

Anlamsal Web Tabanlı Etmen Sistemlerinin Geliştirilmesinde Model Tabanlı Yaklaşım

Arda Göknil¹, Geylani Kardeş², N. Yasemin Topaloğlu¹, Oğuz Dikenelli¹

¹Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, Bornova, İzmir, Türkiye

²Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, Bornova, İzmir, Türkiye
{arda.goknil, geylani.kardas, yasemin.topaloglu, oguz.dikenelli}@ege.edu.tr

Özet

Model Tabanlı Mühendislik farklı soyutlama seviyelerindeki modelleri kullanarak yazılım geliştirmede karşılaşılan sorunları gidermeyi hedeflemektedir. Yakın zamanda model tabanlı yaklaşımlar etmen tabanlı yazılım geliştirme alanında da kullanılmaya başlanmış ve bu alanda yürütülen araştırmalar yoğunluk kazanmıştır. Ancak Anlamsal Web ortamında çalışacak etmen sistemlerinin modellenmesini göz önüne aldığımızda yürütülen çalışmaların yeterli olmadığı ve henüz hiçbirinin model tabanlı tekniklerde anlamsal web bileşenlerini tanımlamadıkları görülmektedir. Bu çalışmamızda, Model Tabanlı Mühendislik kullanılarak Anlamsal Web yetenekli Çok-etmenli Sistemler'in geliştirilmesine ait yaklaşımımız tanıtılmaktadır. Bu kapsamda, Model Tabanlı Mühendislik'in bir uygulaması olan Model Tabanlı Mimari bağlamında Anlamsal Web tabanlı etmen varlıklarının nasıl tanımlandığı, bu tür Çok-etmenli Sistemler'in geliştirilmesi için Model Tabanlı Mimari'nin bileşenleri ve gereksinimleri incelenmiş ve ayrıca Anlamsal Web yetenekli bir Çok-etmenli Sistem'in temel varlıklarının tanımlandığı çekirdek bir etmen üst-modeli (metamodel) sunulmuştur.

1. Giriş

Bilgi ve ağ teknolojilerinde görülen ilerlemeler aynı zamanda yazılım sistemlerinin de giderek karmaşıklaşmasına neden olmaktadır. Bu karmaşıklıklarla başa çıkmak için yazılım mühendisliği alanında çalışmalarda bulunan araştırmacılar yeni yazılım geliştirme yaklaşımları, metodları ve teknikleri öne sürmektedirler. *Model Tabanlı Mühendislik* farklı soyutlama seviyelerindeki modelleri kullanarak yazılım geliştirmedeki karmaşıklığı azaltmayı hedeflemektedir. Model tabanlı mühendisliği yerine getirmek ve bundan faydalanmak için farklı

seviyelerde model dönüşümü sağlayacak yazılım araçlarının bulunması gerekmektedir. *Model dönüşümü*, farklı soyut seviyelerindeki modellerin otomatik eşlenmesi için bir veya daha fazla kaynak modelini girdi olarak alır ve gerekli bir dizi dönüşüm kuralına bağlı olarak yine bir veya daha fazla hedef modelini çıktı olarak ortaya koyar.

Yakın zamanda model tabanlı yaklaşımlar etmen tabanlı yazılım geliştirme alanında da kullanılmaya başlanmış ve belli başlı araştırma konularından biri haline gelmiştir [5] [8] [4]. Çok-etmenli Sistemler'in geliştirilmesinde mümkün olan en yüksek soyutlama seviyesinde çalışmak oldukça kritik bir öneme sahiptir. Çünkü iç karmaşıklıklarından, dağıtık yapılarından ve açık sistemler olmalarından dolayı Çok-etmenli Sistemler'in kod seviyesi detaylarını ortaya çıkarmak ve gözlemek neredeyse imkansızdır.

Anlamsal Web evrimi şüphesiz etmen araştırmalarına yeni bir vizyon getirmiştir. Bu *İkinci Jenerasyon Web*, Dünya Geneli Ağ'ı (WWW) web sayfası içeriklerinin ontolojiler kullanılarak yorumlanabileceği bir seviyeye taşınmayı hedeflemektedir. Söz konusu yorumlamanın ve anlam çıkarsamaların otonom hesaplama birimleri – yani etmenler – tarafından insan kullanıcıları adına yerine getirileceği açıktır.

Anlamsal Web ortamının kendine özgü mimari varlıklarının ve farklı bir semantiğinin olduğu, bu ortam üzerinde çalışacak Çok-etmenli Sistemler modellenirken göz önünde bulundurulmalıdır. Bu nedenle etmen sistemleri için ortaya konan modelleme tekniklerinin ve geliştirme sürecinin bu yeni ortamı yeni üst-varlıklar (meta-entity) ve üst-yapılar (meta-structure) tanımlayarak desteklemesi gerekmektedir. Literatürde her ne kadar etmen sistemlerinin geliştirilmesi için model tabanlı yaklaşımlar tanımlayan çeşitli çalışmalar [4] [8] [9] [16] yer alsada da bu çalışmalarda etmenlerin Anlamsal Web ortamında

çalışmaları ve bu ortama ait sistem bileşenlerinin tanımlanması göz önünde bulundurulmamıştır.

Bu çalışmamızda, Model Tabanlı Mühendislik kullanılarak Anlamsal Web yetenekli Çok-etmenli Sistemler'in geliştirilmesi için yaklaşımımızı sunmaktayız. Bu kapsamda, Model Tabanlı Mühendislik'in bir uygulaması olan Model Tabanlı Mimari bağlamında Anlamsal Web tabanlı etmen varlıklarının nasıl tanımlandığı, bu tür Çok-etmenli Sistemler'in geliştirilmesi için Model Tabanlı Mimari'nin bileşenleri ve gereksinimleri incelenmiş ve ayrıca Anlamsal Web yetenekli bir Çok-etmenli Sistem'in temel varlıklarının tanımlandığı çekirdek bir etmen üst-modeli (metamodel) tanıtılmıştır.

Bildirinin ikinci bölümünde Model Tabanlı Mühendislik, üst-modelleme, model dönüşümleri, Anlamsal Web tabanlı Çok-etmenli Sistem mimarisi ve bunların bütünleştirilmesine ait genel bir bakış sunulmaktadır. Üçüncü bölümde önerdiğimiz etmen üst-modeli anlatılmaktadır –ki bu üst-model, Anlamsal Web yetenekli Çok-etmenli Sistemler'in geliştirilmesinde model tabanlı bir mimariyi dahil etmenin ilk adımınıdır. Önerilen bu üst-modelden çalıştırılabilir modellerin türetilmesi konusu ise dördüncü bölümde yer almaktadır. Son bölümde elde edilen sonuçlar ve gelecek için hedeflenen çalışmalar yer almaktadır.

2. Anlamsal Web Tabanlı Etmen Sistemleri için Model Tabanlı Mühendislik

Anlamsal Web'in şu anki Web'in bir uzantısı olan, bilginin düzgün tanımlanmış bir anlama sahip olduğu ve insanlar ile bilgisayarların beraber çalışabildikleri bir web olarak çalışması hedeflenmektedir [25]. Günümüze kadar Web, veri ve bilgilerin otomatik olarak işlenebildiği bir ortamdan çok insanlar için doküman sağlayan bir ortam olacak şekilde geliştirilmiştir. Anlamsal Web ise ilgili otomatik işlemeyi gerçekleştirme amacına sahiptir. Doğal olarak bu otomatik işleme ortamında insan kullanıcıları adına faaliyetlerde bulunması gereken otonom yapıların bulunması gerekmektedir.

Günümüzde hem etmen araştırmaları hem de anlamsal web üzerine yapılan çalışmalar birbirini destekleyecek şekilde devam etmektedir. Çünkü yazılım etmenlerinin anlamsal web ortamlarında çalışacağına ve özellikle yetenekleri anlamsal olarak modellenmiş servislerle etmenlerin etkileşime gireceğine inanılmaktadır. Bu nedenle etmen sistemlerini geliştirmeye yönelik kullanılacak modelleme dillerinin ve onların ürünü olan etmen üst-

modellerinin anlamsal web yapılarını da kapsayacak şekilde tanımlanması gerekmektedir.

Model Tabanlı Mühendislik (Model Driven Engineering – MDE) [12], yazılım sistemlerinin model tabanlı geliştirilebilmesi için ortaya konulan mühendislik ilkelerini kapsamaktadır. Anlamsal web tabanlı yazılım etmenlerinin geliştirilmesi için *Model Tabanlı Mühendislik* kapsamında ortaya konan bu ilkelerden yararlanılabilir. *Object Management Group'un (OMG)*, Kasım 2000 tarihinde *Model Tabanlı Mühendislik* yaklaşımının bir uygulaması olarak ortaya koyduğu *Model Tabanlı Mimari (Model Driven Architecture – MDA)* [20], etmen sistemlerinin model tabanlı geliştirilebilmesi için uygun bir alt yapı sunmaktadır. Literatürde etmen tabanlı yazılım sistemlerinin Model Tabanlı Mimari'ye uygun olarak geliştirilmesi için bazı öneriler [4] [5] [8] [24] sunulmuş olmasına rağmen bu önerilerden hiç biri anlamsal web tabanlı etmen sistemlerini desteklememektedir.

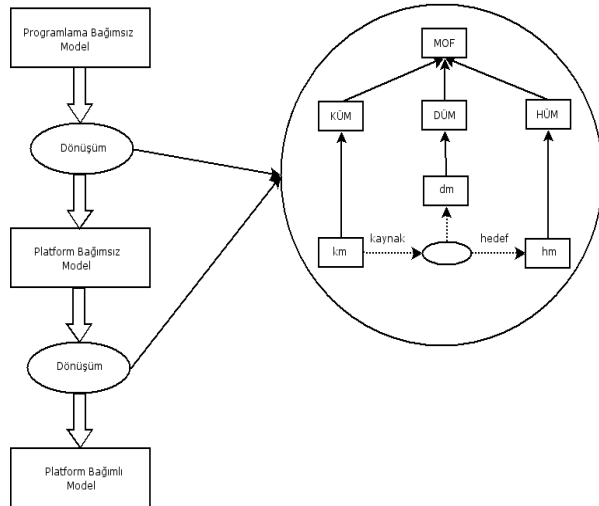
Model Tabanlı Mimari ile nesne modellerinin çalıştırılabilir bileşenlere ve uygulamalara dönüştürülerek yazılım sistemlerinin geliştirilmesi amaçlanmaktadır. Model Tabanlı Mimari kapsamında üç soyutlama seviyesi tanımlanmaktadır: Programlama Bağımsız Modeller (*Computation Independent Models – CIMS*), Platform Bağımsız Modeller (*Platform Independent Models – PIMS*) ve Platforma Özgü Modeller (*Platform Specific Models – PSMs*). Tüm bu soyutlama seviyelerinin en altında da kaynak kod yer almaktadır. Geliştirici, programlama bağımsız modelleri kullanarak sadece problem içerisinde yer alan varlıkları ve bu varlıklar arasındaki ilişkileri modelleyecektir. Bu soyutlama seviyesindeki bir modelde uygulama ile ilgili herhangi bir bilgi yer almamaktadır. Örneğin bir etmen sistemi için programlama bağımsız model, turizm alanı gibi anlamsal web etmenlerinin otel işlemlerine destek vereceği bir alanı tanımlayabilir. Ancak programlama bağımsız modellerden platform bağımsız modellere otomatik olarak geçebilmek için geliştirici, programlama bağımsız model üzerinde işaretleme işlemini yapmalıdır. Model dönüşüm tanımları ise programlama bağımsız modelde yer alan işaretlemelerden yola çıkarak bir alt seviyedeki platform bağımsız modeli oluşturacaktır. Örneğin, programlama bağımsız modelde yer alan varlıklar işaretlenerek platform bağımsız modelde etmenler ve anlamsal web etmenleri oluşturulabilmelidir. Bauer ve Odell [5] bir etmen sistemi için muhtemel programlama bağımsız modelleri belirtmişlerdir.

Platform bağımsız modeller, herhangi bir platformu temel almamasına rağmen programlanabilir yapıları içerisinde barındırmaktadır. Platform bağımsız

modellerden herhangi bir platforma özgü modele geçiş tanımlanabileceği gibi bu modellerden doğrudan platforma özgü kaynak kod da üretilebilir olmalıdır.

Anlamsal web tabanlı bir etmen sistemi için tanımlanacak olan platform bağımsız model, sisteme değişik bakış açılarından yaklaşan bir yapıda olmalıdır. Benguria [10], servis tabanlı mimariler için tanımlanacak platform bağımsız modellerde dört ana bakış açısı tanımlamıştır: *Bilgi (information aspect)*, *Servis (service aspect)*, *İşlem (process aspect)* ve *Servis Kalitesi (quality of service aspect)*.

Anlamsal web tabanlı bir etmen sistemi için platform bağımsız model, belirli bir etmen çerçevesine ait olmayan yapıları ve bu yapılar arasındaki ilişkileri içermelidir. Bir etmen platformuna bağımlı olmayan bu yapılar, SEAGENT [21] gibi anlamsal web desteği sağlayan bir etmen sistemine ait platform bağımlı model içerisindeki yapılara dönüştürülebilir. Bu yaklaşımın en belirgin özelliği, platform bağımlı modellere dönüşümde otomatik olarak istenilen platforma ait model üretilebilmektedir. Şekil-1’de Model Tabanlı Mimari kapsamında geliştirme süreci ve modeller arasında tanımlanan dönüşüm adımları verilmektedir.



Şekil 1: Model Tabanlı Mimari için Model Seviyeleri ve Dönüşümler

Şekil 1’deki gibi üst modele dayanan dönüşümlerde kaynak model (km), dönüşümün sonunda elde edilecek hedef model (hm) ve dönüşüm modeli (dm) – dönüşüm kuralları tanımlanmış olmalıdır. Model dönüşüm işleminde öncelikle başlangıç modeli yani kaynak model içerisinde kaynak sorgular işletilir ve değiştirilmek istenen yapılar belirlenir. Daha sonra model dönüşümü için tanımlanmış olan dönüşüm

kuralları, belirlenen bu yapılar üzerinde uygulanır. Dönüşüm kurallarının uygulanması sonucu üretilen hedef modelin hedef üst modeline (HÜM) uygun olması gereklidir. Böylece üretilen modelin doğruluğu, hedef üst model ile yapılan eşlemlerde sınanır. Şekil 1’de görülen kaynak ve hedef üst modeller, dönüşüme girecek kaynak modeller ile dönüşüm sonucu elde edilecek hedef modellerin bilgilerinin tutulduğu modellerdir. Bu dönüşüm tanımları ise değişik model dönüşüm dilleri [1] [2] [7] [18] kullanılarak ifade edilebilir. Şekil 1’de verilen Model Tabanlı Mühendislik için tanımlanan değişik modeller arasındaki dönüşümler ve dönüşüm mekanizması, etmen tabanlı modeller ve bu modeller arasındaki dönüşümler için kullanılabilir. Perini [4] tarafından ortaya konan çalışma etmen tabanlı sistemlerin Model Tabanlı Mimari kapsamında geliştirilebilmesi için tanımlanan dönüşümlere güzel bir örnektir. Perini [4], çalışmada TEFKAT model dönüşüm dilini [18] kullanarak Tropos üst modelinde tanımlı plan yapılarını UML Activity modellerine dönüştürmüştür.

Şekil 1’den de görüleceği üzere modeller arası işbirliğini ve dönüşümü sağlamada önemli bir bileşen, üst modelleme (*meta-modeling*) kavramı ve üst modellerdir. Bir üst model (*meta-model*), belirli bir modelleme dilinde geçerli olan modelleri tanımlayan bir modeldir. Üst model gibi üst düzey tanımlamalar, modeller üzerinde çalışılmasını kolaylaştırır. Etmen tabanlı sistemlerin modellenmesinde ve bu etmen modellerinin birbirine dönüştürülmesi çalışmalarında etmen üst modellerinin tanımlanması önemli bir yer tutmaktadır. Bu kapsamda Gaia, Adelfe, PASSI [6] ve SODA [3] gibi etmen metodolojileri geliştirmek için çeşitli etmen üst modelleri önerilmiştir. Ancak bu etmen üst modellerinin bir çoğu sadece genel kavramları modellemek için kullanılmaktadır ve yine bu üst modeller, etmen geliştirme ortamları için tanımlanmış modeller değildir [16].

Önde gelen etmen üst modeli çalışmalarından biri de FIPA ile OMG’nin (*Object Management Group*) Etmen Özel İlgi Grubu’nun (*Agent SIG*) ortaklaşa yürüttükleri *Agent Class Superstructure Metamodel* [15] çalışmasıdır. Bu çalışma, bu grupların kendilerinin de belirttiği gibi başlangıç safhasındadır. Önerilen üst model; etmenleri, etmen rollerini, etmen gruplarını ve bunların birbirleri ile olan ilişkilerini tanımlamaktadır. Bu üst model, UML tabanlı olup aynı zamanda onun bir uzantısıdır. Etmen sistemleri için üst model çalışmaları yeni olmayıp ALAADIN üst modeli [13], bu üst model çalışmalarının öncülerindedir.

Diğer yandan MetaDIMA [26], Model Tabanlı Mimari yaklaşımından etkilenerek etmen sistem mimarileri ile etmen tabanlı metodolojiler arasındaki kopukluğu gidermeye çalışan model tabanlı bir diğer

proje. Pavon [16] ise çalışmasında INGENIAS adını verdikleri etmen geliştirme metodolojilerini yeniden tanımlayarak model tabanlı geliştirme paradigmasına uygun hale getirmiştir.

Etmen ortamlarına anlamsal web desteği getirmeleri açısından etmen üst modeli çalışmalarını incelediğimizde henüz bu çalışmalardan hiçbirinin bu desteği verecek yapıları oluşturmadıkları ortaya çıkmaktadır. Bunun gelecek nesil çok-etmenli yazılım sistemlerinin Anlamsal web ortamlarında çalışması gerekliliği ve etmenlerin planları dahilinde yetenekleri anlamsal olarak tanımlanmış web servisleri ile etkileşimde olma gereksinimleri de düşünülürse önemli bir dezavantaj olduğu açıktır.

3. Anlamsal Web Üzerinde Çalışacak Çok-etmenli Sistemler İçin Bir Üst-Model

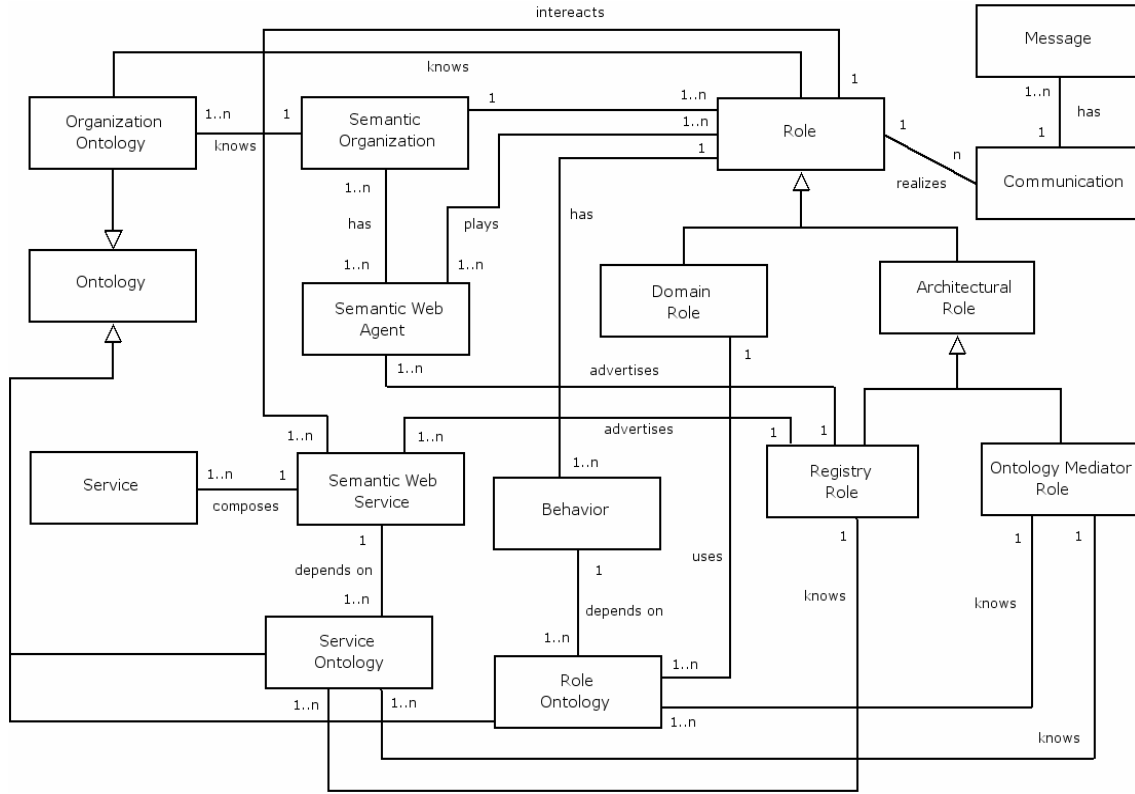
Anlamsal Web ortamında çalışacak etmen sistemleri için genel kavramları ve varlıkları (entity) tanımlayan bir üst model (metamodel) [11]'de tanımlanmıştır. Bu üst model, Model Tabanlı Mimari'ye dayalı olarak bu tip sistemlerin geliştirilmesi sırasında gereksinim duyulan Platform Bağımsız Model ve Platforma Özgü

Model'lere ait varlıkların tanımlanmasında anahtar rol üstlenmektedir. Tanımlanan çekirdek üst model Şekil 2'de yer almaktadır. Açıklamalar sırasında anlam karmaşıklıklarını önlemek amacıyla model varlıkları, orijinal İngilizce adları ile kullanılacaktır.

Semantic Organization, *Semantic Web Agent* larının oluşturduğu bir yapı olup etmenlerin organizasyonel rollerine dayalı olarak oluşturulur.

Bir *Semantic Web Agent*, ortamdaki hem diğer etmenler ile hem de anlamsal web servisleri ile etkileşime geçme yeteneğine sahip otonom bir varlık olarak tanımlanmaktadır.

Burada dikkat edilmesi gereken organizasyonun sadece etmenlerden oluştuğu ve etmen harici anlamsal yapılar içermediğidir. Sadece organizasyonel rollere yer verilmektedir ve bu roller üye etmenler tarafından oynanmaktadır. Organizasyonel rolleri göz önüne aldığımızda bir Anlamsal Web Etmeni aynı anda birden fazla organizasyonun üyesi olabilir. Bu demektir ki *bir etmen aynı anda birden fazla rolü oynayabilir ve bir rol Semantic Organization içeriğinde (context) birden fazla Semantic Web Agent tarafından oynanabilir.*



Şekil 2: Anlamsal Web yetenekli Çok-etmenli Sistemler için çekirdek bir üst-model

Role varlığının mimari ve alan (domain) tabanlı rollerin görev tanımlarına bağlı olarak model içerisinde özelleşmesinin gerektiğine inanmaktayız. Bu nedenle model içerisinde bu üst-varlığın (meta-entity) iki alt-varlığı (sub-entity) yer almaktadır: *Architectural Role* ve *Domain Role*. Bir *Architectural Role*, organizasyonun içeriğinden bağımsız olarak platform içerisinde en az bir etmen tarafından üstlenilmesi gereken anlamsal web yetenekli çok-etmenli sistemler için zorunlu bir rolü tanımlamaktadır. Buna karşılık *Domain Role* etmen organizasyonu içeriğine özgü olup tamamıyla özelleşmiş bir iş alanı için oluşturulmuş bir *Semantic Organization*'ın gereksinimlerine ve görev tanımlamalarına dayalıdır.

Organizasyonda yer alan bir kısım etmenin [11]'de tanımlanan kavramsal mimarinin Mimari Servis Katmanı'nda tanımlanmış olan servisleri sağlamak amacıyla tanımlanmış rolleri oynaması gerekir. Bu nedenle üst-modelde *Architectural Role*'un iki özelleşmiş alt-varlığı da yer almaktadır: *Registry Role* ve *Ontology Mediator Role*.

Bir *Role* bir ya da daha fazla *Behavior*'u içermektedir. Bir *Semantic Web Agent*'in görev tanımları ve ilgili görev işletim süreçleri *Behavior* varlıkları içerisinde modellenmektedir.

Oynanan rollere bağlı olarak etmenlerin çoğu zaman başka bir etmen ile iletişime geçmesi gerekmektedir. Üst-modeldeki *Communication* varlığı platformdaki iki etmen arasındaki bir etkileşimi tanımlamaktadır. Etkileşim önceden tanımlı etmen etkileşim protokolüne dayalı olarak gerçekleşir. Bir *Communication* varlığı bir ya da daha fazla *Message* varlığı içermektedir. Her bir mesajın içeriği *RDF* (*Resource Description Framework*) tabanlı bir anlamsal içerik dili ile ifade edilebilmektedir.

Bir *Semantic Web Service* üst-varlığı etmen servisi hariç olmakla birlikte yetenekleri ve etkileşimleri mevcut sistem içerisinde anlamsal olarak tanımlanmış herhangi bir servisi temsil etmektedir. Bir *Semantic Web Service* bir ya da daha fazla *Service* varlığını içermektedir. Her servis bir web servisi ya da gerçek hayat uygulamasında önceden tanımlı başka özel bir etkileşim protokolüne sahip bir servis olabilir. Ancak mutlaka ilgili servisin platformun etmenleri tarafından kullanılabilir bir anlamsal arayüzünün olması gerekmektedir. Burada dikkat edilmesi gereken modelde *Semantic Web Agent*'leri ile *Semantic Web Service*'leri arasındaki ilişkinin *Role* varlığı üzerinden sağlanmış olmasıdır. Çünkü **etmenler organizasyon içerisinde tanımlanmış olan rollerine dayalı olarak anlamsal web servisleri ile etkileşimde bulunurlar.**

Diğer Anlamsal Web ortamları gibi anlamsal web yetenekli çok-etmenli sistemler de ontolojiler olmaksızın düşünülemezler. Bu nedenle önerdiğimiz

üst-model bir *Ontology* varlığı ve onun ihtiyaç duyulan alt-varlıklarını içermektedir. Bir *Ontology* bir çok-etmenli sistem üyesi için herhangi bir bilgi toplama ve muhakeme (reasoning) kaynağını temsil eder. Ontolojiler, belirli bir iş alanı içeriğini sağlayan, çok-etmenli sisteme ait bilgi tabanını oluşturmaktadır.

Ontology üst-varlığının *Organization* *Ontology*, *Service Ontology* ve *Role Ontology* adı verilen alt-varlıkları ilgili üst-model varlıkları tarafından kullanılmaktadır. Örneğin servislerin anlamsal arayüzleri ve yetenek tanımları *Service Ontology*'e bağlı olarak oluşturulmaktadır ve bu ontoloji *Semantic Web Agent*'leri tarafından *Semantic Web Service*'lerinin bulunması ve çalıştırılması için kullanılmaktadır.

4. Anlamsal Web Tabanlı Etmen Sistemleri için Platform Bağımsız Model Önerisi

Anlamsal web tabanlı etmen sistemleri için önerdiğimiz [11] üst model, sistemde yer alan genel kavramları ve yapıları içermektedir. Bu üst model, bir etmen sistemi için Model Tabanlı Mimari kapsamında platform bağımsız model ve platform bağımlı model yapılarını tanımlamak için gerekli olan anahtar noktaların tanımlanması için önemlidir. Ancak bu üst model, anlamsal web tabanlı etmen sistemleri için platform bağımsız model üst modeli olarak kabul edilemez. Bunun için üst modelin detaylandırılması gerekmektedir. Örneğin üst modelde, *SemanticWebService* ile *ServiceOntology* gibi yeni model elemanlarının etmen-servis etkileşimi doğrultusunda detaylandırılması gerekmektedir. Bu doğrultuda çözülmesi gereken sorunlardan ilki bu etmen-servis iletişimde gerekli olan yapıların üst model içerisinde tanımlanması olacaktır.

Anlamsal web etmenlerinin bulunması ve çalıştırılması için bir anlamsal web servisi, servis arayüzünü (service interface) ve servis süreç mekanizmasını (service process mechanism) içermektedir. Anlamsal web etmen kaynak kodunu platform bağımlı model üzerinden üretebilmek için bu arayüz ve süreç yapılarının üst model içerisinde tanımlanması gerekmektedir. *OWL-S* (*Ontology Web Language-Service*) [22] ve *WSMO* (*Web Service Modeling Ontology*) [27] kapsamında web servislerinin anlamsal olarak modellenmesi üzerine çalışılmasına rağmen anlamsal web ortamında çalışan bu tip web servislerinin herhangi bir platformdan bağımsız olarak gösterilmesine ilişkin bir standart geliştirilmemiştir.

Gronmo [23] tarafından gerçekleştirilen çalışma, anlamsal web servislerinin Model Tabanlı Mimari

kullanılarak geliştirilmesi için iyi bir örnektir. Gronmo [23], çalışmasında platformdan bağımsız olarak anlamsal web servislerinin tanımlanabilmesi için bir UML profili tanımlamıştır. Bu UML profili sayesinde anlamsal web servislerini tanımlayan değişik platformlara ait OWL-S ve WSML dokümanlarını otomatik olarak üretmek mümkün olmaktadır. Bu çalışma anlamsal web servislerini model tabanlı geliştirmek için iyi bir örnek olmasına rağmen, anlamsal web servisleri ile etmenlerin etkileşimini içeren herhangi bir yapı bulundurmamaktadır.

Anlamsal web tabanlı etmen üst modeli için geliştirilmesi gereken diğer bir bölüm ise planlama alt yapısıdır. Davranış kütüphanesi içerisinde yer alan yeniden kullanılabilir plan alt yapılarından birisi de *Hierarchical Task Network planning* adı verilen HTN planlarıdır [17]. Model Tabanlı Mimari bakış açısıyla HTN gibi planlama mekanizmaları tarafından tanımlanan plan yapıları platform bağımlı model seviyesinde yer almaktadır. Örneğin anlamsal web tabanlı etmen platformu olan SEAGENT içerisindeki plan yapıları, [14] ve [19]'da belirtilen plan mekanizmasına göre gerçekleştirilmiştir. SEAGENT platformunun üst modelini platform bağımlı model üst modeli olarak düşünürsek, HTN mekanizmasına özel yapılar bu üst model içerisinde yer alacaktır. Platform bağımsız model seviyesinde ise HTN ve diğer planlama mekanizmalarından bağımsız olarak genel planlama kavramlarıyla ilgili yapılar üst model içerisinde yer almalıdır.

5. Sonuç ve Gelecekteki Çalışmalar

Bu bildiriye, anlamsal web tabanlı etmen yazılım sistemlerinin tasarımı ve geliştirilmesine yönelik model tabanlı bir yaklaşım tartışılmıştır. Literatürde bulunan etmen sistemlerinin geliştirilmesine yönelik modelleme dillerinin ve üst modellerin anlamsal web tabanlı etmen sistemleri için herhangi bir destek sağlamadıkları görülmüştür.

Model Tabanlı Mimari bağlamında Anlamsal Web yetenekli etmen varlıklarının bir etmen üst modelinde nasıl yer aldığı bu bildiriye incelenmiştir. Bu tip çok-etmenli sistemlerin geliştirilmesi için Model Tabanlı Mimari kapsamında platform bağımlı ve platform bağımsız modellerin sahip olması gereken yapılar üzerinde durulmuştur.

İleriye yönelik çalışmalardan ilki, üst modelin gereksinimlere uygun üst varlıklarla genişletilmesidir. Statik bakış açısı yanında modelin üst yapılar arası etkileşimlerinin de ortaya konulması ikinci planlanan çalışma olarak tanımlanmıştır. Tüm bu alt yapının tamamlanmasından sonra alt yapıya uygun etmen üst

modelleri arasında dönüşümlerin tanımlanarak SEAGENT [21] etmen geliştirme platformuna uygun kaynak kod üretilmesi hedeflenmektedir.

6. Kaynaklar

[1] Agrawal, A., Karsai, G., Shi., F. Interpreter Writing using Graph Transformations, *Vanderbilt University Technical Report*, ISIS-03-401, 2003.

[2] Kalnins, A., Barzdins, J., Celms, E. Model Transformation Language MOLA, Lecture Notes in Computer Science (LNCS), 3599/2005, MDAFA 2004.

[3] Molesini, A., Denti, E., Omicini, A. MAS Meta-models on Test: UML vs. OPM in the SODA Case Study, *CEEMAS 2005*, Budapest, Hungary.

[4] Perini, A., Susi, A. Automating Model Transformations in Agent-Oriented Modeling, *Agent Oriented Software Engineering Workshop (AOSE 2005)*, Utrecht, the Netherlands, 2005.

[5] Bauer, B., Odell, J. UML 2.0 and Agents: How to Build Agent-based Systems with the New UML Standard, *Journal of Engineering Applications of Artificial Intelligence*, 18(2):141– 157, 2005.

[6] Bernon, C., Cossentino, M., Gleizes, M., Turci, P., Zambonelli, F. A Study of some Multi-Agent Meta-Models, *Agent Oriented Software Engineering Workshop (AOSE 2004)*, New York, USA

[7] Jouault, F., Kurtev, I. Model Transformations with ATL, *MODELS Satellite Events 2005*, Revised Selected Papers in LNCS 3844 Springer 2006, Montego Bay, Jamaica, 2005.

[8] Jayatilleke, G.B., Padgham, L., Winikoff, M. Towards a Component based Development Framework for Agents, *Second German Conference on Multiagent System Technologies (MATES 2004)*, Erfurt, Germany, 2004.

[9] Jayatilleke, G.B., Padgham, L., Winikoff, M. A Model Driven Development Toolkit for Domain Experts to Modify Agent Based Systems, *Agent Oriented Software Engineering Workshop (AOSE 2006)*, Hakodate, Japan.

[10] Benguria, G., Larrucea, X., Elvesaeter, B., Neple, T., Beardsmore, A., Winchester, M. A Platform Independent Model for Service Oriented Architectures, *I-ESA 2006*, Bordeaux, France, 2006.

[11] Kardas, G., Goknil, A., Dikenelli, O., Topaloglu, N.Y. Metamodeling of Semantic Web Enabled Multiagent Systems, *Multiagent Systems and Software Architecture (MASSA 2006)*, Erfurt, Germany, 2006.

[12] Bezin, J. Model Driven Engineering: Principles, Scope, Deployment and Applicability, *Proceedings of the*

- Summer School on Generative and Transformational Techniques in Software Engineering (GTTSE'05)*, Braga, Portugal, 2005.
- [13] Ferber, J., Gutknecht, O. A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems, In: *Proc. 3rd International Conference on Multi-Agent Systems*, IEEE Computer Society, 128-135, 1998
- [14] Graham, J.R., Decker, K., Mersic, M. DECAF – a flexible multi agent system architecture, *Autonomous Agents and Multi-Agent Systems*, 2003.
- [15] Odell, J., Nodine, M., Levy, R. A Metamodel for Agents, Roles and Groups, In: *Agent Oriented Software Engineering V: 5th International Workshop, AOSE 2004*. Lecture Notes in Computer Science, Vol. 3382, pp. 78-92. Springer-Verlag, Berlin Heidelberg (2005)
- [16] Pavon, J., Gomez, J., Fuentes, R. Model Driven Development of Multi-Agent Systems, *Lecture Notes in Computer Science (LNCS)*, 4066/2006, EC-MDA 2006.
- [17] Erol, K., Hendler, J.A., Nau, D.S. Complexity Results in HTN Planning, *Ann. Math. Artif. Intell.*, 1996.
- [18] Duddy, K., Gerber, A., Lawley, M. Model Transformation: A declarative, reusable patterns approach, *Proceedings of the 7th International Enterprise Distributed Object Computing Conference (EDOC'03)*, Brisbane, Australia, 2003.
- [19] Sycara, K., Williamson, M., Decker, K. Unified Information and Control Workflow in Hierarchical Task Networks, In: *Working Notes of the AAAI-96 workshop 'Theories of Action, Planning, and Control'*, 1996.
- [20] Object Management Group (OMG). MDA Guide Version 1.0.1. Document Number : omg/2003-06-01, 2003.
- [21] Dikenelli, O., Erdur, R.C., Kardas, G., Gümüs, Ö, Seylan, I., Gurcan, Ö, Tiryaki A.M., Ekinci, E.E. Developing Multi Agent Systems on Semantic Web Environment using SEAGENT Platform, *Proceedings of the 6th International Workshop Engineering Societies in the Agents World (ESAW'05)*, Kusadasi, Turkey, 2005.
- [22] The OWL Services Coalition. Semantic Markup for Web Services (OWL-S). <http://www.daml.org/services/owl-s/1.1/>, 2004.
- [23] Gronmo, R., Jaeger, M.C., Hoff, H. Transformations between UML and OWL-S, *Proceedings of the 1st European Conferenc on Model Driven Architecture – Foundations and Applications (ECMDA-FA'05)*, Nuremberg, Germany, 2005.
- [24] Depke, R., Heckel, R., Küster, J.M. Agent-Oriented Modeling with Graph Transformations, *Agent Oriented Software Engineering Workshop (AOSE 2000)*, Limerick, Ireland, 2000.
- [25] Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. *Scientific American*, 284(5):34– 43, 2001.
- [26] Guessoum, Z., Thieffaine, A., Perrot, J., Blain, G. META-DIMA: a Model-Driven Architecture for Multi-Agent Systems, URL: <http://www-poleia.lip6.fr/%7Eguessoum/MetaDima.html>, (last accessed: 2006)
- [27] WSMO Working Group. Web Service Modeling Ontology (WSMO). <http://www.wsmo.org/>.