gerekmektedir.

3. Model Güdümlü ÇES Geliştirme Yöntemi

Ontoloji tabanlı ÇES'lerin model güdümlü olarak geliştirilmesi için aşağıda listelenen ihtiyaçların karşılanması gerekmektedir:

- ÇES geliştiricilerinin, sadece problem içerisinde yer alan varlıkları, bu varlıklar arasındaki ilişkileri ve varlıklara bağlı bilgileri (ontoloji)

¹ <http://www.omg.org/docs/agent/04-12-02.pdf>, son erişim: Mayıs, 2009.

modelleyebilecekleri genel bir YBÜ'in tanımlanması,

- Özellikle ontolojilerin ve etmen etkileşimlerinin de göz önünde bulundurulduğu bir ÇES PBÜ'nin tanımlanması,
- Hazırlanan YBÜ ve ÇES PBÜ arasında model dönüşümlerinin tanımlanması ve uygulanması,
- Literatürde varolan ve ÇES geliştiriciler tarafından kullanılan en az iki önemli etmen yazılım geliştirme çerçevesine ait PÖÜ'lerin çıkarılması,
- Hazırlanan PBÜ ve PÖÜ'ler arasında da model dönüşümlerinin tanımlanması ve uygulanması,
- Yukarıda sözü edilen iki etmen yazılım geliştirme çerçevesi için şablon seviyesinde yazılım kodlarının bu çerçevelere ait PÖÜ'lerin kullanıldığı modelden koda dönüşüm süreçleri sonrasında otomatik olarak elde edilmesi,
- Yukarıdaki yer alan tüm modelden modele ve modelden koda dönüşümler sırasında yazılım geliştiricilerin ihtiyaç duyacağı yazılım araçlarının tasarımı ve gerçekleştirimi.

Söz konusu ihtiyaçların karşılanması için çalışmamızda izlediğimiz yöntemi, dayandığı teknik altyapı ve uygulanan süreci ayrı ayrı inceleyerek burada aktarmak yararlı olacaktır. Teknik altyapı başlıklı aşağıdaki ilk alt bölümde, Şekil 1'de resmedilen mimari elemanlarının gerçekleştirimi sırasında kullanılan teknolojiler tanıtılırken, süreç başlıklı bölümde ise mimarinin gerçekleştirimi fazlara ayrılarak nasıl bir yol izlendiği anlatılmaktadır.

3.1. Teknik Altyapı

Çalışmamızda model güdümlü ÇES geliştirme sürecinin uygulanmasına yardımcı olacak yazılım araçları Java programlama dili kullanılarak Eclipse² yazılım geliştirme ortamında yapılmaktadır.

OMG'nin MOF standardını destekleyerek üstmodel tanımlamayı sağlayan ve Eclipse'in temel eklentileri arasında yer alan EMF³ (*"Eclipse Modeling Framework"*) kullanılarak çalışmamızda ÇES mimarisi için tasarlanan YBÜ, PBÜ ve PÖÜ'ler tanımlanmıştır. Öte yandan kaynak modellerimizden hedef modellere dönüşümlerin tanımlanmasını için ATL⁴ (*"ATLAS Transformation Language"*) ve buna bağlı Eclipse araç kümesinden faydalanılmıştır.

YBM, PBM ve PÖM'lerin kullanıcılar tarafından görsel olarak geliştirilebilmesi için sunduğumuz yazılım araçları GMF⁵ (*"Graphical Modeling Framework"*) kullanılarak hazırlanmıştır.

Model seviyesinden (M1) programlama seviyesine (M0) dönüşüm gerçekleştirilerek SEAGENT ve JADDEX kaynak kodlarının oluşturulması sistem mimarimizin en son parçasıdır. Modelden program üretmek için AST (*"Abstract Syntax Tree"*) kütüphanesi kullanılmıştır. Bu kütüphane, Java kodunun sözdizim ağacı ile modellenip üretilmesini sağlar. SEAGENT ve JADDEX ÇES platformları Java dilini destekliyor olsalar da ihtiyaç duydukları çalıştırılabilir yazılım bileşenleri her zaman için Java kodları değildir.

² <http://www.eclipse.org/>, son erişim: Mayıs 2009

³ <http://www.eclipse.org/modeling/emf/>, son erişim: Mayıs 2009

⁴ <http://www.eclipse.org/m2m/at/>, son erişim: Mayıs 2009

⁵ <http://www.eclipse.org/gmf/>, son erişim: Mayıs 2009

Örneğin SEAGENT etmen planlamada OWL ontolojilerini kullanmakta; JADEX'de ise etmen tanımları XML'e dayalı dosyalarda saklanmaktadır. Bu tip ontoloji ve XML dosyalarının hazırlanması için de modelden metinlere dönüşümleri sağlayan MOFScript¹ dili kullanılmıştır.

3.2. Süreç

Çalışmanın gerçekleştirim süreci *Platforma Özgü Ortam Üretimi*, *Platform Bağımsız Ortam Üretimi* ve *Yazılım Bağımsız Ortam Üretimi* olarak adlandırılan 3 faza ayrılmıştır. Her fazda MGM'nin YBÜ, PBÜ ve PÖÜ seviyelerinden birine odaklanılmaktadır. Ardışık olarak gerçekleştirilen fazlar, bir önceki fazdaki üstmodelleri ve araçları girdi olarak almaktadır. Fazların her birinde temel iki iş parçası gerçekleştirilmektedir. Bunlardan ilki, fazın odaklandığı MGG seviyesinde yer alan üstmodellerin EMF ile tanımlanıp bu üstmodelleri temel alan modellerin görsel geliştirimini sağlayan araçların GMF kullanılarak üretilmesidir. Diğeri ise odaklanılan seviyenin bir üstünde olan seviyeye geçişi sağlamak için uygun model dönüşüm kurallarının ATL ile yazılması ve uygulamaya geçirilmesidir. Fazların bitiminde modellerin ve ortamın tasarımını değerlendirebilmek için durum çalışmaları gerçekleştirilmektedir. Aşağıdaki alt bölümlerde bu fazlarda yer alan çalışmalar açıklanmaktadır.

3.2.1. Platforma Özgü Ortam Üretimi

Gerçekleştirimin ilk adımı olan Platforma Özgü Ortam Üretimi fazı, SEAGENT ve JADEX yazılım mimarilerine dayanan üstmodellerin oluşturulmasını ve modelden kaynak koda dönüşüm kurallarının tanımlanmasını amaçlamaktadır. Bu üstmodellere ek olarak OWL dilinin üstmodeli de EMF ile tanımlanmaktadır. SEAGENT, JADEX ve OWL üstmodellerinin çözümlenip oluşturulmasından sonra görsel model geliştirim araçları GMF'nin yardımıyla üretilmektedir. Daha sonra üretilen bu görsel geliştirim araçlarında kullanıcılar tarafından geliştirilen örnek modellerden kaynak kodların oluşturulmasını sağlayan yazılım birimi geliştirilmektedir. Bu yazılım birimi örnek modelleri çözümlüyüp AST sözdizim ağacını oluşturmakta ve bu ağacı yine AST kullanarak Java koduna dönüştürmektedir. Ayrıca görsel olarak tanımlanan OWL modellerinden ve diğer PÖM'lerden (SEAGENT ve JADEX) ontoloji üretimi bu faz kapsamında ele alınıp modelden ontolojiye dönüşüm yazılım birimi gerçekleştirilmektedir.

3.2.2. Platform Bağımsız Ortam Üretimi

Bu faz kapsamında ÇES'lerin geneline hitap etmesi amaçlanan bir ÇES üstmodelinin EMF altyapısıyla oluşturulması yer almaktadır. Bu ÇES üstmodeline dayanan ÇES modellerinin görsel olarak hazırlanmasını sağlayacak GMF tabanlı bir yazılım aracının da geliştirilmesi bu faz kapsamında yer alan çalışmalardan biridir. ÇES üstmodelinden SEAGENT ve JADEX üstmodellerine dönüşüm kurallarının ATL dili kullanılarak yazılması bu fazda yer alan bir diğer çalışmadır. Dönüşüm kuralları

sayesinde ÇES modelinden SEAGENT veya JADEX PÖM'leri elde edilebilecektir.

Bu fazda oluşturulacak diğer bir üstmodel, geliştiricilerin yazılımlarında kullanılacak ontolojileri platform bağımsız seviyede ve ayrıntılı olarak tanımlamasına olanak sağlayacak bir ontoloji üstmodelidir. Bir sonraki çalışma ise bu üstmodel ile PÖÜ seviyesindeki OWL üstmodelleri arasındaki dönüşümlerin ATL ile tanımlanarak PBM ile PÖM arasındaki geçiş sağlamayı hedeflemektedir.

3.2.3. Yazılım Bağımsız Ortam Üretimi

Yazılım Bağımsız Ortam Üretimi, sürecin son fazıdır. Bu fazda yazılımın üretileceği alanın yazılım teknolojilerinden bağımsız olarak modellenmesini sağlayacak görsel modelleme araçları üretilmektedir. Bu bağlamda öncelikle YBÜ ve bilgi üstmodeli tanımlanır. Bu üstmodeller kullanılarak görsel geliştirim araçları GMF yardımıyla ortaya çıkarılır. YBÜ görsel geliştirim aracından ÇES modeli görsel geliştirim aracına ve bilgi modelinden ontolojileri görselleştiren araca model dönüşümlerinin sağlanması bu fazın diğer önemli etkinliğidir. Dönüşümler önceki fazlara benzer şekilde ATL dili kullanılarak gerçekleştirilir.

4. ÇES'lerin Platforma Özgü Tasarımları için Modelleme Araçları

Ontoloji tabanlı ÇES'lerin model güdümlü geliştirilmesini hedefleyen çalışma kapsamında şu ana kadar Bölüm 3.2.1'de anlatılan üstmodelleme ve yazılım aracı geliştirme faaliyetleri tamamlanmıştır. ÇES'lerin platforma özgü modellenmesini desteklemek için SEAGENT ve JADEX ÇES yazılım geliştirme ortamlarına ait üstmodellerin oluşturulması, bu üstmodellere dayalı ÇES modellerinin yazılım geliştiriciler tarafından görsel olarak oluşturulup kaynak kodlarının elde edilebildiği yazılım araçlarının üretilmesi gerçekleştirilmiştir. Aşağıdaki alt bölümlerde söz konusu bu çalışmalar detaylandırılmıştır.

4.1. JADEX Platformuna Özgü Ortam Üretimi

JADEX [8] hedef yönelimli etmenlerin, Kanı-İstek-Hedef ("Belief-Desire-Intention") (KİH) modeline uygun olarak oluşturulmasına izin veren bir yazılım çatısıdır. Bir KİH mimarisinde [9] etmenler hangi amaçlara ne şekilde ulaşmak istediklerine karar vermektedirler. *Kanılar* etmenlerin çevreleri hakkında ellerinde bulunan bilgileri temsil ederken *Hedefler* adlarından da anlaşılacağı üzere etmenlerin yerine getirilmesini arzuladıkları herşeyi temsil eder. *İstekler* ise etmenlerin bilinçli tutumları ("deliberative attitude") olup hedeflerine ulaşmak için etmenlerin plan yapmalarını içerir.

Etmen yazılımı geliştirme açısından bakıldığında JADEX etmenlerinin geliştirilmesinin melez bir yaklaşıma dayandığı görülmektedir. Bu yaklaşım statik etmen özelliklerinin tanımlanmasını ve işletilebilir etmen planlarının programlanmasını içermektedir. Statik etmen özellikleri Etmen Tanımlama Dosyası ("Agent Declaration File") (ETD) adı verilen dosyalarda verilmektedir. Bir ETD dosyası XML kullanılarak yazılır ve ilgili etmenin KİH modelini belirtir. Öte yandan etmen planları işletilebilir bileşenlerdir ve Java program dosyalarında verilirler.

JADEX etmenlerinin model güdümlü geliştirilmesi için çalışmalarımız sonucunda JADEX KİH etmenlerini görsel olarak modelleyecek ve bu etmenlere ait ETD ve plan

¹ <http://www.eclipse.org/gmt/mofscript/>, son erişim: Mayıs 2009

dosyalarının modelden koda dönüşümler sonucunda otomatik olarak üretilmesini sağlayacak yazılım araçları hazırlanmıştır. Özellikle ETD dosyalarının hazırlanması oldukça güç olduğundan söz konusu yazılım araçlarının etmen yazılımı geliştiricileri için oldukça faydalı olacağına inanılmaktadır.

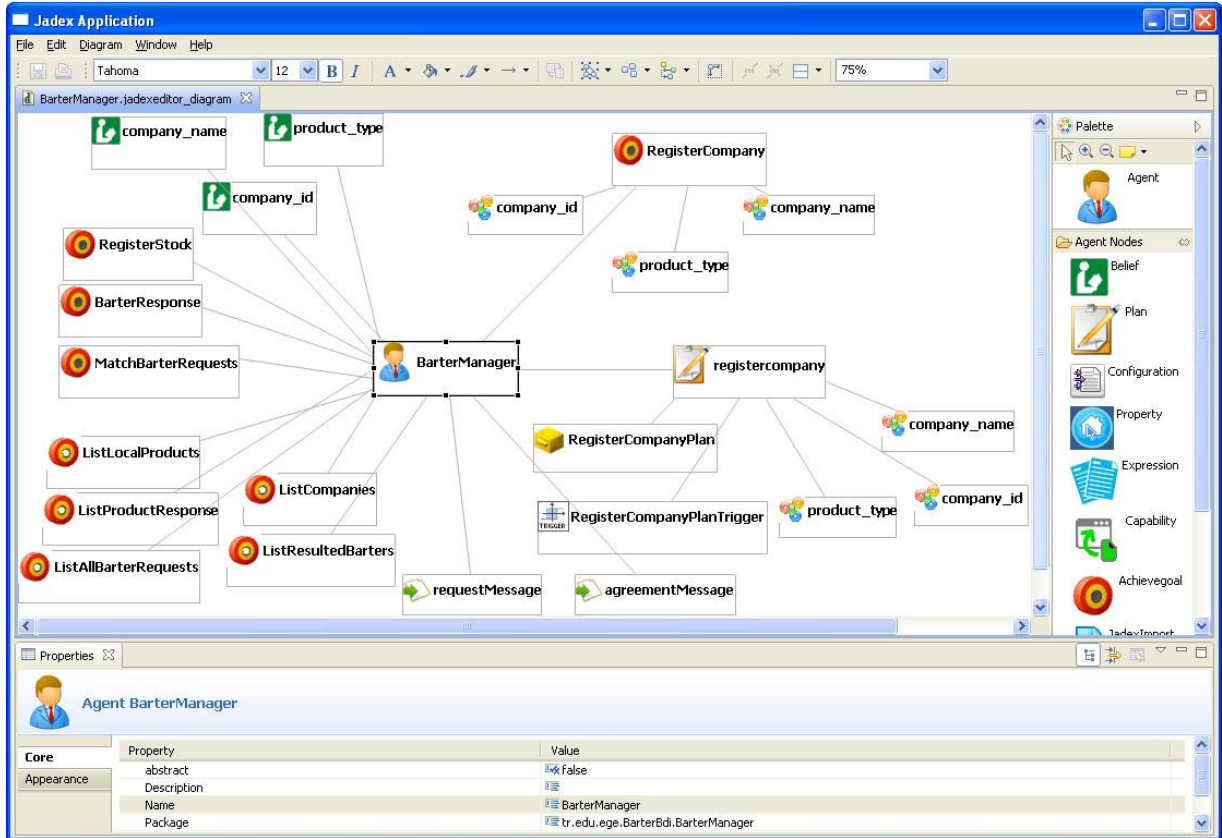
Görsel modelleme aracı JADEX üstmodeline dayalı olarak KİH etmenlerinin modellenmesini sağlamaktadır. JADEX üstmodelinin EMF'ye dayalı olarak hazırlanması ve Ecere karşılığının elde edilmesi yerine getirilen ilk çalışma olmuştur. Daha sonra bu üstmodelle dayalı olarak yazılım geliştiricilerinin JADEX sistem modellerini hazırlayabilecekleri görsel bir editör Bölüm 3.1 ve 3.2.1'de anlatılan tasarım prensiplerine uygun bir şekilde GMF kullanılarak oluşturulmuştur.

Etmen yazılımı geliştiricileri etmen sistemlerini geliştirmek için hem JADEX hem de SEAGENT için hazırlanan editörleri kullanabilir. Bu editörlerde etmen iş alanına ait elemanlar ve bunların ilişkilerine ait linkler editörlerin sağında kalan paletlerde yer almaktadırlar. Geliştiriciler palet üzerinden arzu ettikleri model elemanlarını seçmekte; aralarındaki bağları oluşturmakta ve görsel olarak sistem modellerini ortaya koymaktadırlar. Hazırlanan editör ortamları aynı zamanda modellerin bütünlüğünü ve yanlış model elemanı ilişkilerinin kurulmasının önlenmesini sağlamaktadırlar. Modellenen etmen sistemleri yine Ecere ile kodlanmış model dosyalarında saklanmaktadır. Şekil 2'de görüldüğü gibi JADEX editör ortamı etmenlerin KİH

modelinin görselleştirilmesini sağlamakta ve JADEX etmenlerinin kolay ve verimli bir şekilde geliştirilmesini sağlamaktadır. Aynı zamanda KİH elemanları arasındaki ilişkilerin doğru kurulmasını da sağlamaktadır. Örneğin bir Plan Gövde ("Body") elemanı JADEX üstmodelinin KİH anlamsalına göre ("semantics") sadece bir Plan düğümü ("node") ile ilişkilendirilebilir. Bu nedenle editör Gövde elemanlarının Plan düğümleri haricinde başka bir KİH elemanı ile ilişkilendirilmesine izin vermemektedir.

Şekil 2'de Takas Yöneticisi ("Barter Manager") adlı bir yazılım etmenine ait KİH modeli editör aracılığı ile oluşturulmuştur. Müşterilerinden aldığı kayıt bilgilerine göre işlem yapan bu etmenin elde ettiği ortam bilgileri etmenin başlangıç kanıları olarak Şekil 2'de sol üst kısımda verilmiştir. Etmenin amaçları (Şekil 2'de sol üstte), plan yapısı (Şekil 2'de sağda) ve birbirleri ile olan ilişkileri de yine örnek modelde yer almaktadır.

JADEX etmenlerinin görsel modellenmesinden sonra yazılım geliştiriciler entegre kod üreticisi kullanarak ETD ve etmen Java plan sınıflarına ait şablon kodları elde edebilirler. JADEX görsel geliştirim editöründe hazırlanan KİH modelleri Ecere dosyalarında saklanmaktadır. Bu dosyaları girdi olarak alan kod üretim modülü model dosyaları üzerinde MOFScript ile yazılan model-metin dönüşümlerini uygulayarak ilgili ETD ve Java dosyalarını kullanıcılara sunmaktadır. Örneğin Şekil 3'te Takas Yöneticisi etmenine ait otomatik üretilen ETD dosyasından bir parça verilmiştir.



Şekil 2: JADEX KİH etmenlerinin modellenmesi

```

<agent xmlns="http://jadex.sourceforge.net/jadex" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jadex.sourceforge.net/jadex-2.0.xsd name="BarterManager" package="tr.edu.ege.BarterBdi.BarterManager" description="" propertyfile="jadex.config">
  <imports></imports>
  <beliefs>
    <belief name="product_type" class="Product" exported="false" description="" updatarate="0" transient="false" dynamic="false">
      <fact>"null"</fact>
    </belief>
    <belief name="company_name" class="Company" exported="false" description="" updatarate="0" transient="false" dynamic="false">
    </belief>
    <belief name="company_id" class="Company" exported="false" description="" updatarate="0" transient="false" dynamic="false">
    </belief>
  </beliefs>
  <goals>
    <achievegoal name="RegisterCompany" recur="false" description="" exclude="when_tried" exported="false" posttoall="false" randomselection="false" rec
    <parameter name="company_name" class="Company" description="" direction="in" dynamic="false" exported="false" optional="false" transient="f
    <value>${beliefbase.company_name}</value>
    </parameter>
    <parameter name="company_id" class="Company" description="" direction="in" dynamic="false" exported="false" optional="false" transient="false
    <parameter name="product_type" class="" description="" direction="in" dynamic="false" exported="false" optional="false" transient="false" update
    <creationcondition>
    <contextcondition>
    <dropcondition>
    <recurcondition>
    <targetcondition>
    <failurecondition>
    </achievegoal>
    <achievegoal name="RegisterStock" recur="false" description="" exclude="when_tried" exported="false" posttoall="false" randomselection="false" rec
    <achievegoal name="MatchBarterRequests" recur="false" description="" exclude="when_tried" exported="false" posttoall="false" randomselection="f
    <querygoal name="ListProductResponse" recur="false" description="" exclude="when_tried" exported="false" posttoall="false" randomselection="fals
  </goals>
  <plans>
    <plan name="registercompany" description="" exported="false" priority="0">
    <parameter name="company_name" class="Company" description="" direction="in" dynamic="false" exported="false" optional="false" transient="f
    <value></value>
    <goalmapping ref="RegisterCompany.company_name"/>
    </parameter>
    <parameter name="company_id" class="Company" description="" direction="in" dynamic="false" exported="false" optional="false" transient="false
    <parameter name="product_type" class="Product" description="" direction="in" dynamic="false" exported="false" optional="false" transient="false"
    <body name="registerCompany" description="" class="RegisterCompanyPlan" inline="false" type="" language="">
    <trigger name="RegisterCompanyPlanTrigger" description="">
    <goal ref="RegisterCompany"/>
    </trigger>
  </plan>

```

Şekil 3: Bir JADEX KİH etmeni için otomatik üretilmiş ETD dosyasından bir parça

4.2. SEAGENT Platformuna Özgü Ortam Üretimi

SEAGENT [10] Anlamsal Web yetenekli ÇES'lerin etkileşimli ve test yönelimli bir biçimde geliştirilebilmesini sağlayan amaç tabanlı ("goal-oriented") bir etmen geliştirme platformudur. SEAGENT etmenleri tüm içsel bilgilerini amaç ontolojisi, plan ontolojisi, vb. OWL ontolojilerinde saklamaktadır. SEAGENT platformu anlamsal web servislerinin etmenler tarafından çalıştırılabilmesi için bir etmen planlayıcısı sunmaktadır. Dış servislere ait OWL-S¹ ontolojileri etmenler içerisinde saklanmakta; etmen planlayıcıları da servisleri ait OWL-S tanımlarını kullanarak anlamsal web servislerinin etmenler tarafından çalıştırılmasını sağlamaktadır.

SEAGENT ÇES'lerinin geliştirilmesi 4 temel modele dayandırılmıştır. Bunlar *Organizasyon*, *Plan*, *Protokol* ve *İş Alanı* modelleridir. Organizasyon Modeli'nde organizasyon ve ona ait roller, hedefler, rollerin hedefler üzerindeki sorumlulukları, etmenler, etmenlerin uzlaştığı roller gibi temel kavramlar ve bu kavramlara ait özellikler tanımlanmıştır. Öte yandan Plan Modeli SEAGENT etmenlerinin Hiyerarşik Görev Ağı ("Hierarchical Task Network") (HGA) [11] paradigmasına dayanan içsel etmen planlarının nasıl olacağını modellemektedir. Protokol Modelleri SEAGENT etmenleri arasındaki etkileşimlere ait protokolleri içermektedir. Protokol tanımları protokol, mesaj, argüman, sıra, seçim gibi protokol etkileşim kavramlarının ve bunlar arasındaki ilişkilerin saklandığı protokol ontolojilerinde verilmektedir. Son olarak İş Alanı Modeli ilgili ÇES'in iş alanı kavramlarının ve bunların ilişkilerin modellenmesini sağlamaktadır.

Yukarıda sözü edilen modeller bir ÇES'in yazılım tasarımına farklı bakış açılarından yaklaşmayı sağlamaktadır. Bu modeller ve bunların birbirleri ile olan ilişkileri aslında bir SEAGENT ÇES'inin dayandığı üstmodeli oluşturmaktadır. SEAGENT platformuna özgü ortam üretimi çalışmalarının ilk adımı JADEX'dekine benzer bir biçimde öncelikle SEAGENT üstmodelinin oluşturulmasını gerektirmiştir. Daha sonra bu üstmodele dayalı olarak yazılım geliştiricilerin sistemlerini modelleyebilecekleri yine GMF'ye dayalı modelleme araçları hazırlanmıştır.

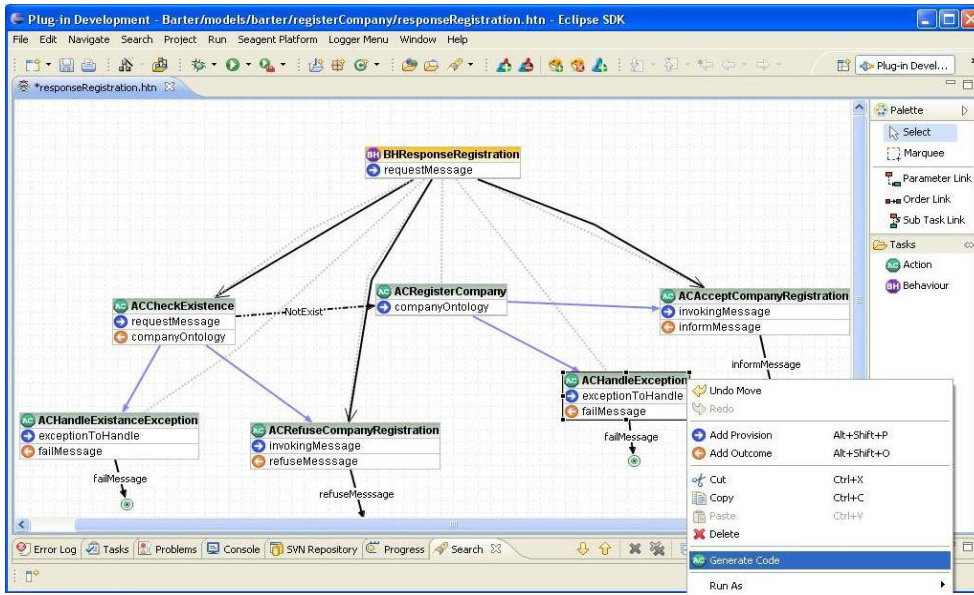
Hazırlanan modelleme araç seti yukarıdaki 4 temel modelin her biri için ayrı görsel editörler sunmaktadır. Örneğin Şekil 4 ve 5'te SEAGENT ÇES'lerinin planlama ve organizasyonel bakış açılarına göre modellenmesini sağlayacak editörlere ait ekran görüntüleri verilmiştir. Yine JADEX editörüne benzer şekilde bu editörlerde de her bakış açısına ait üstmodelde yer alan üst-varlıklar ile ilgili temsili şekiller editörlerdeki paletlerde yer almakta; yazılım geliştiricisi bu paleti kullanarak model elemanları ve ilişkilerini hazırlamakta ve uygun ara yüzü kullanarak arzu ettiği kaynak kodları elde etmektedir. Örneğin Şekil 4'te verilen ekran görüntüsünde "Takas Yöneticisi" adlı bir etmenin HGA'ya dayalı etmen plan yapısının nasıl modellendiği ve kod üretimi yer almaktadır. Plan hiyerarşisinde üstte yer alan davranışlar için SEAGENT plan ontolojileri üretilmekte; bu üst davranışları oluşturan eylemler için de uygun Java sınıflarına ait şablon yazılım kodları Bölüm 3.1'de anlatılan teknolojiler kullanılarak otomatik bir şekilde üretilmektedir. HGA'ya dayalı SEAGENT planlama ile ilgili detaylı bilgi [12]'de yer almaktadır. Şekil 6'da Şekil 4'teki etmen plan modelini üzerine model-metin dönüşümlerinin uygulanmasından sonra elde edilen şablon yazılım kodlarından örnekler verilmiştir.

¹ <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>, son erişim: Mayıs 2009

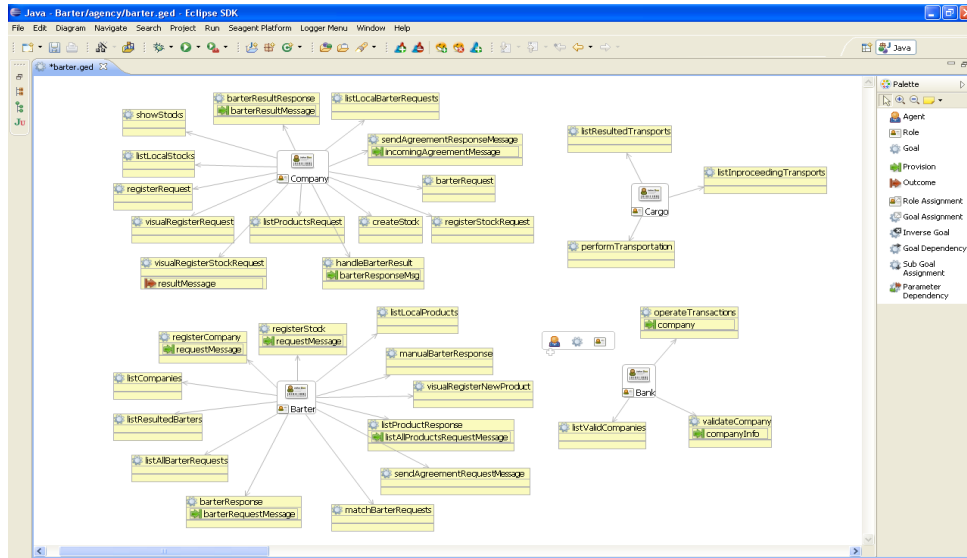
5. İlgili Çalışmalar

Model güdümlü yaklaşımların ÇES'lerin geliştirilmesinde uygulanması ETYM'nin belli başlı araştırma konularından biri haline gelmiştir. [3]'te tanıtılan "Cougaar MDA" etmen yazılım ürünlerini elde etmek amacıyla tasarım seviyesini, bireysel bileşenlerden alan seviyesi model tanımlarına yükselterek etmen sistemleri için üst seviye bir uygulama çerçevesi sunmaktadır. Amor ve ark. [13] etmen tabanlı tasarım ve MAS gerçekleştirimi arasındaki boşluğu doldurmak amacıyla MGM tabanlı bir süreç önermişlerdir. Model dönüşümlerinin etmen tabanlı sistem geliştirme amacıyla kullanılmasına yönelik başka bir çalışma [6]'da verilmiştir. MGM dönüşüm deseninin uygulandığı bu çalışmada belirli bir ÇES metodolojisine ait yapılar ile UML

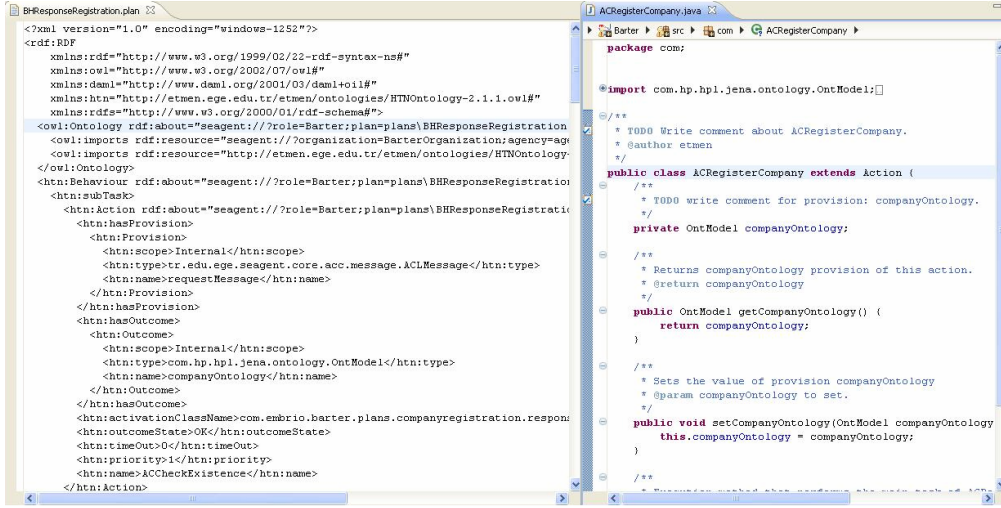
modelleri arasında dönüşümler hazırlanmış ve bu dönüşümler hayata geçirilmiştir. Hahn ve ark. ÇES'lerin modellenmesi için platform bağımsız bir üstmodeli [4]'te tanıtmışlardır. Bu üstmodel ile JADE ve JACK ÇES geliştirme platformlarına ait üstmodeller arasında model dönüşümleri tanımlanmıştır. Öte yandan [5]'te INGENIAS adı verilen etmen geliştirme metodolojisinin MGG paradigmasına uygun olarak yeniden tasarımı anlatılmıştır. Ancak bu çalışmaların genel bir ÇES MGG süreci tanımlamadıkları ve daha çok üzerinde çalıştıkları etmen yazılımı geliştirme çerçevesine bağlı oldukları görülmektedir. Ayrıca bu çalışmalarda etmenlerin Anlamsal Web ortamında çalışmaları ve bu ortama ait sistem bileşenlerinin tanımlanması da göz önünde bulundurulmamıştır.



Şekil 4: SEAGENT etmen planlarının HGA'ya dayalı olarak modellenmesi



Şekil 5: SEAGENT ÇES organizasyonlarının modellenmesi



Şekil 6: SEAGENT etmen planları için otomatik üretilen OWL ontolojileri (solda) ve eylem kodları (sağda)

6. Sonuç ve İleriye Yönelik Çalışmalar

Bu bildiri ontoloji tabanlı ÇES'lerin model güdümlü geliştirilmesi için gereken sürece ait adımlar tanımlanmıştır ve uygulanacak yöntem üzerinde durulmuştur. ÇES sistemlerinin platforma özgü seviyede modellenmesi için gereken süreç adımı somutlaştırılarak yazılım araçları ile nasıl desteklendiği anlatılmıştır.

Devam eden çalışmalarımızın ilk ürünleri bildiri de anlatıldığı gibi platforma özgü seviyede etmen modellemeye ait modelleme ve model dönüşümü araçlarıdır. İleriye yönelik planlanan ilk çalışma birden çok etmen sistemini değişik bakış açılarından destekleyecek ve aynı zamanda ontoloji desteği de sağlayacak platform bağımsız bir üstmodelin oluşturulmasıdır. Bu üstmodel oluşturulduktan sonra etmen yazılımı geliştiricilerinin bu üstmodele uygun modelleme yapabilecekleri ve modellerinden bu bildiri de tanıtılan SEAGENT ve JADEx ÇES platformlarına dönüşümleri sağlayacak araçların hazırlanması hedeflenmektedir. Benzer şekilde yazılım bağımsız bir üstmodel ve buna bağlı araçların oluşturulması ise bir sonraki çalışma olacaktır.

7. Teşekkür

Bu çalışma Türkiye Bilimsel ve Teknik Araştırma Kurumu (TÜBİTAK) Elektrik, Elektronik ve Enformatik Araştırma Grubu (EEEAG) tarafından 108E141 no'lu proje kapsamında desteklenmektedir.

8. Kaynakça

- [1] Bergenti, F., Gleizes, M-P. ve Zambonelli, F., *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, Kluwer Academic Publishers, Boston, 2004.
- [2] Berners-Lee, T., Hendler, J. ve Lassila, O., "The Semantic Web", *Scientific American*, 284(5):34-43, 2001.
- [3] Gračanin, D., Singh, H. L., Bohner, S. A. ve Hinchey, M. G., "Model-Driven Architecture for Agent-Based

Systems", *Lecture Notes in Artificial Intelligence*, 3228:249-261, 2005.

- [4] Hahn, C., Madrigal-Mora, C. ve Fischer, K., "A platform-independent metamodel for multiagent systems", *Autonomous Agents and Multi-agent Systems*, 18(2):239-266, 2009.
- [5] Pavon, J., Gomez, J. ve Fuentes, R., "Model Driven Development of Multi-Agent Systems", *Lecture Notes in Computer Science*, 4066:284-298, 2006.
- [6] Perini, A. ve Susi, A., "Automating Model Transformations in Agent-Oriented Modeling", *Lecture Notes in Computer Science*, 3950:167-178, 2006.
- [7] Kardaş, G., Göknil, A., Dikenelli, O. ve Topaloğlu, N. Y., "Model Driven Development of Semantic Web Enabled Multi-agent Systems", *International Journal of Cooperative Information Systems*, 18(2): 261-308, 2009.
- [8] Pokahr, A., Braubach, L., Walczak, A. ve Lamersdorf, W., "Jadex - Engineering Goal-Oriented Agents", (Kitap Bölümü) ss.254-258, *Developing Multi-Agent Systems with JADE*, Wiley Publishing, West Sussex, 2007.
- [9] Rao, A. ve Georgeff, M., "BDI Agents: From Theory to Practice". First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA, ss. 312-319, 1995.
- [10] Dikenelli, O., Erdur, R. C., Kardaş, G., Gümüş, Ö., Seylan, İ., Gürçan, Ö., Tiryaki, A. M. ve Ekinci, E. E., "Developing Multi Agent Systems on Semantic Web Environment using SEAGENT Platform", *Lecture Notes in Artificial Intelligence*, 3963:1-13, 2006.
- [11] Williamson, M., Decker, K. ve Sycara, K., "Unified Information and Control Flow in Hierarchical Task Networks", *AAAI-96 Workshop*, ss. 142-150, 1996.
- [12] Gürçan, Ö., Kardaş, G., Gümüş, Ö., Ekinci, E. E. ve Dikenelli, O., "An MAS Infrastructure for Implementing SWSA based Semantic Services", *Lecture Notes in Computer Science*, 4504:118-131, 2007.
- [13] Amor, M., Fuentes, L. ve Vallecillo, A., "Bridging the Gap Between Agent-Oriented Design and Implementation Using MDA", *Lecture Notes in Computer Science*, 3382:93-108, 2005.