



EGE ÜNİVERSİTESİ

YÜKSEK LİSANS TEZİ

**ANLAMSAL WEB YETENEKLİ ÇOK ETMENLİ
SİSTEMLER İÇİN BİR ALANA ÖZGÜ
MODELLEME DİLİNE AİT SÖZDİZİM**

Sebla DEMİRKOL

Tez Danışmanı : Yrd. Doç. Dr. Geylani KARDAŞ

Uluslararası Bilgisayar Anabilim Dalı

Bilim Dalı Kodu : 619.02.04

Sunuş Tarihi : 18.06.2012

Bornova-İZMİR

2012

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(YÜKSEK LİSANS TEZİ)

ANLAMSAL WEB YETENEKLİ ÇOK ETMENLİ

SİSTEMLER İÇİN BİR ALANA ÖZGÜ

MODELLEME DİLİNE AİT SÖZDİZİM

Sebla DEMİRKOL

Tez Danışmanı : Yrd. Doç. Dr. Geylani KARDAŞ

Uluslararası Bilgisayar Anabilim Dalı

Bilim Dalı Kodu : 619.02.04

Sunuş Tarihi : 18.06.2012

Bornova-İZMİR

2012

Sebla DEMİRKOL tarafından YÜKSEK LİSANS tezi olarak sunulan “Anlamsal Web Yetenekli Çok Etmenli Sistemler İçin Bir Alana Özgü Modelleme Diline Ait Sözdizim” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 18/06/2012 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı : Yrd. Doç. Dr. Geylani KARDAŞ

Raportör Üye : Yrd. Doç. Dr. Cengiz GÜNGÖR

Üye : Prof. Dr. Haluk GÜMÜŞKAYA

ÖZET**ANLAMSAL WEB YETENEKLİ ÇOK ETMENLİ SİSTEMLER
İÇİN BİR ALANA ÖZGÜ MODELLEME DİLİNE AİT SÖZDİZİM**

DEMİRKOL, Sebla

Yüksek Lisans Tezi, Uluslararası Bilgisayar Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Geylani KARDAŞ

Haziran 2012, 70 sayfa

Yazılım etmenleri ve bu etmenlerin bir araya gelerek oluşturduğu Çok-etmenli Sistemler, karmaşık sistemlerin hayata geçirilmesi için güçlü birer teknoloji olarak kullanılmaktadır. Etmen tabanlı yazılım sistemlerinin geliştirilmesine yönelik birçok yazılım çerçevesi bulunmaktadır. Ancak bu yazılım çerçevelerinin çoğu Anlamsal Webi desteklememektedir. Anlamsal Web, web sayfası içeriklerinin ontolojiler aracılığıyla makineler tarafından yorumlanabilmesini sağlamak ve Dünya Geneli Ağ'ın yeni jenerasyonunu oluşturmaktadır. Etmenlerin Anlamsal Web ile işbirliğini destekleyen mevcut yazılım geliştirme çerçeveleri, etmen alanında uzman olmayan sistem geliştiricilerinin kullanması için oldukça karmaşık yapıdadır. Ancak genellikle sistem ihtiyaçlarını ortaya koyan kullanıcıların programlamaya özgü yapıları bilmesine de gerek yoktur.

Bu tezde, çok-etmenli sistem geliştirme sırasında Anlamsal Web yapılarını ve bunlarla geleneksel etmen sistemi elemanları arasındaki etkileşimleri destekleyen yeni bir alana özgü modelleme diline ait sözdizim görsel ve metinsel olarak tanımlanmıştır. Oluşturulan bu yeni alana özgü dilin kullanımını örneklemek amacıyla metinsel somut söz dizimden kod üretimi gerçekleştirilmiştir.

Anahtar sözcükler: çok-etmenli sistem, Anlamsal Web, alana özgü modelleme dili, sözdizim, EMF, GMF, XTEXT, XPAND.

ABSTRACT**SYNTAX OF A DOMAIN SPECIFIC MODELING LANGUAGE
FOR SEMANTIC WEB ENABLED MULTI-AGENT SYSTEMS**

DEMİRKOL, Sebla

MSc in International Computer

Supervisor: Asst. Prof. Dr. Geylani KARDAŞ

June 2012, 70 pages

Agents and Multi-agent systems (MASs), which consist of the agents, are used as substantial technologies in order to develop complex systems with autonomous and reactive behaviour. In this sense, various software frameworks exist intended for improving agent oriented software development. However, semantic web is not supported by most of these software frameworks. The semantic web provides the interpretation of web page contents through ontologies by machines and represents the new generation of World Wide Web. The current software development frameworks which support the collaboration of agents with semantic web elements are complicated enough in order to be used by system developers who are not expert on the related domain. Furthermore, users, who define system requirements, usually do not need to be aware of the structures specific for agent programming.

In this thesis, graphical and textual syntaxes are defined for a domain specific modeling language (DSML) for MASs which specifically supports the interaction between agents and semantic web services on the Semantic Web environment. In order to exemplify the use of the related DSML, code generation from defined syntax is discussed.

Keywords: Multi-agent system, semantic web, domain specific language, syntax, EMF, GMF, XTEXT, XPAND.

TEŞEKKÜR

Öncelikle bu tez konusu üzerinde bana çalışma imkanı sunan tez danışmanım Yrd. Doç. Dr. Geylani KARDAŞ’a deneyimleri, bilgileri ve önerileriyle araştırma ve geliştirmeye yönlendirmesi, sağladığı kaynaklar ile destek olması, projede çalışma imkanı sunarak tez süresince her türlü yardımı esirgememesinden dolayı teşekkürü bir borç bilirim.

Tez çalışmalarım sırasında bilgi ve görüşlerini paylaşarak destek olan arkadaşlarım Moharram CHALLENGER ve Araş. Gör. Sinem GETİR’e, tezimin taslağını okuyarak son halini almasında yardımlarını esirgemeyen çalışma araştırma görevlisi arkadaşlarım Sercan DEMİRCİ ve Cemre CANDEMİR’e, çalışmalarımın her evresinde beni destekleyerek yanımda olan ablam Araş. Gör. Şebnem AKYOL ve Kasım Emre AKOVA’ya, teşekkür ederim. Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü’nde görevli değerli hocalarım, araştırma görevlisi arkadaşlarım ve idari personele, bütün çalışmalarım süresince her zaman yanımda olan aileme manevi desteklerinden dolayı teşekkürü bir borç bilirim.

“109E125” numaralı “Anlamsal Web Yetenekli Çok-Etmenli Sistemler İçin Alana Özgü Modelleme Dili” isimli Ar-Ge projesi kapsamında yüksek lisans çalışmamı maddi olarak destekleyen Türkiye Bilimsel ve Teknolojik Araştırma Kurumu’na (TÜBİTAK) teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
ABSTRACT.....	vii
TEŞEKKÜR.....	ix
ŞEKİLLER DİZİNİ.....	xiv
ÇİZELGELER DİZİNİ.....	xvii
KISALTMALAR DİZİNİ.....	xviii
1. GİRİŞ.....	1
1.1 Yaklaşım.....	3
2. ALT YAPI.....	6
2.1 Yazılım Etmenleri.....	6
2.2 Anlamsal Web.....	8
2.2.1 Anlamsal Web servisleri.....	9
2.3 Alana Özgü Dil.....	11
2.3.1 Soyut sözdizimi.....	11
2.3.2 Grafiksel somut sözdizimi geliştirimi için kullanılan araçlar.....	12

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
2.3.3 Metinsel somut sözdizimi geliştirimi için kullanılan araçlar.....	13
2.4 Etmen Geliştirme Çerçeveleri.....	14
3. İLGİLİ ÇALIŞMALAR.....	16
3.1 Etmen ve MAS Üstmodelleri ve Modelleme Dili Çalışmaları.....	16
3.2 MAS DSL ve DSML Çalışmaları.....	19
4. ANLAMSAK WEB ORTAMINDA ÇALIŞAN MAS'LAR İÇİN PLATFORM BAĞIMSIZ BİR ÜSTMODEL.....	21
4.1 MAS Bakışaısı.....	23
4.2 Anlamsal Web Etmeni İçsel Bakışaısı.....	23
4.3 Plan Bakışaısı.....	25
4.4 Rol Bakışaısı.....	26
4.5 Etkileşim Bakışaısı.....	27
4.6 Çevre Bakışaısı.....	28
4.7 Etmen-Anlamsal Web Servis Etkileşim Bakışaısı.....	29
4.8 Ontoloji Bakışaısı.....	31
5. GRAFİKSEL SOMUT SÖZDİZİMİN GERÇEKLEŞTİRİMİ.....	32

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
5.1 Anlamsal Web Etmeni İçsel Bakışaçası için Görsel Somut Sözdizimin Gerçekleştirimi.....	34
6. METİNSEL SOMUT SÖZDİZİMİN GERÇEKLEŞTİRİMİ.....	42
7. GRAFİKSEL SÖZDİZİMİN DEĞERLENDİRİLMESİ.....	52
8. METİNSEL SÖZDİZİMİN DEĞERLENDİRİLMESİ.....	55
9. SONUÇ.....	59
KAYNAKLAR DİZİNİ.....	60
ÖZGEÇMİŞ.....	73
EKLER	
Ek 1 Türkçe – İngilizce Terimler Sözlüğü	

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 Ecore üstmodeli ile geliştirilen örnek üstmodel.....	12
2.2 Ecore üstmodelinden oluşturulmuş diyagram görüntüsü.....	12
2.3 GMF ile geliştirilmiş örnek bir model.....	13
4.1 MAS bakış açısı.....	23
4.2 Anlamsal Web Etmeni içsel bakış açısı.....	24
4.3 Plan bakış açısı.....	25
4.4 Rol bakış açısı.....	27
4.5 Etkileşim bakış açısı.....	28
4.6 Çevre bakış açısı.....	29
4.7 Etmen-Anlamsal Web Servis Etkileşim bakış açısı.....	30
4.8 Ontoloji bakış açısı.....	31
5.1 “Agent.Ecore” üstmodeli.....	36
5.2 “Simple Tooling Definition Model”de üstvarlıkların seçimi.....	37
5.3 “Agent.gmfgraph”ın gösterimi.....	39

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
5.4 “Agent.gmfmap”ın gösterimi.....	40
6.1 Etmen-Anlamsal Web servis bakışaçasına ait metinsel sözdizimin bir kısmı.....	43
6.2 “checkAtLeastOneSemanticWebService” kuralı.....	44
6.3 “checkSWSNameStartsWithLowerCase” kuralı.....	45
6.4 “Name must be lower case” uyarısı için tanımlanan hızlı düzeltme.....	46
6.5 “IssueCodes.java” dosyası.....	46
6.6 Bir değişkenin birden fazla kez tanımlanmaması için yazılan kod parçası	47
6.7 “AgentSWSInteractionFormatter.java” dosyası.....	47
6.8 “SampleNewWizard” dosyasının bir kısmı.....	48
6.9 “Service.owl” dosyasının oluşturulması.....	50
6.10 “Service pofile”, “service process” ve “service grounding” dosyalarına referansların tanımlanması.....	50
7.1 Takas yöneticisinin Anlamsal Web etmeni içsel bakışaçasına göre GMF tabanlı editörde modellenmesi.....	53
7.2 Takas yöneticisinin etmen-Anlamsal Web servis etkileşim bakışaçasına göre GMF tabanlı editörde modellenmesi.....	54

ŞEKİLLER DİZİNİ (devam)ŞekilSayfa

8.1 Takas yöneticisinin Anlamsal Web etmeni içsel bakış açısına göre Xtext tabanlı editörde modellenmesi.....	55
8.2 Takas yöneticisinin etmen-Anlamsal Web servis etkileşim bakış açısına göre Xtext tabanlı editörde modellenmesi.....	57

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
5.1 SEA_ML somut sözdizim elemanları.....	33
6.1 SEA_L üstmodel elemanlarının JADEX ve OWL-S üstmodel elemanları ile eşleştirilmesi.....	49

KISALTMALAR DİZİNİ

Kısaltmalar

ACL	Agent Communication Language.
ACSM	Agent Class Superstructure Metamodel.
ADF	Agent Definition File.
AML	Agent Modeling Language.
AOSD	Agent Oriented Software Development.
AOSE	Agent Oriented Software Engineering.
ATL	Atlas Transformation Language.
AUML	Agent UML.
BDI	Belief-Desire-Intention.
BNF	Backus-Naur Form.
CDF	Capability Definition File.
CNP	Contract Net Protocol.
DFKI	German Research Centre for Artificial Intelligence.
DSL	Domain Specific Language.
DSM	Domain Specific Modeling.
DSML	Domain Specific Modeling Language.
EBNF	Extended BNF.

KISALTMALAR DİZİNİ (devam)

Kısaltmalar

EMF	Eclipse Modeling Framework.
IEEE FIPA	IEEE Foundation for Intelligent Physical Agents.
GMF	Eclipse Graphical Modeling Framework.
HTML	HyperText Markup Language.
IEEE	Institute of Electrical and Electronics Engineers
JADE	Java Agent Development Framework.
KQML	Knowledge Query and Manipulation Language.
MAS	Multi-Agent Systems.
MDD	Model Driven Development.
MOF	Meta-Object Facility.
ODM	Ontology Definition Metamodel.
OMG	Object Management Group.
OWL	Ontology Web Language.
OWL-S	OWL for Services.
PIMM	Platform Independent Metamodel.
PRS	Procedural Reasoning System.
RDF	Resource Description Framework.

KISALTMALAR DİZİNİ (devam)Kısaltmalar

RDFS	RDF Schema.
SEA_ML	Semantic web Enabled Agent Modeling Language.
SOAP	Simple Object Access Protocol.
SS	Semantic Service.
SWA	Semantic Web Agent.
SWO	Semantic Web Organization.
SWS	Semantic Web Service.
UML	Unified Modeling Language.
URI	Uniform Resource Identifier.
W3C	World Wide Web Consortium.
WSDL	Web Services Description Language.
WSMO	Web Service Modeling Ontology.
WWW	World Wide Web.
XMI	XML MetaDate Interchange.
XML	eXtensible Markup Language.

1. GİRİŞ

Yazılım etmenleri (*“software agents”*), kullanıcısı adına bulunduğu ortamda bağımsız eylemlerde bulunabilen, çevresel değişimleri fark edip tepkide bulunabilen bilgisayar sistemleridir. Yazılım etmenleri birbiriyle iş birliği yapmak için bir araya gelerek Çok-etmenli Sistemleri (*“Multi-agent Systems”* – MAS) oluştururlar. MAS’ların modellenerek hayata geçirilmesi için Etmen Tabanlı Yazılım Mühendisliği (*“Agent Oriented Software Engineering”* – AOSE) kapsamında çeşitli etmen geliştirme çerçevesi çalışmaları yapılmaktadır. Etmen sistemlerinin temel elemanları etmenin iç yapısı, amacı, rolü ve diğer etmenlerle etkileşimi olarak düşünülebilir. Bu bileşenler nesne tabanlı geliştirme tekniklerine göre farklılıklar gösterirler. MAS araştırmacılarının, üstvarlıkları ve aralarındaki ilişkileri içeren etmen üstmodellerini (*“metamodel”*) tanımlayan çeşitli çalışmaları (Ör. Ferber and Gutknecht, 1998; Hahn et al., 2009; Beydoun et al., 2009) literatürde yer almaktadır. MAS geliştirme çalışmalarının ve var olan MAS geliştirme çerçevelerinin AOSE’de önemli katkıları olsa da yakın gelecekte etmenlerin üzerinde çalışacağı düşünülen Anlamsal Web ortamı ve MAS’ların bu ortam üzerinde çalışabilmesi için ihtiyaç duyulan yapıların bu çalışmalarda yeterince desteklenmediği görülmüştür. İnternet ortamında dağıtık olarak çalışan MAS’ların bilgi paylaşım problemi Anlamsal Web araştırma alanı ile kesişmektedir. Bu yeni nesil web, Dünya Geneli Ağ’ı (*“World Wide Web”* – WWW) web sayfası içeriklerinin ontolojiler kullanılarak yorumlanabileceği bir seviyeye taşımayı hedeflemektedir. Söz konusu yorumlamanın ve anlam çıkarsamaların özerk etmenler tarafından insanlar adına yerine getirileceği açıktır.

Mümkün olduğunca yüksek soyutlama seviyesinde çalışmak MAS geliştirmeyi çoğu zaman daha kolay bir hale getirmektedir. Gerek içsel yapılarının karmaşıklığından gerek sosyal etkileşimlerinden dolayı MAS’ların kod seviyesindeki detaylarını ortaya koymak, MAS alanında uzman olmayan kullanıcılar için oldukça zordur. Bu nedenle model güdümlü etmen yazılımı geliştirme yaklaşımlarının MAS geliştirme açısından önemi etmen araştırmacıları tarafından anlaşılmış (Ör. Bauer et al., 2001; Cervenka et al., 2005) ve bu konu AOSE’nin belli başlı araştırma konularından biri haline gelmiştir.

Günümüzde etmen tabanlı yazılım sistemlerinin model güdümlü geliştirilmesine yönelik çalışmalar olsa (Ör. Beydoun et al., 2009; Bauer et al., 2001; Hahn et al., 2009) da bu çalışmaların önemli bir kısmı sadece tek bir MAS platformu için model güdümlü geliştirme yaklaşımları önermektedir. Çalışmaların genelinde Anlamsal Web ortamı ve bu ortam bileşenleri ile MAS'ların bir arada çalışabilmesi dikkate alınmamıştır. Anlamsal Web ortamı dikkate alınarak geliştirilen yazılım çerçevelerinin eksiği ise ilgili alanda uzman olmayan sistem geliştiricileri tarafından kullanılmalarının oldukça güç olmasıdır. Çoğu zaman sistem ihtiyaçlarını ortaya koyan kişilerin programlamaya özgü yapıları bilmesine gerek yoktur.

Etmen Tabanlı Yazılım Geliştirme (*“Agent Oriented Software Development – AOSD”*) çalışmalarında görülen önemli bir eksiklik, şu anki model güdümlü MAS geliştirme önerilerinden hiçbirisinin Anlamsal Web üzerinde çalışacak etmen sistemlerinin görsel ya da metinsel bir şekilde tanımlanmasını sağlayacak alana özgü bir dile (*“Domain Specific Language” – DSL*) ya da alana özgü bir modelleme diline (*“Domain Specific Modeling Language” – DSML*) ait uygun sözdizime (*“syntax”*) ve anlamsala (*“semantics”*) sahip olmamasıdır. Tez kapsamında, belirtilen bu eksiklikleri tamamlayacak bir DSL ve DSML'nin sözdizimleri oluşturulmuştur. Tezin takip eden bölümleri şu şekilde düzenlenmiştir:

Bu bölüm içerisindeki alt bölümde, tezde kullanılan yaklaşım tanıtılmaktadır. Geliştirilen sözdizim için kullanılan araçlar ile ilgili bilgiler bu bölüm içerisinde verilmektedir.

Tezin ikinci bölümünde, çalışma için gerekli alt yapı anlatılmıştır. Yazılım etmenleri, Anlamsal Web, soyut sözdizim (*“abstract syntax”*) ve somut sözdizim (*“concrete syntax”*) kavramları, alana özgü dil tanımlamaları ve etmen geliştirme çerçeveleri kısaca tanıtılmıştır.

Üçüncü bölümde, ilgili alanda daha önce gerçekleştirilen çalışmalar ve bu çalışmaların tez kapsamında yapılan çalışma ile benzer ve farklı yönleri ortaya konulmuştur.

Dördüncü bölümde, Anlamsal Web ortamında çalışan MAS'lar için geliştirilen bir platform bağımsız üstmodel, farklı bakışaçlarıyla tanıtılmıştır. Tezde sunulan sözdizimin dayandığı bu üstmodel, içerdiği üstmodel elemanları ve bu elemanlar arasındaki ilişkiler ile birlikte ayrıntılı olarak anlatılmıştır.

Beşinci bölümde, geliştirilen dile ait grafiksel somut sözdizim tanıtılmıştır. Görsel sözdizimin oluşturulması sırasında kullanılan görsel modelleme araçları tanıtılmış ve grafiksel somut sözdizimin oluşumu anlatılmıştır.

Altıncı bölümde, metinsel somut sözdizim anlatılmaktadır. Metinsel sözdizimin tanımlanması için kullanılan metinsel modelleme araçları tanıtılmış ve metinsel somut sözdizimin oluşumu anlatılmıştır. Metinsel modelden koda dönüşüm için kullanılan araçların tanıtımı, kullanımı, yazılan kurallar ve koda dönüşüm işlemlerinin nasıl yapılacağı ile ilgili bilgiler bu bölümde detaylandırılmıştır.

Yedinci bölümde, hazırlanmış olan görsel somut sözdizimin değerlendirilmesi yapılmıştır. Örnek bir durum çalışmasıyla modelleme aşamaları gösterilmiştir.

Sekizinci bölümde, hazırlanmış olan metinsel somut sözdizimin değerlendirilmesi yapılmıştır. Örnek bir durum çalışmasıyla metinsel model oluşturma aşamaları uygulanmış ve oluşturulan örnek modelden otomatik kod üretimi sağlanmıştır.

Son bölümde tezin sonucu ve ileriye dönük yapılabilecek çalışmalar hakkında bilgilere yer verilmiştir.

1.1 Yaklaşım

Etmenlerin içsel yapıları ve kendi aralarındaki etkileşimleri oldukça karmaşıktır. Etmen sistemlerinin Anlamsal Web yetenekli olması, yazılım etmenlerinin web içeriğini farklı kaynaklardan elde edebildiği, bilgiyi işleyebildiği ve sonuçları gerektiğinde diğer etmenlere gönderip bilgi alışverişinde

bulunduğu bir ortamda olması anlamını taşır. MAS'lar kapsamında özerk etmenler, anlamsal bilgiyi değerlendirebilirler ve fonksiyonları anlamsal olarak tanımlanmış varlıklar ile beraber çalışabilirler. Karmaşık olan etmen yapısının üzerine Anlamsal Web yeteneğinin eklenmesi, etmen sistemlerini olduğundan daha karmaşık hale getirmektedir. Bu karmaşık yapıları daha basit bir hale getirmenin çözüm yollarından birisi, belirli bir uygulama alanına özgü yapıları ve gösterim biçimleri olan DSL'ler (van Deursen et al, 2000; Mernik et al., 2005; Varanda-Pereira et al., 2008; Fowler, 2011) tanımlamaktır.

DSL kullanıcıları karşılaştıkları problemle ilgili bilgiye sahip olmalıdır ancak bu kullanıcıların çoğu zaman yazılım konusunda deneyimleri yoktur ya da oldukça başlangıç seviyesindedir (Sprinkle et al., 2009). DSL'ler yazılım kodları yerine görsel bir şekilde ya da belli anlamlara sahip temel elemanlarla temsil edildiği için, soyutlama düzeyini, anlamlılığı ve kullanım kolaylığını arttırmaktadır (Schmidt, 2006; Gray et al., 2007).

Görsel modelleme ile zenginleştirilen bir DSL olarak görülebilecek bir DSML, Alana Özgü Modellemede ("*Domain Specific Modeling*" - DSM) kullanılan, Model Güdümlü Geliştirmenin ("*Model Driven Development*" - MDD) önemli bileşenlerinden biridir (Schmidt, 2006). Bir DSML'de grafiksel sözdizim, modelleme yaparken pek çok avantaj sağlar.

DSML geliştirme sürecine genellikle dilin modelinin tanımlanması ile başlanır (Zdun and Strembeck, 2009). Bir başka deyişle, üzerinde çalışılan alandaki kavramlar ve soyutlamalar hedef modele (oluşturulacak dilin modeline) yansıtılacak şekilde tanımlanmalıdır. Sonra, dilin kavramları arasındaki ilişkilerin tanımlanması gerekir. Dilin soyut sözdizimi olarak tanımlanan model genellikle bir üstmodelle tanımlanır.

Bu tez kapsamında ilk olarak bir üstmodel tanımlanmasına yer verilmiştir. Verilen üstmodel 109E125 numaralı TÜBİTAK Araştırma Destek Programları Başkanlığı (ARDEB) tarafından desteklenen bir Ar-Ge projesi kapsamında, ortak bir çalışma sonucunda geliştirilmiştir. Geliştirilen üstmodel, Eclipse Modelleme Ortamı ("*Eclipse Modeling Framework*" – EMF) (Eclipse EMF, 2006) üzerinde

Ecore üstmodeli ("*Ecore metamodel*") ile tasarlanmıştır. EMF, yeni bir model geliştirebilme ve bu model için gerçek zamanlı kod üretebilmeyi sağlayan araçlar sunmaktadır. EMF platformunun sağladığı araçlar ile hazırlanan üstmodel, geleneksel etmen elemanlarını içeren platform bağımsız bir üstmodeldir ("*platform independent metamodel*" - PIMM).

Soyut sözdizimin tanımlanmasından sonra, tez çalışmaları kapsamında grafiksel somut sözdizim ve metinsel somut sözdizim tanımlamaları yapılmıştır. Hem grafiksel hem metinsel sözdizimin tanımlanması kullanıcıya alternatif sunmak amacıyla gerçekleştirilmiştir. Kullanıcı ister görsel bir editör üzerinden sürükle-bırak yöntemiyle ister metinsel bir editor üzerinden kendi modelini tanımlayarak örnek bir MAS oluşturabilir.

Grafiksel somut sözdizim, Eclipse Grafiksel Modelleme Ortamı'nda ("*Eclipse Graphical Modeling Framework*" – GMF) (Eclipse GMF, 2006) gerçekleştirilmiştir. GMF, kullanıcılar için, EMF üzerine dayalı kullanışlı araçlar ve gerçek zamanlı bir grafiksel arayüz geliştirme ortamı sunmaktadır. Kullanıcılar zengin içerikli grafiksel arayüzler içeren GMF ortamını kullanarak kendi modellerini geliştirebilmektedirler.

Metinsel somut sözdizim, Xtext (Xtext, 2009) ile gerçekleştirilmiştir. Xtext, DSL oluşturmak için kullanılan bir dil geliştirme çerçevesidir. Bir geliştiricinin kendi modelleme dilini metinsel olarak tanımlamasını sağlayan tekrar kullanılabilir kütüphaneler topluluğu olarak düşünülebilir. Xtext, BNF ("*Backus-Naur Form*")'in genişletilmiş şekli olan EBNF ("*Extended BNF*") kullanarak bir DSL/DSML'in özelliklerinin belirlenmesini sağlar. Oluşturulan dildeki anahtar kelimelerin neler içereceğine, hangi bileşenlerden oluşacağına ya da bu bileşenlerin ne tür yapılar döndüreceğine kullanıcının karar vermesine olanak sağlar. Ayrıca, Xtext, EMF ile uyumludur. Somut sözdizimlerin oluşturulmasından sonra, her iki sözdizimden gerçek MAS platformları üzerinde çalıştırılacak yazılım kodlarının üretimi gerçekleştirilmiştir.

2. ALT YAPI

Bu bölümde tez çalışması için gereken alt yapı bilgileri yer almaktadır. Bu bölümün alt bölümlerinde yazılım etmenleri, Anlamsal Web, Anlamsal Web servisleri, alana özgü dil kavramı ve alana özgü dil tanımlamak için gerekli soyut ve somut sözdizimler anlatılacak ve son olarak etmen geliştirme çerçevelerinden bahsedilecektir.

2.1 Yazılım Etmenleri

Yazılım etmenleri kavramı için bütün araştırmacıların anlaştığı tek bir tanım yoktur. Ancak, etmenleri diğer sistemlerden ayıran özelliklerini belirterek şöyle bir tanım yapmak mümkündür: Etmenler, algılayıcıları (*“sensor”*) yardımıyla ortamı algılayan ve etkileyicileri (*“effector”*) yardımıyla ortamı etkileyen, özerk (*“autonomous”*) bilgisayar sistemleridir (Russel and Norvig, 2003). Yazılım etmenleri kullanıcıları adına birtakım görevleri yerine getirmekle yükümlüdür. Tek bir etmenin kendi bilgi ve bireysel yeteneklerini kullanarak çözemediği veya etkin bir biçimde çözemeyeceğini düşündüğü problemleri, birbirleriyle işbirliği yaparak eşgüdümlü bir biçimde çözmek için bir araya gelen etmenlerin oluşturduğu sistemler de MAS adını almaktadır (Weiss, 1999). Yazılım etmenleri ve MAS’lar, bilgi ve iletişim teknolojisi iş senaryolarının çeşitli alanlarda uygulanmasıyla birlikte ortaya çıkan karmaşıklığın giderilmesi amacıyla kullanılmaktadır (Kardaş, 2008). Bir etmene ilişkin özellikler, özerklik (*“autonomy”*), karşıt-eylemlilik (*“reactivity”*), amaç-yönelimlilik (*“goal-orientedness”*), sosyal-yetenek (*“social ability”*) ve kalıcı süreklilik (*“permanent continuity”*) olarak düşünülebilir (Erdur, 2001).

Yazılım etmenleri için önerilen mimariler, etmenlerin içsel yapısı ve işleyişi ile ilgili olarak çeşitlilik göstermektedir. Güncel olan iki önemli etmen mimarisinden ilki karşıt eylemlilik etmen mimarisidir (*“reactive architecture”*). Karşıt eylemlilik, bir etmenin sürekli olarak bulunduğu ortamı ve çevresindeki değişiklikleri algılayarak “etki” olarak tabir edeceğimiz bu değişimlere önceden tanımlanmış planlardan uygun olanı seçip uygulayarak karşılık vermesi, bir başka deyişle “tepki”de bulunmasıdır. İkinci önemli etmen mimarisi ise Kanı-İstek-

Hedef (*“Belief-Desire-Intention”* – BDI) mimarisidir. BDI, etmen mimarilerinin en önemlisi ve en çok örneğe sahip olanıdır. BDI ilk olarak Bratman (Bratman, 1987) tarafından önerilen ve daha sonra Rao ve Georgeff (Rao and Georgeff, 1995) tarafından genişletilen bir etmen mimarisidir. Bu mimariye sahip bir etmen hangi amaçları (*“goal”*) elde etmek istediğine ve amaçlarına ulaşmak için nasıl hareket etmesi gerektiğine karar vermektedir. Kanılar bir etmenin içinde bulunduğu ortam ile ilgili bildiklerini temsil etmektedir. İstekler ise bir etmenin elde etmek istediklerine karşılık gelmektedir. Son olarak hedefler etmenin üzerinde düşünülmüş tavırları (*“deliberative attitudes”*) olup etmenin amaçlarına ulaşmak için uygulamaya koyacağı planlama mekanizmasını içermektedir. BDI mimarisi etmenlerin çevrelerine dair bilgiler tutmasına, bu bilgiler doğrultusunda tasarım hedeflerini gerçekleştirebilmek için planlarını (*“plan”*) oluşturmalarına izin veren bir mimaridir. BDI mimarisi karşıt eylemlilik mimarisinden farklı olarak sadece etkiye tepki vermekle kalmaz, çıkarsamayı da (*“inference”*) kapsar. En çok bilinen BDI mimarisi prosedürel çıkarsama sistemi (*“procedural reasoning system”* – PRS) (Georgeff and Lansky, 1987) dir. Bu mimari kanılar, istekler, hedefler ve planlar olmak üzere dört temel veri yapısına dayanır (Bellifemine et al., 2007). Tez kapsamında tanıtılan sözdizim, hem karşıt-eylemli etmen mimarisine hem de BDI mimarisine sahip etmenlerin tasarım ve çalışma zamanı bileşenlerinin modellenmesi için gerekli elemanları ve ilişkileri içermektedir.

Etmenlerin birbirleriyle etkileşim halinde olabilmesi ve birbirlerinin dilinden anlayabilmesi için bazı standartlara ihtiyaç vardır. Institute of Electrical and Electronics Engineers (IEEE) Foundation for Intelligent Physical Agents (FIPA) (FIPA, 2002a) etmen tabanlı bilgi işlem alanında standartları belirleyen kurumların başında gelir. FIPA, etmenler ve etmen tabanlı yazılım sistemleri arasındaki etkileşimi desteklemek amacıyla 1996 yılında kurulmuş, kar amacı olmayan bir kuruluştur. Etmen iletişim dilleri, içerik dilleri, etkileşim protokolleri, etmen yönetimi ve etmen mesajlaşma alt yapıları konularında standartlar geliştirmiştir.

Tez kapsamında tanıtılan sözdizim, çok-etmenli sistemlerdeki etkileşimlerin ve mesajlaşmaların soyut bir gösterimini sağlamaktadır. Örneğin

“Contract Net Protocol”ünü (Smith, 1980) uygulayan etmenlerin etkileşimleri soyut bir şekilde önerilen üstmodel ile gösterilebilir. Benzer şekilde FIPA’nın Etmen İletişim Dili (“*Agent Communication Language*” – ACL) (FIPA, 2002a) mesajları ve mesaj tipleri de yine burada tanımlanan varlıklar ile üst düzeyde gösterilebilir.

2.2 Anlamsal Web

Anlamsal Web (Berners-Lee et al., 2001), temel amacı iyi tanımlanmış ve bağlantılandırılmış olan bilgilerin ve servislerin web ortamında bilgisayarlar tarafından kolay bir şekilde okunup anlaşılabilmesini sağlayacak, bugünkü webin bir uzantısı olarak gelişmekte olan bir ağıdır. Anlamsal Web yeni ve ayrı bir web değil, bilgilere iyi tanımlanmış anlamların verildiği, bilgisayarların ve insanların birlikte çalışmalarına imkan veren yeni nesil bir web olarak düşünülmelidir. Bugünkü webin içeriğine temel oluşturan yapılara bakıldığında, bu yapıların insanların okuması, anlaması ve kullanımı için tasarlanmış olduğu görülür. Anlamsal Web, veri ve bilgilerin otomatik olarak işlenebilirliğinden çok insanlar için doküman sağlayan bir medya olacak şekilde geliştirilmiştir. Mevcut web ortamındaki içeriğin bilgisayarlarca okunur ve anlaşılabilir olması için Anlamsal Web ihtiyacı vardır. Böylelikle web, hem insanlar tarafından okunabilecek hem de makineler tarafından anlaşılacaktır.

Anlamsal Webin kullanılabilir olması için, bilgisayarların yapılandırılmış bilgi koleksiyonlarına ve otomatik akıl yürütmeyi (“*automated reasoning*”) sağlayacak çıkarsama kuralları kümelerine erişimleri gerekmektedir (Kardaş, 2008). Çıkarsamaların yapılabilmesi için, Anlamsal Webin hem veriyi hem de veri hakkında yapılacak çıkarsamaları sağlayan kuralların ifade edildiği bir dil özelliklerini sağlaması gerekmektedir. Böylece, HyperText Markup Language (HTML) ve eXtensible Markup Language (XML) dilleri üzerine Anlamsal Webi destekleyecek Kaynak Tanımlama Çerçevesi (“*Resource Description Framework*” – RDF) ve RDF Schema’nın (RDFS) teknolojilerine olan ihtiyaç ortaya çıkmaktadır. RDF (W3C, 2004) web ortamındaki kaynakların, kaynak özelliklerinin ve özellik değerlerinin tanımlanmasını sağlamaktadır. RDF ile her kaynağın bir Tekbiçimli Kaynak Tanımlayıcısı’na (“*Uniform Resource*

Identifier” – URI) sahip olması ve özne-yüklem-nesne (“*subject-predicate-object*”) üçlünün yer alması, anlamın ifade edilmesini sağlamaktadır. RDFS ise RDF için bir sistemi tanımlar. Kaynakların sınıflar, özellikler ve değerler ile tanımlanmasını sağlar. İçerdiği sınıf kavramı nesne tabanlı programlama dilindeki sınıf kavramı ile benzerlik gösterir.

Anlamsal Webde karşılaşılan bir diğer temel kavram ontolojilerdir. Ontoloji, kavramlar arasındaki ilişkileri biçimsel (“*formal*”) olarak içeren bir dokümandır. Ontoloji dokümanlarının hazırlanmasında yukarıda bahsedilen RDF ve RDFS teknolojileri kullanılabilir. Bir kavramın farklı çevrelerde farklı anlamlara sahip olması genellikle kavram kargaşalarına neden olmaktadır. Ontolojilerde, aynı kavramı ifade eden farklı tanımlayıcılar belirlenmektedir. Ontolojiler, belirli bir alana özgü bilgilerin herkes tarafından kabul edilen genel bir anlamının oluşmasına imkan verir. Ontolojiler, herhangi bir alanda standart olarak kullanılacak ortak ve paylaşılan sözcük kümelerini veya terminolojiyi belirler.

Ontoloji Web Dili (“*Ontology Web Language*” – OWL), ontolojilerin tanımlanması ve geliştirilmesinde Dünya Çapında Ağ Birliği’nin (“*World Wide Web Consortium*” – W3C) standart dilidir. OWL, isimlendirme için URI’yi, web için yapılan tanımlamalarda da RDF’yi kullanır. OWL, RDF ve RDFS üzerine kurulmuştur ve özellikleri (“*properties*”), sınıfları (“*classes*”), sınıflar arasındaki ilişkileri, niceliği (“*cardinality*”) ve denkliği (“*equality*”) tanımlamak için daha çok kelime haznesine sahiptir.

2.2.1 Anlamsal Web servisleri

Web servisleri, bir URI ile tanımlanan, herkesin kullanımına açık arayüzlere ve bağlantılara sahip olan yazılım sistemleridir (Sycara et al., 2003a). Web servislerinin arayüzleri ve bağlantıları XML ile tanımlanmıştır. Bugün kullanılmakta olan web servisleri, kendilerini temsil eden ve Web Servis Tanımlama Dili (“*Web Services Description Language*” – WSDL) kullanılarak hazırlanan arayüzleri sayesinde, geliştirildikleri yazılım dili ve ortamına bağlı kalmaksızın çeşitli ortamlarda çalışan istemci yazılımlar tarafından

kullanılabilmektedir. İstemci yazılım, WSDL'yi işleyerek sunulan servise ait işlemi ve bu işlem için gerekli parametreleri öğrenir ve servisi web servisleri için bir standart olan basit nesne erişim protokolü ("*simple object access protocol*" – SOAP) kullanarak çalıştırabilir. Kullanılan bu yapı, sözdizimsel birlikte işlerliği ("*interoperability*") sağlasa da anlamsal birlikte işlerliği ya da web servislerin otomatik tümleşimini ("*composition*") sağlamamaktadır (Sycara et al., 2003a). Anlamsal birlikte işlerliği sağlamak amacıyla, web servislerin yeteneklerinin servis ontolojilerinde tutulması ve böylece bu ontolojiler yardımıyla ihtiyaca uygun olan servislerin bulunarak dinamik çağrının gerçekleştirilmesi üzerine çalışmalar yapılmaktadır.

OWL Servisleri ("*OWL for Services*" – OWL-S) (OWL-S Coalition, 2004) ve Web Servis Modelleme Ontolojisi ("*Web Service Modeling Ontology*" – WSMO) (WSMO Working Group, 2005) bahsedilen web servis tanımlama ontolojilerine örnek oluşturmaktadır. Tanımlamaları servis ontolojilerine uygun olacak biçimde verilmiş web servisleri, Anlamsal Web servisleri olarak adlandırılmaktadır. Bu tezde ortaya konan DSL ve DSML'nin MAS'lar için DSL/DSML geliştirme amacıyla daha önceden yapılmış çalışmalardan temel farklarından biri de Anlamsal Web servislerinin etmenlerle etkileşimini destekliyor olmasıdır.

Günümüzde Anlamsal Web servislerini modellemek için en çok kullanılan ontoloji OWL-S'dir. OWL-S, bir servisi genellikle üç anlamsal doküman ile tanımlamaktadır. Servis profili ("*service profile*"), bir servisin yeteneklerinin neler olduğunu gösterir ve bir servis keşfetmek için gerekli bilgileri sağlar. Servis süreç modeli ("*process model*"), servisin iç birleşimini ve servis çalıştırma dinamiklerini tanımlamaktadır. Son olarak servis zemini ("*service grounding*") ise, bir servisin çalışma protokolünü, ilgili servislerle nasıl etkileşimde bulunulabileceğini tanımlamaktadır. OWL-S'de bulunan servis sınıfı, tanımlanan web servis için yapısal bir referans belirtmektedir.

2.3 Alana Özgü Dil

Sistem gereksinimlerinin karşılanması amacıyla kullanılan terminolojinin kavramlarından oluşturulan bir DSL, son kullanıcı konumundaki programcılarının ve alan (“*domain*”) uzmanlarının bir problemin özünü belli bir alana özgü problem uzayına yakın olan soyutlamalarla tanımlamasına olanak vermektedir. Bu tip alana özgü dillerin oluşturulmasındaki en önemli zorluk bu dillerin söz dizimlerinin ve anlamsallığının ifade edilmesinde yaşanmaktadır.

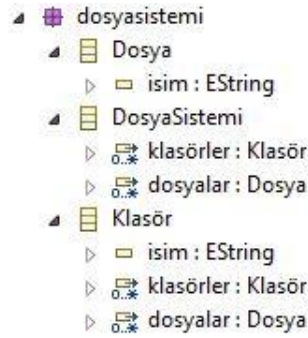
Bir alana özgü dil tanımlamak için yapılması gereken ilk iş, o alandaki elemanların ve bu elemanlar arasındaki ilişkilerin belirlendiği soyut sözdizimi oluşturmaktır. Soyut sözdizim oluşturulduktan sonra, bir üst seviyeye geçerek kullanıcının dili kullanması için tanımlanacak bir somut sözdizim oluşturulmalıdır. Somut sözdizim, isteğe bağlı olarak grafiksel ya da metinsel olarak gerçekleştirilebilir. Tez kapsamında hem grafiksel somut sözdizim, hem de metinsel somut söz dizim tanımlamalarına yer verilmiştir.

2.3.1 Soyut sözdizimi

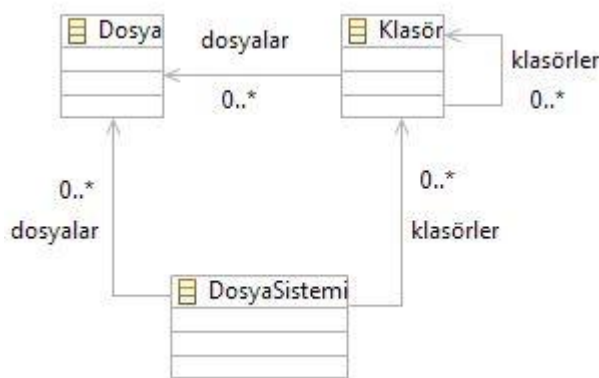
Bir dilin soyut sözdizimi dildeki kavramlara ait bir sözlüktür ve kavramların modeller ve programlar oluşturmak için nasıl birleştirilmesi gerektiğini tarif etmektedir. Sözdizim bir programlama dilindeki kavramların sunum ve anlamlarını gözetmeksizin sadece şekilleri ve yapısı ile ilgilenmektedir. MDD açısından bakıldığında soyut sözdizim modellerin nasıl görüldüğünü tanımlayan bir üstmodel ile ifade edilmektedir. Bu tez kapsamında tanıtılan üstmodel, 109E125 numaralı TÜBİTAK ARDEB destekli Ar-Ge projesi kapsamında geliştirilmiştir. Geliştirilen üstmodel, özellikle etmenler ile Anlamsal Web servisleri arasındaki etkileşimin uygun bir şekilde modellenmesini sağlayacak bir şekilde oluşturulmuştur. Önceden de bahsedildiği gibi, üstmodel geliştirmeleri için EMF kullanılmıştır.

2.3.2 Grafiksel somut sözdizimi geliştirimi için kullanılan araçlar

Tez kapsamında kullanılan Eclipse Modelleme Ortamı, XML MetaData Interchange (XMI) standartlarında dosyalama yapısı kullanan, kullanıcı desteği devam eden ve birçok kullanışlı araca sahip açık kaynak kodlu bir yazılım geliştirme ortamıdır. EMF, kullanıcıların kendi üstmodellerini geliştirebilmeleri için Ecore adında bir üstmodel sunmaktadır. Ecore üstmodeli ile geliştirilen üstmodeller “.ecore” uzantılı dosyalarda tanımlanmaktadır. Örnek olması açısından, Şekil 2.1’de Ecore üstmodeli ile EMF ortamında geliştirilen bir üstmodel tanımı verilmiştir. Dosya, dosya sistemi ve klasör üst varlıkları ve bunlar arasındaki ilişkiler Şekil 2.1’de gösterilmektedir. Eclipse Modelleme Ortamı’nda geliştirilen modellerin diyagram görüntüleri otomatik olarak elde edilebilmektedir. Şekil 2.2’de ise Ecore üstmodelinden oluşturulmuş diyagram görüntüsü verilmiştir. EMF, hem diyagram üzerinde hem de XML benzeri bir editör üzerinde modeller geliştirmeye olanak vermektedir.



Şekil 2.1. Ecore üstmodeli ile geliştirilen örnek üstmodel.



Şekil 2.2. Ecore üstmodelinden oluşturulmuş diyagram görüntüsü.

Geliştirilen üstmodel için grafiksel bir arayüzün hazırlanması ve tanımlanan üstmodel doğrultusunda yeni örnek modeller gerçekleştirecek kullanıcılar için modelleme ortamının daha kullanışlı olması amacıyla Eclipse ortamında GMF tanımlanmaktadır. Şekil 2.3'te, yukarıda verilen üstmodelden oluşturulmuş grafiksel bir arayüz verilmektedir.



Şekil 2.3. GMF ile geliştirilmiş örnek bir model.

2.3.3 Metinsel somut sözdizim geliştirimi için kullanılan araçlar

Soyut sözdizimin tamamlanmasından sonra, metinsel gösterim için Xtext (Xtext, 2009) tercih edilmiştir. Xtext ilk olarak bağımsız bir dil geliştirme çerçevesi olarak ortaya çıkmış olsa da, 2009 yılından itibaren Eclipse eklentisi olarak kullanılmaktadır. Xtext, DSL oluşturmak için kullanılan bir dil geliştirme çerçevesidir. BNF'nin uzatılmış şekli olan EBNF kullanarak bir DSL/DSML'nin özelliklerinin belirlenmesini sağlar. Oluşturulan dildeki anahtar kelimelerin neler içereceğine, hangi bileşenlerden oluşacağına ya da bu bileşenlerin ne tür yapılar döndüreceğine kullanıcının karar vermesine olanak tanır.

Xtext'in sunduğu yetenekler; otomatik tamamlama ("*auto completion*"), sözdizimi renklendirme ("*syntax coloring*"), yeniden adlandırma ("*rename refactoring*"), parantez eşleme ("*bracket matching*"), otomatik düzeltme ("*auto edit*"), modelin anlamsal yapısını gösteren taslak gösterim ("*outline view*") ve kullanıcının daha düzenli şekilde kod yazmasını sağlamak amacıyla kullanılan kod formatlama ("*code formatting*") olarak sıralanabilir.

Metinsel somut sözdizimden kod üretimi Xpand (Xpand, 2009) aracı ile gerçekleştirilmiştir. Xpand EMF modellerini kullanarak metinsel çıktılar oluşturulmasına olanak sağlar. Xpand ile modelden metine dönüşüm yapabilmek için EMF ile tanımlanmış bir üstmodele ve dönüşümü sağlamak amacıyla kullanılan en az bir taslak ("*template*") dosyasına ihtiyaç vardır. Xpand aracı Eclipse eklentisi olarak kullanılmaktadır.

2.4 Etmen Geliştirme Çerçeveleri

Yazılım etmenlerini hayata geçirebilmek için çeşitli etmen geliştirme çerçeveleri bulunmaktadır. Bu etmen geliştirme çerçevelerinden en çok kullanılanlar JADE (Bellifemine et al., 2001), JACK (Howden et al., 2001) ve JADEX (Pokahr et al., 2005)'tir.

Java Agent DEvelopment Framework (JADE), karşıt eylemli etmen mimarisine dayanan, FIPA standartlarına uyan bir yazılım etmeni platformudur. Amacı, kapsamlı bir servis ve etmen kümesi aracılığıyla standart uyumluluğu sağlayarak etmen yazılımları geliştirmeyi kolaylaştırmaktır. Tamamen Java programlama dili ile geliştirilmiştir ve açık kaynak kodludur. JADE, Farklı makineler üzerinde bir arada çalışabilecek dağıtık bir etmen platformu olarak kullanılabilir. BDI mimarisini desteklemiyor olması sebebiyle bu tez çalışmasında hedef platform olarak seçilmemiştir.

JACK, BDI mimarisine dayanan çok etmenli bir sistem platformudur. Bu platformda, etmenlerin planları olaylar tarafından tetiklenir ve mesajlar alt-olaylar olarak algılanır. Bir JACK programı; etmenler, yetenekler, planlar, olaylar ve kanı kümeleri bileşenlerinden oluşur. JACK, savunma sistemlerinden endüstriye kadar birçok uygulama alanında kullanılmaktadır. BDI mimarisini desteklediği halde ticari bir etmen yazılım platformu olması nedeniyle bu tez kapsamında hedef platform olarak seçilmemiştir.

JADEX, hedef yönelimli etmenlerin, BDI mimarisine uygun olarak oluşturulmasına ve yönetilmesine izin veren bir etmen geliştirme platformudur. Bir JADEX etmeni BDI özelliklerinin tanımlandığı ve XML kullanılarak

hazırlanan bir Etmen Tanımlama Dosyası'ndan (*“Agent Definition File”* – ADF) ve Java programlama dili kullanılarak geliştirilen planlardan oluşmaktadır (Afşar ve ark., 2011).

JADEX mimarisinde, dışarıdan gelen mesajlar ya da hedef tabanlı olaylar, tepki (*“reaction”*) ve müzakere (*“deliberation”*) mekanizmasına bir girdi olarak servis edilir ve böylece çıkarsama işlemi, plan kütüphanesinden uygun bir plan seçilmesi ve hedeflerin tekrar belirlenmesiyle gerçekleşmiş olur. Plan, kanı ve amaç bileşenleri yetenek (*“capability”*) denilen yeniden kullanılabilir bir modulün içinde gruplandırılmıştır. Bir yeteneğin farklı etmenler tarafından yeniden kullanılabilirliğini sağlamak amacıyla oluşturulan bir Yetenek Tanımlama Dosyası (*“Capability Definition File”* – CDF) bulunmaktadır.

Kanı, etmenin kendisi ve çevresi hakkında sahip olduğu bilgileri temsil eder. Kanılar herhangi bir Java nesnesi olabilir ve kanı-tabanında (*“beliefbase”*) depolanmaktadırlar. Amaç, etmenlerin gerçekleştirmek istedikleri amaçları temsil eder. Amaçlar, JADEX'in çıkarsama sürecinde güncellenebilir, değişebilir veya yenileri eklenebilir. Plan, JADEX platformunda bulunan etmenlerin amaçlarını gerçekleştirmek için gerekli olan eylemleri temsil eder. Planlar iki bölümden oluşur. Etmenlerin amaçlarını gerçekleştirmek için gerekli olan eylemler, planın gövdesini oluşturan Java kodlarıdır. Diğer bölüm ise, ADF'de tanımlanan plan başlığıdır. Yetenek; kanı, hedef ve planların paketlenmiş hali şeklinde, kendi muhakeme süreci olmayan bir etmen olarak düşünülebilir. Bir etmen, en az bir yeteneğe sahip olmalıdır.

JADEX platformunun BDI mimarisini desteklemesi, etmen çalışmalarında çok kullanılan platformlardan biri olması ve açık kaynak kodlu bir kütüphane sağlamasından dolayı tezdeki kod üretim aşamasında hedef platform olarak seçilmiştir.

3. İLGİLİ ÇALIŞMALAR

AOSD’de MAS’ların model güdümlü geliştirilme çalışmaları en önemli araştıma konularından biri olduğu için, etmen kullanıcılarının daha kullanışlı bir şekilde model geliştirebilmeleri amacıyla birçok etmen üstmodelleri ve etmen modelleme dilleri oluşturulmuştur. Birçok DSL ve DSML’de de bu üstmodeller kullanılmıştır. Bu nedenle, bu bölümde ilgili çalışmalar iki alt bölüme ayrılarak incelenmiştir. Birinci bölümde, MAS’lar için geliştirilmiş üstmodeller ve modelleme dili çalışmalarına, ikinci bölümde ise MAS’lar için geliştirilmiş DSL ve DSML’lere yer verilmiştir.

3.1 Etmen ve MAS Üstmodelleri ve Modelleme Dili Çalışmaları

MAS’lar için genel amaçlı olarak geliştirilmiş ilk üstmodelin AALAADIN (Ferber and Gutknecht, 1998) olduğu söylenebilir. Geliştirilen bu üstmodel, MAS yapısını etmen, grup ve rol ana kavramları ve bu kavramlar arasındaki ilişkiler ile ifade etmektedir.

Diğer bir yandan, AOSD araştırmalarının çoğunda, genel amaçlardan çok üzerinde çalışılan metodolojinin kavramlarının biçimsel bir şekilde tanıtılması hedeflenmiştir. Örneğin (Bernon et al., 2005)’de ADELFE (Bernon et al., 2003), GAIA (Zambonelli et al., 2003) ve PASSI (Cossentino and Potts, 2002) MAS metodolojilerinin üstmodelleri tanıtılmıştır.

Benzer bir şekilde (Molesini et al., 2005)’de, SODA metodolojisi (Omicini, 2000) için bir üstmodel tanıtılmıştır. Bu çalışmada, SODA’nın etkileşimleri ve sosyal bakışaçıları modellenmiş ve bu bakışaçıları göz önünde bulundurarak bir üstmodel tanımlanmıştır. Bu çalışmalar incelendiğinde, tanımlanan üstmodel kavramlarının ve kavramlar arasındaki ilişkilerin formal gösterimlerinin yapıldığı görülmektedir. Metodolojilerin açıkça analiz edilebilmeleri ve uygun bir şekilde genişletilebilmeleri için bu formal gösterimler oldukça önemli ve yararlıdır. Bunun yanı sıra, üstmodeller etmen topluluklarında farklı metodolojileri karşılaştırmaya yardımcı birer araç olarak da kullanılabilir. Ancak, bu üstmodellerin genel amaçlı MAS modellemeleri için uygun olmadığı da gözden

kaçmamalıdır. Çünkü, bu üstmodellerle oluşturulmak istenen modeller, ilişkili metodolojilere özel olarak hazırlanmaktadır ve çoğu zaman sadece belli çerçevelerde uygulanması mümkündür.

IEEE FIPA Modelleme Teknik Komitesi ve Nesne Yönetim Grubu (*“Object Management Group”* – OMG) Etmen Özel İlgi Grubu bir MAS bünyesinde yer alan etmenler, etmen rolleri ve etmen grupları arasındaki ilişkileri belirtmek amacıyla geçmişte bir çalışma yürütmüşlerdir. Bu çalışmanın ürünü olarak FIPA Etmen Sınıf Üstyapısı Üstmodeli (*“Agent Class Superstructure Metamodel”* – ACSM) (Odell et al., 2005) adı verilen bir etmen üstmodeli ortaya konmuştur. Etmen organizasyonlarının biçimsel bir gösterimini sunan bu üstmodel hem Birleşik Modelleme Dili (*“Unified Modeling Language”* – UML) (OMG, 2000) üstyapısına dayanır hem de bu üstyapının bir uzantısıdır. Ancak geliştiricilerinin de belirttiği gibi ACSM sadece sekiz adet üst düzey varlık ve bunların ilişkisini tanımlaması ile oldukça yüksek bir soyutlama seviyesine sahiptir. Özellikle etmen iç davranışlarının ve etmen iletişimlerinin modellenmesi için kullanışlı değildir ve uzantılara ihtiyaç duymaktadır. Örneğin (Kardaş et al., 2007)’de, Anlamsal Web yetenekli MAS’lar için FIPA ACSM’yi genişleten bir üstmodel tanıtılmıştır. 2009 yılında yayınlanan (Kardaş et al., 2009) çalışmasında ise model güdümlü bir MAS yazılımı geliştirme süreci önerilmiştir. MAS geliştirme sırasında Anlamsal Web yapılarını ve bunlarla geleneksel etmen sistemi bileşenleri arasındaki etkileşimleri destekleyen yeni bir etmen yazılımı geliştirme süreci tanıtılmaktadır. Ancak Anlamsal Web yetenekli MAS’lar için bir DSML’nin formal gösterimi ve bu DSML’nin model güdümlü MAS geliştirme için kullanılması üzerine bir çalışma henüz mevcut değildir.

Bir diğer önemli MAS üstmodeli (Hahn et al., 2009)’da tanıtılmıştır. Bu üstmodelde, etmen modelleme kavramları yedi MAS bakış açısında toplanmıştır. Bu bakış açıları, çoklu etmenler (*“multi agents”*), etmen, davranışsal (*“behavioural”*), organizasyon (*“organization”*), rol (*“role”*), etkileşim (*“interaction”*) ve çevre (*“environment”*) olarak tanımlanmıştır. Hahn ve arkadaşları, (Kardaş et al., 2009)’da da olduğu gibi, bu etmen üstmodelini, etmen sistemleri geliştirmek için ve çalıştırılabilir MAS’lar ortaya çıkartmak için bir PIMM olarak tanımlamışlardır.

(Beydoun et al., 2009)'da tanıtılan FAML üstmodeli, etmen sistemler için geliştirilmiş çeşitli üstmodellerin bir sentezi olarak düşünülebilir. Bu çalışmada, MAS'lar için tasarım ve çalışma zamanı kavramları verilmiş ve bu kavramların doğrulanması çeşitli MAS geliştirme metodolojilerinin kullanımı ile gerçekleştirilmiştir.

MAS yazılım mimarisi ilgili MAS'ların uygun şekilde modellenmesine dayanır. Bu nedenle, etmen araştırmacıları, MAS'lar için geliştirilen üstmodel çalışmalarının yanı sıra, MAS modelleme dillerine de çok önem vermektedirler. UML yaygın olarak kabul edilen bir yazılım modelleme dili olduğu için, birçok önemli MAS modelleme çalışması UML temeline ya da UML'nin etmen tabanlı uzantılarına dayanmaktadır. Örneğin, (Depke et al., 2001)'de UML gösterimine dayalı bir etmen tabanlı modelleme tekniği tanıtılmıştır. Burada sunulan modelleme tekniği sistem ihtiyaçlarını tanımlamada ve analizinde kullanılarak etmen birlikte çalışılabilirliğini ve özerkliğini modellemeyi hedeflemektedir.

Etmen UML (*“Agent UML”* – AUML), (Bauer et al., 2001) etmen araştırmacılarının sıklıkla kullandıkları modelleme dillerinden bir tanesidir. AUML, UML'in paket, şablon, sıra diyagramları, etkileşim diyagramları, aktivite diyagramları ve sınıf diyagramlarına etmen temelli uzantılar getirmiştir. AUML, MAS araştırmalarında bir dönem oldukça popüler hale gelmesine rağmen, deneyimler AUML'nin UML uzantılarında bazı eksikliklerin olduğunu göstermiştir. Öncelikle, AUML, nesne tabanlı sistem spesifikasyonu için önerilen UML üzerine çok fazla dayanmaktadır. Bu bağımlılık, AUML'nin nesne tabanlı düşünceden yeteri kadar soyutlanamamasına sebep olmaktadır. (Huget, 2005)'te yer alan çalışmada, AUML'nin görsel gösteriminin eksik olduğu, nesne tabanlı yapılara bağımlı olduğu, diğer geliştiricilerle bilgi alışverişini sağlayacak metinsel gösteriminin bulunmadığı, anlam bilgisinin yarı biçimsel olduğu, ve son olarak sunduğu tanımlamaları kullanma adına hiçbir aracın mevcut olmadığı belirtilmiştir.

Etmen Modelleme Dili (*“Agent Modeling Language”* – AML) (Cervenka et al., 2005), MAS'lar için oluşturulmuş bir diğer genel modelleme dilidir. UML 2.0 üstyapısının (*“superstructure”*) (OMG, 2004) UML'nin soyut sözdizim, anlam ve

gösterimini tanımlamasından hareketle UML dili katmanı, AML'nin MAS'lara özel modelleme yapılarını tanımlamasına izin verir. (Huget, 2005)'te belirtildiği gibi AML, AUML'ye göre daha yeni bir çalışmadır. Ancak onun da metinsel gösteriminin eksik olduğu ve şu an için herhangi bir geliştirme aracı tarafından desteklenmediği görülmektedir.

3.2 MAS DSL ve DSML Çalışmaları

Etmen modelleme ve model tabanlı MAS geliştirme için yapılmış pek çok çalışma olmasına rağmen, etmenler için yapılmış DSML çalışmaları kısıtlı sayıdadır ve genellikle başlangıç seviyesinde kalmıştır. Örneğin, (Kulesza et al., 2005)'te Agent_DSL adında bir DSL tanıtılmıştır. DSL, bir etmenin görevlerini yerine getirmesi için sahip olduğu özellikleri tanımlamayı hedeflemektedir. Ancak çalışmada sadece DSL'nin üstmodeline ve etmen sistemlerinin özelliklerine göre görsel modellenmesine yer verilmiştir. Benzer şekilde (Rougemaille et al., 2007)'de DSML olarak tanımladıkları özelleşmiş iki modelleme dilini tanıtmışlardır. Söz konusu diller yine (Rougemaille et al., 2007)'de tanıtılan özel bir iş alanına ait ana kavramları ve ilişkilerini gösteren üstmodeller ile tanımlanmışlardır. Ancak çalışma sadece ilgili DSML'nin soyut sözdizimini içermektedir ve ne somut dil yapısını ne de DSML dil anlamını tanımlamaktadır. Aslında çalışma bu hali ile model güdümlü MAS geliştirme için sadece genel bir üstmodel tanımlama görünümündedir.

Alman Yapay Zeka Araştırma Merkezi'nde ("*German Research Centre for Artificial Intelligence*" – DFKI) görev alan bir grup etmen araştırmacısının bütün bir MAS DSML'ini tanımladığı görülmektedir. Söz konusu DSML (Hahn, 2008)'de tanıtılmıştır. DSML'ye ait somut sözdizim ve formal anlam ("*formal semantics*") ise sırasıyla (Warwas and Hahn, 2008) ve (Hahn and Fischer, 2009) yer almaktadır. Yakın zamanda ortaya konan bu DSML'ye ait soyut sözdizim bir MAS'ın farklı bakışaçılarını (örneğin MAS, etmen, organizasyon ve rol gibi) dikkate alan ve genel etmen üstvarlıklarını içeren platform bağımsız bir üstmodelden türetilmiştir. Somut sözdizimi ortaya koymak için alan kavramlarının grafiksel notasyonu ve bunun soyut sözdizim elemanları ile olan bağlantıları verilmiştir. Dilin anlamı da ayrıca biçimlendirilmiştir. DFKI etmen

araştırmacılarının bu çalışması tüm tanımlamaları ile bütün bir etmen DSML'sini sunduğundan dolayı önemlidir. Ancak bu DSML de ne Anlamsal Web ortamındaki etmenleri ne de bu etmenler ile Anlamsal Web servisi gibi diğer ortam bileşenleri arasındaki etkileşimleri desteklemektedir.

(Hahn et al., 2008) çalışmasında, etmenlerin Anlamsal Web servislerle bütünleştirilmesi yaklaşımı ele alınmıştır. (Hahn, 2008)'de tanımlanan üstmodelle ek olarak Anlamsal Web servislerin kullanıldığı yeni bir platform bağımsız üstmodel önerilmektedir. Anlamsal Web servislerin entegrasyonunu sağlamak için, MAS üstmodelinin yeni üstvarlıklarla ve ilişkilerle genişletilmesi sonucu her iki üstmodel arasında bir ilişki kurulmuştur. Bu tezin dayandığı üstmodel, farklı iki üstmodel kullanımı yerine, etmen modellenmesi ve etmen-Anlamsal Web servis etkileşiminin modellenmesine olanak tanıyacak şekilde farklı bakış açıları içermektedir ve söz konusu etkileşimleri yerleşik (*"built-in"*) olarak desteklemektedir. Ayrıca, etmenlerin anlamsal içsel bileşenleri (etmenin bilgi-tabanı gibi) de tezde tanıtılan üstmodel kullanılarak modellenebilmektedir.

MAS'lar için oluşturulmuş bir başka DSML (Gascuena et al., 2012) çalışmasında belirtilmiştir. Bu çalışmada, soyut sözdizim Meta-object Facility (MOF) (OMG, 2002) kullanımıyla, grafiksel somut sözdizim GMF ile gerçekleştirilmiştir. Daha sonra, JET (The Eclipse Foundation, 2009) model dönüşümleri kullanılarak JACK etmen çerçevesine (AOS, 2001) kod üretimi sağlanmıştır. Ancak, geliştirilen modelleme dili sadece Prometheus (Padgham and Winikoff, 2004) MAS metodolojisine özgü olduğundan genel kullanıma uygun değildir. Ayrıca bu DSML'de de Anlamsal Web servisleri kullanılmamaktadır.

4. ANLAMSAL WEB ORTAMINDA ÇALIŞAN MAS’LAR İÇİN PLATFORM BAĞIMSIZ BİR ÜSTMODEL

Anlamsal Web ortamında çalışan MAS’lar için üstmodel geliştirimine, (Kardaş et al., 2009)’da tanıtılan MAS üstmodelin revize edilmesiyle başlanmıştır. Üstmodel, özellikle etmenler ile Anlamsal Web servisleri arasındaki etkileşimin uygun bir şekilde modellenmesini sağlayacak yeni meta-varlıklar ve ilişkiler ile genişletilmiştir. 109E125 numaralı TÜBİTAK projesi kapsamında SEA_ML (“*Semantic web Enabled Agent Modeling Language*”) adı verilen bir DSML’ye ait soyut sözdizim bu tezin de yazarının içinde bulunduğu ortak bir çalışma sonucunda (Challenger et al., 2011)’de gerçekleştirilmiştir. Bu tez kapsamında ise, oluşturulan bu soyut sözdizimden somut sözdizim elde edilerek metinsel ve görsel somut sözdizimler ortaya konmuştur.

Anlamsal Web ortamında çalışan MAS’lar için oluşturulan DSML’ye SEA_ML adı verilmiştir (Benzer şekilde, oluşturulan DSL’ye de SEA_L adı verilmiştir). SEA_ML, Anlamsal Web yetenekli MAS’ların geliştirilmesini sağlamaktadır. Bu sayede, yazılım etmenlerinin web içeriklerini farklı kaynaklardan elde edebildiği, bilgiyi işleyebildiği ve sonuçları gerektiğinde diğer etmenlere göndererek bilgi alışverişinde bulunabildiği bir ortam oluşturulmuştur.

SEA_ML’nin üstmodeli genel olarak değerlendirildiğinde, temel elemanlardan birinin Anlamsal Web etmeni (“*SemanticWebAgent*” – SWA) olduğu görülmektedir. Bilgiyi değerlendirebilen ve içerik dilleri aracılığıyla anlamsal olarak tanımlanmış varlıklar ile beraber çalışabilen yazılım etmenlerine SWA denilmektedir. SWA bir Anlamsal Web organizasyonunun (“*SemanticWebOrganization*” – SWO) içinde çalışır ve farklı servislerle etkileşime geçebilme yeteneğine sahiptir. Bir diğer önemli üstvarlık ise, Anlamsal Web servis (“*SemanticWebService*” – SWS)’dir. SWS, anlamsal olarak tanımlanmış bir web servisini temsil eder. Ayrıca, bir SWS bir ya da daha fazla web servis üstvarlıklarından oluşmaktadır. Servislerin etmenler tarafından kullanılabilmesi için bir anlamsal arayüze ihtiyaçları vardır.

Etmenler, servis yeteneklerini keşfedebilmek için bir servis kayıtçısına başvururlar. Bu nedenle, SEA_ML üstmodelinde SWA'nın bir uzantısı olan Anlamsal Servis Arabulucu Etmeni ("*Semantic Service Matchmaker Agent*" – SSMatchmakerAgent) bulunmaktadır. Bu üstvarlık, Anlamsal Web servislerinin yetenek ilanlarını MAS içerisinde depolayan ve bu yetenekleri diğer etmenler tarafından ihtiyaç duyulan servis yetenekleri ile karşılaştıran aracı ("*matchmaker*") etmenleri temsil etmektedir. SWA, yerine getirmesi gereken hedefler doğrultusunda içsel bilgileri yönetmek ve çıkarsamalar yapmak amacıyla ontolojileri kullanır. Bir SWA, bir veya daha fazla rol ile ilişkilendirilebilir ve zaman içerisinde bu roller değişkenlik gösterebilir.

Ontolojik kavramların tanımlanması amacıyla SEA_ML üstmodeline OMG'nin Ontoloji Tanımlama Üstmodeli ("*Ontology Definition Metamodel*" – ODM) entegre edilmiştir. Ayrıca, SEA_ML üst modeli (Kardaş et al., 2009) çalışmasında desteklenmeyen BDI mimarisini de destekleyecek nitelikte oluşturulmuştur.

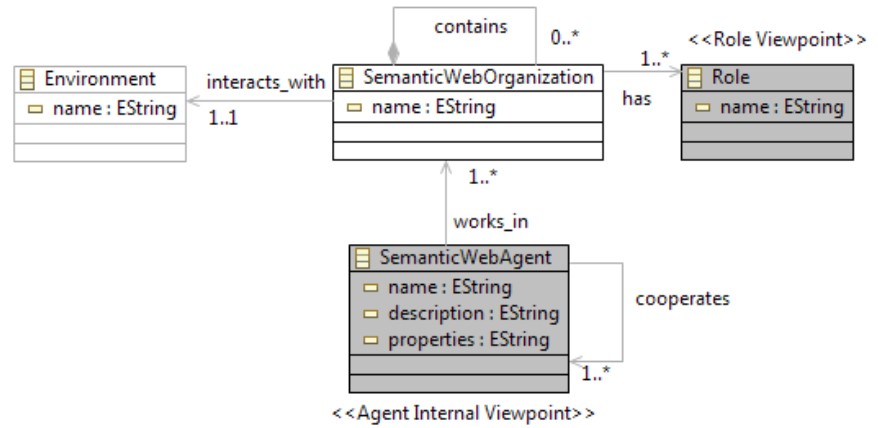
SEA_ML için geliştirilen üstmodelin daha etkin kullanılabilir olması ve daha anlaşılır hale gelmesi için üstmodel farklı bakışaçılarına göre bölümlere ayrılmıştır. Bu bölümler, etmen iç yapısı, etmenler arası etkileşim ve Anlamsal Web yapıları ile etmenlerin etkileşimleri temel alınarak oluşturulmuştur.

Aşağıdaki alt bölümlerde her bir bakışaçısı detaylı bir şekilde anlatılmıştır. Bakışaçılarını gösteren şekillerde, farklı bakışaçılarından gelmiş olan üstvarlıklar gri renkle belirtilmiştir. Bu üstvarlıkların hangi bakışaçısına ait olduğu "<<" ve ">>" karakterleri arasında belirtilmiştir. Örneğin, Şekil 4.1'de rol üstvarlığının <<Role Viewpoint>> etiketi ile Rol bakışaçısına ait olduğu gösterilmektedir. Bunun yanında, bakışaçılarında mutlaka bulunması gereken üstvarlıklar koyu çerçeveye belirtilmiştir. Şekil 4.1'den anlaşılacağı üzere, en az bir tane bulunması gereken eleman SWO'dur ve bu nedenle koyu renk çerçeve ile belirtilmiştir.

4.1 MAS Bakışaçasısı

MAS bakışaçasısı, bir MAS yapısını ve üstmodelin genel görünümünü belirtmektedir. SWO, SWA’ların organizasyonel rollerine dayalı olarak oluşturdukları bir yapıdır. Şekil 4.1’de de görüldüğü gibi, herhangi bir SWA belli bir amaca ulaşmak için organizasyon içerisinde bir ya da birden fazla etmen ile işbirliği (*“cooperates”*) yapabilir. Ayrıca bir etmen aynı zamanda birden fazla SWO’da da bulunabilir. Bir SWO herhangi bir zamanda birden fazla etmeni içerebilir. Bunların yanı sıra bir organizasyon bir ya da birden fazla alt organizasyonun birleşiminden meydana gelebilir.

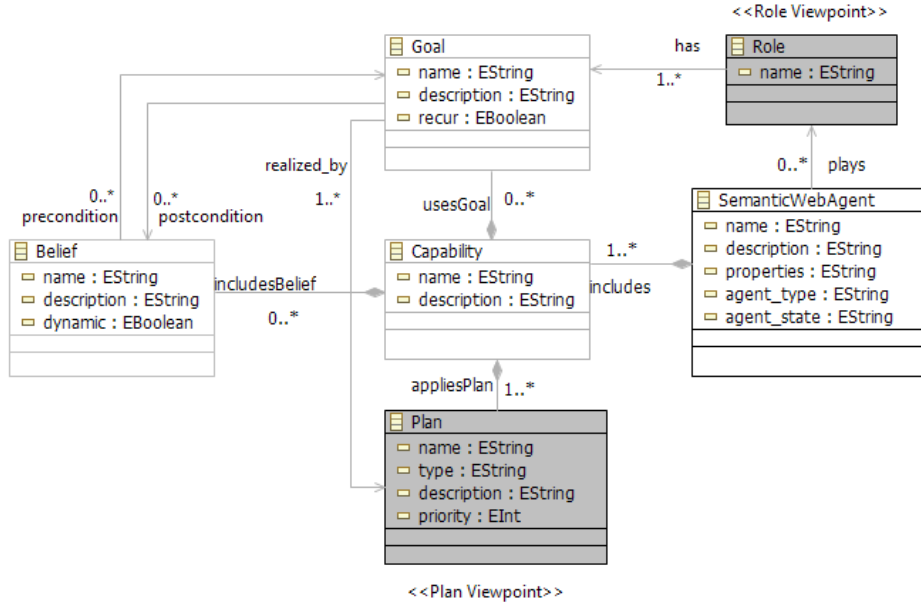
Her bir SWO, kaynakları, servisleri ve veritabanı gibi kavramları içeren bir “Environment” ile etkileşime girer. SWA’lar içinde çalıştıkları SWO’ların kaynaklarını kullanır. Bir SWO, organizasyonun amacını temsil eden bir ya da daha fazla “Role”e sahip olabilir.



Şekil 4.1. MAS bakışaçasısı.

4.2 Anlamsal Web Etmeni İçsel Bakışaçasısı

Anlamsal Web etmeni içsel bakışaçasısı, bir MAS’ta yer alan her bir etmenin içsel yapısına odaklanmaktadır. SEA_ML sözdiziminin bu bakışaçasısını temsil eden kısmi üstmodeli Şekil 4.2’de verilmiştir.



Şekil 4.2. Anlamsal Web Etmeni içsel bakış açısı.

SWA rolleri kapsamında içsel bilgilerini yönetmek ve bildikleri gerçeklere dayalı olarak ortam hakkında çıkarsama yapmak amacıyla ontolojileri kullanır. Bir SWA birden fazla “Role” ile ilişkilendirilebilir ve zamanla rolünü değiştirebilir.

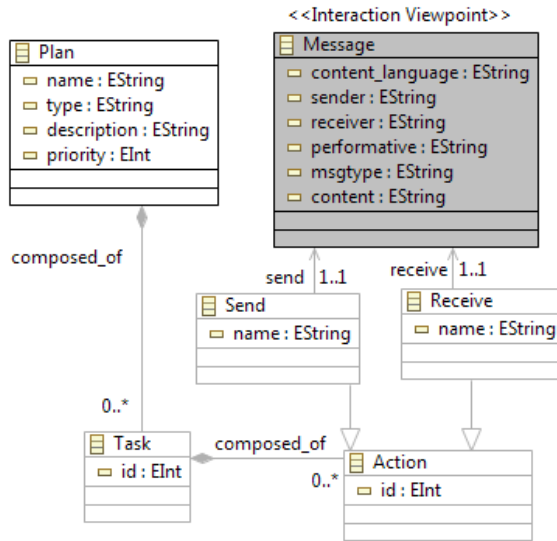
Etmenlerin farklı tiplerde rolleri olabileceğinden, bir etmen herhangi bir zamanda *Yönetici Rolü*, *Komisyoncu Rolü* ya da *Müşteri Rolü* oynayabilir. Belirli bir zamanda bir etmen tek bir durumda (“agent state”) bulunabilir. Etmenin pasif durumda bir etmeni ya da bir kaynağı beklemesi, etmen durumuna bir örnektir. Benzer şekilde, etmen kendi içerisinde bir işlemi yerine getirirken ya da dışarıdaki varlıklarla etkileşim halindeyken aktif durumdadır. Bu nedenle, bir etmenin başka bir etmen ile iletişime geçebilmesi için, iletişime geçilecek olan etmenin durumu göz önünde bulundurulur.

Bir etmen, MAS ortamında bulunduğu sürece tek bir etmen tipine (“agent type”) sahip olabilir. Etmen Tipi, (Haag et al., 2003)’te belirtildiği gibi, etmenin uygulamada yer alacağı bölüme göre belirlenir. “Agent_state” ve “agent_type” özellikleri, SWA içerisinde birer öznitelik (“attribute”) olarak bulunmaktadır.

SEA_ML üstmodelinde, BDI mimarisine sahip etmenlerin hem tasarım hem de çalışma zamanı bileşenlerinin modellenmesi için gerekli elemanlar mevcuttur. Anlamsal Web etmeni içsel bakış açısında bu mimariye göre çalışacak etmenlerin içyapısının modellenmesi hedeflenmiştir. JADEX ve JACK gibi somut BDI çerçeveleri de göz önüne alınarak her etmenin amaçlarını, planlarını ve ortam ile ilgili kanılarını barındıran yetenek üstvarlığı bulunmaktadır.

4.3 Plan Bakış açısı

Adından da anlaşıldığı gibi, plan bakış açısı etmen planlarının içsel yapısını tanımlar. Bir etmen bir “Plan”ı uyguladığında “Plan”, en atomik üstvarlık olan eylemlerden (“*Action*”) oluşan görevleri (“*Task*”) çalıştırmaya başlar. Eylemler atomic olduğundan, gönder (“*Send*”) ve al (“*Receive*”) gibi üstvarlıklar eylemlerin bir uzantısı olup çalıştırılabilir bir yapıya sahiptirler. Şekil 4.3’te de gösterildiği gibi, bu “*Action*” tipleri bir mesaj (“*Message*”) üstvarlığı ile ilişkilendirilmiştir. Bir etmene “*Message*” göndermek ya da bir ontoloji sorgulamak “*Action*” üstvarlığına örnek oluşturur.



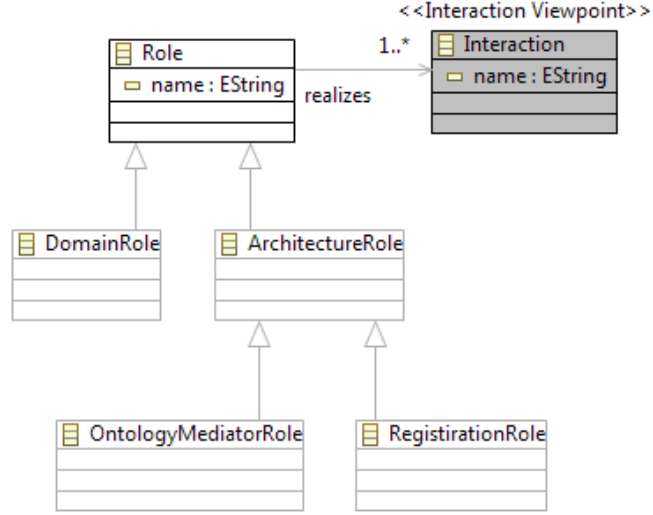
Şekil 4.3. Plan bakış açısı.

4.4 Rol Bakış açısı

SWA'lar ve SWO'lar içinde bulundukları sistem içerisinde bazı "Role"leri oynamakta ve hem içsel bilgilerini yönetmek için hem de ortam hakkında bildikleri gerçeklere göre çıkarsama yapabilmek için ontolojiler gibi anlamsal kaynakları kullanmaktadırlar. Etmenler aynı anda birden fazla "Role" ile ilişkilendirilebilir ve aynı "Role" bir sistem içerisinde birden fazla etmen tarafından oynanabilir. Etmenler zaman içerisinde rollerini değiştirebilir.

Ancak oldukça genel bir anlama sahip bu üstmodel varlığının mimari ve iş alanı tabanlı rollerin görev tanımlarına bağlı olarak model içerisinde özelleşmesinin gerektiğine inanılmış ve üstmodelde bu varlığın iki alt varlığı da mimari rol ("*ArchitectureRole*") ve alan rolü ("*DomainRole*") olarak yer almıştır. Bir "ArchitectureRole", 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 "DomainRole" etmen organizasyonu içeriğine özgü olup tamamıyla özelleştirilmiş bir iş alanı için oluşturulmuş bir Anlamsal Web organizasyonunun ihtiyaçlarına ve görev tanımlamalarına dayalıdır. "DomainRole"ü oynayan bir etmen bir ya da birden fazla rol ontolojisini ("*RoleOntology*") kullanarak sistem ve iş alanı kavramlarından haberdar olur. Organizasyonda yer alan bir kısım etmenin organizasyona özgü bazı servisleri sağlamak amacıyla ortaya konan rolleri oynaması gerekir. Bu nedenle üstmodelde "ArchitectureRole"ün iki özelleşmiş alt model elemanı olan kayıt rolü ("*RegistrationRole*") ve ontoloji arabulucu rolü ("*OntologyMediatorRole*") yer almaktadır. "RegistrationRole" organizasyondaki etmenler için sarı sayfa servisi sunan bir ya da daha fazla SWA tarafından oynanmaktadır. Üstlendiği rolün kapsamına göre söz konusu bu etmenler SWA'ların ya da SWS'lerin yetenek ilanlarını saklayabilir.

SEA_ML sözdiziminin bu bakış açısını temsil eden kısmi üstmodeli Şekil 4.4'te verilmiştir.

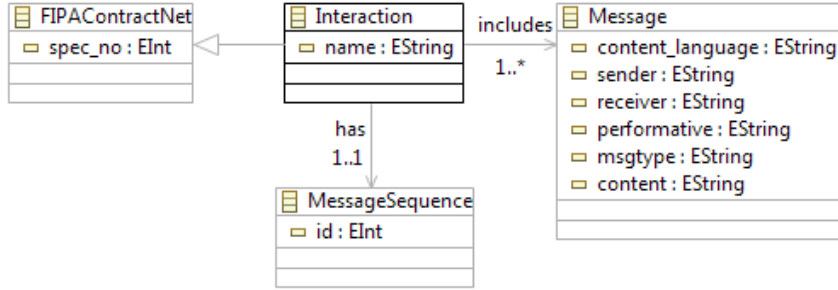


Şekil 4.4. Rol bakış açısı.

4.5 Etkileşim Bakış açısı

Bu bakış açısı etmen iletişimlerine ve MAS içerisindeki etkileşimlere odaklanmakta, “Interaction”, “Message” ve mesaj dizisi (“*MessageSequence*”) gibi üstvarlıkları ve ilişkilerini tanımlamaktadır. Etmenler sosyal yeteneklerine göre birbirleriyle etkileşimde bulunurlar. Her bir “Interaction” kendi içerisinde alt “Message”lardan oluşmaktadır. Bu mesajların bilgilendirme (“*inform*”), talep (“*request*”), onay (“*acknowledgement*”) gibi mesaj tipleri bulunmaktadır. Özellikle, iletişime başlayan ve iletişimde bulunan etmenler, IEEE FIPA standardıyla (FIPA, 2002b) uyumlu, gerçekleştirici (“*performative*”) özelliğe sahip “Message” üstvarlığı ile modellenebilir. “Message”ın içerik dili etmenler arasındaki iletişim için kullanılan Bilgi Sorgulama ve İşleme Dili (“*Knowledge Query and Manipulation Language*” – KQML) (Finin et al., 1997) ya da FIPA ACL (FIPA, 2002c) gibi iletişim dillerinden biri olabilir. Etkileşim üstvarlığı, FIPAContractNet üstvarlığından türetilmiştir. FIPAContractNet, Kontrat Net Protokol’ünü (“*Contract Net Protocol*” – CNP) (Smith, 1980) uygulayan etmen etkileşimleri için belirlenmiş IEEE FIPA spesifikasyonunu temsil etmektedir. Ayrıca, her bir “Interaction”ın, iletişim akışını takip edebilmek için en az bir “Message”a sahip olması gerekmektedir. Dağıtık etmenlerin iletişimi, bir akış diyagramı ya da bir aktivite diyagramı olarak ele alınabilir.

SEA_ML sözdiziminin bu bakış açısını temsil eden kısmi üstmodeli Şekil 4.5’te verilmiştir.

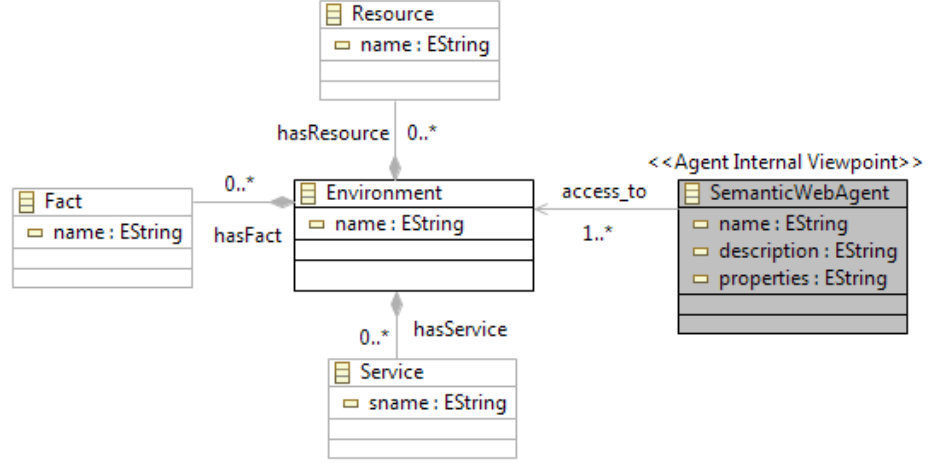


Şekil 4.5. Etkileşim bakış açısı.

4.6 Çevre Bakış açısı

Etmenlerin kaynak kullanımı ve çevre etkileşimleri bu bakış açısında ele alınmaktadır. Etmenlerin bulundukları çevre, kaynakları (“*resource*”), gerçekleri (“*fact*”) ve “Service”leri içermektedir. Her bir “Service”, bir “WebService” ya da gerçek uygulamalarda öntanımlanmış protokoller olabilir. “Fact”ler, “Environment”a göre değişiklik gösterebilir. “Environment” yeni bir bilgi edildiği zaman, ya da koşullar değiştiği zaman “Fact”ler de güncellenebilir. Bir “Environment”deki “Fact”lerle, etmenin “Belief”leri karıştırılmamalıdır. Örneğin, bir “environment” içerisinde, hava durumuyla ilgili bilinen “Fact”in “Hava sıcaklığı 30 °C’dir” şeklinde olduğu varsayalım. Bir etmen bu bilgiyi kendi kanıtabanına kaydedebilir. Daha sonra, hava sıcaklığıyla ilgili bilinen “Fact” çevre koşullarına bağlı olarak “Hava sıcaklığı 15°C’dir” şeklinde değişebilir. Bu durumda, etmen kendi kanıtabanını güncellemediği takdirde, etmenin “Belief”, “Fact”lerden farklılık gösterebilecektir.

SEA_ML sözdiziminin bu bakış açısını temsil eden kısmi üstmodeli Şekil 4.6’da verilmiştir.



Şekil 4.6. Çevre bakış açısı.

4.7 Etmen-Anlamsal Web Servis Etkileşim Bakış açısı

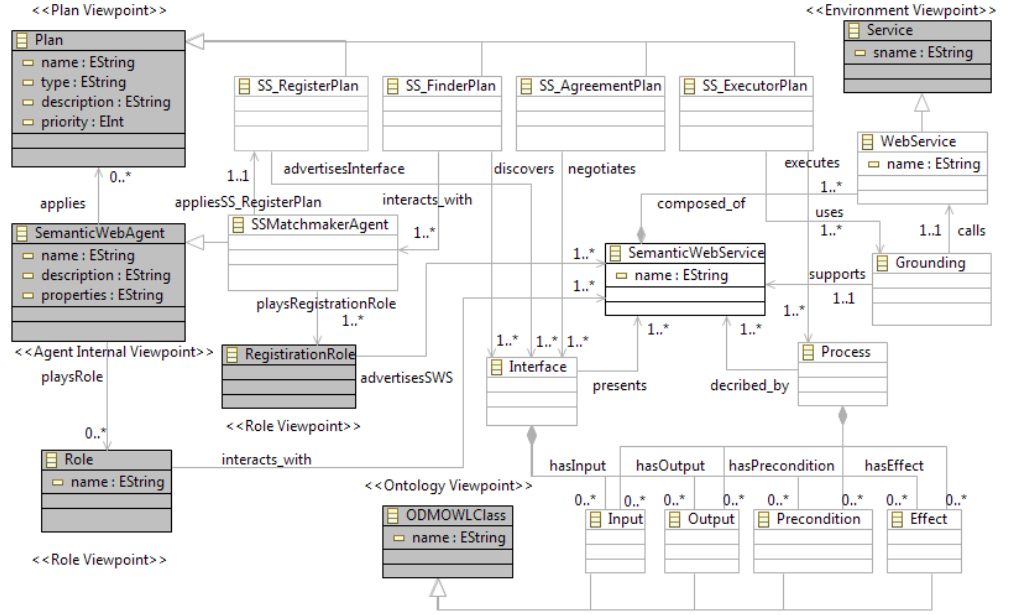
Etmen-Anlamsal Web Servis Etkileşim bakış açısı, SEA_ML üst modelinin en önemli bakış açısı olarak değerlendirilebilir. SWA'ların SWS'ler ile nasıl etkileşime gireceği bu bakış açısı ile modellenir (Challenger et al., 2011). Uygun servisin bulunması, servis ile anlaşılması ve anlaşılan servisin çalıştırılması ile ilgili varlıkların tanımı bu üstmodelde yer alır. Aynı zamanda SWS'lerin içsel yapısı da bu bakış açısı ile modellenir.

Şekil 4.7'de etmen-Anlamsal Web servis etkileşim bakış açısı gösterilmektedir. SWA, "Role", "RegistrationRole" ve "Plan" elemanları üstmodelin diğer bakış açılarında dahil edilen üstvarlıklar olup tanımları önceki altbölümlerde verildiğinden burada sadece yeri geldiğinde bu bakış açısının doğrudan elemanı olan üstvarlıklar ile olan ilişkileri anlatılmaktadır. "ODMOWLClass" üstvarlığı ise ontoloji bakış açısında anlatılmaktadır.

OWL-S'te tanımlanan servis arayüzü, süreç modeli ve fiziksel altyapı elemanları Şekil 4.7'deki arayüz ("interface"), süreç ("process") ve altyapı ("grounding") isimli üstvarlıklara karşılık gelmektedir. Bu bileşenlerin girdi ("input"), çıktı ("output"), önkoşul ("precondition") ve etki ("effect") tanımları da yine üstmodelde bulunmaktadır. Bu model elemanları, (Kardaş et al., 2010)'dan farklı olarak, OMG'nin ODM'sinde bulunan OWL sınıfını

(“*ODMOWLClass*”) genişletmektedir (“*extension*”). SWA’lar görevlerini yerine getirmek için “Plan”ları uygulamaktadır. SWS’lerini dinamik olarak keşfetmek, onlarla anlaşmak ve onları çalıştırmak için “Plan” üstvarlığının üç uzantısı üstmodelde tanımlanmıştır. Anlamsal servis bulucu plan (“*semantic service finder plan*” – SS_FinderPlan) aday SWS’lerin keşfedilmesini kapsayan bir plandır. Anlamsal servis anlaşma planı (“*semantic service agreement plan*” – SS_AgreementPlan) servis kalitesi metrikleri (örneğin servis çalışma maliyeti, çalışma zamanı, servisin yeri, vb.) üzerine anlaşmayı ve anlaşmanın hazırlanmasını içermektedir. Servisin keşfi ve servisle anlaşma tamamlandıktan sonra etmen uygun SWS’yi çalıştırmak amacıyla anlamsal servis çalıştırıcı planını (“*semantic service executor plan*” – SS_ExecutorPlan) uygulamaya geçirir. Söz konusu plan uzantıları Şekil 4.7’de gösterilmiştir.

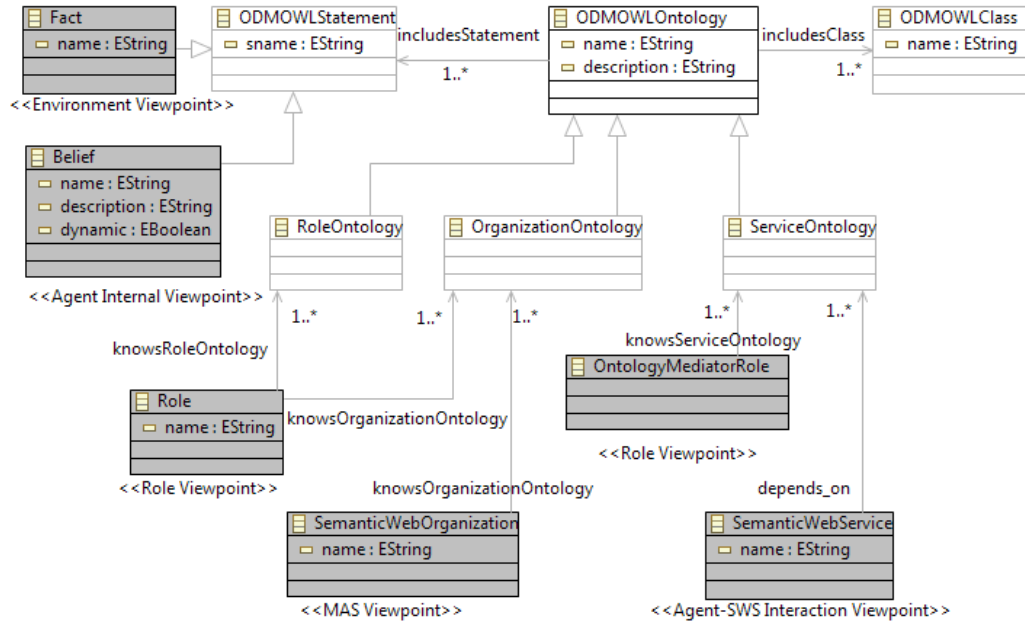
Etmenler servis yeteneklerini keşfetmek amacıyla bir servis kayıtçısı ile iletişime geçme ihtiyacı duyarlar. SWS’lerin yetenek ilanlarını bir MAS içerisinde depolayan ve diğer etmenler tarafından yollanan servis arama mesajları sonucunda ihtiyaç duyulan servis yetenekleri ile ilan edilenleri karşılaştıran aracı etmenler (“*SSMatchmakerAgent*”) modelde temsil edilmektedir. Bu etmenlerin söz konusu iş için uygulamaya koydukları planlar yine plan varlığının bir uzantısı olan anlamsal servis kayıt planı (“*semantic service register plan*” – SS_RegisterPlan) adını almaktadır.



Şekil 4.7. Etmen-Anlamsal Web Servis Etkileşim bakışaçasısı.

4.8 Ontoloji Bakışaçasısı

Anlamsal Web yetenekli MAS organizasyonları ontolojiler olmadan düşünülemez. Bir ontoloji MAS elemanları için bilgi toplama ve akıl yürütme kaynaklarını temsil eder. Ontoloji bakışaçasısı, bütün ontoloji kümelerini ve ontolojik kavramları bir araya getirir. “ODMOWL Ontology”, üstmodelde tanımlanan “Role Ontology”, servis (“Service Ontology”) ve organizasyon ontolojileri (“Organization Ontology”) gibi tüm ontoloji kümeleri için bir standart oluşturur. Şekil 4.8’de resmi verilen bu bakışaçasısına göre, tüm ontolojiler ilgili elemanlar tarafından bilinmektedir. Bazı ontolojilerin bir araya gelmesiyle MAS’ın alan içeriğini oluşturan bilgi-tabanı oluşmaktadır. Bu ontolojiler, SEA_ML modellerinde organizasyon ontolojisinin birer örneğidir. Bir “DomainRole” doğrultusunda bir etmen, ilişkili etmen rolü kavramlarının ve ilişkilerinin tanımlandığı “Role Ontology”yi kullanır.



Şekil 4.8. Ontoloji bakış açısı.

5. GRAFİKSEL SOMUT SÖZDİZİMİN GERÇEKLEŞTİRİMİ

SEA_ML'e ait kavramlar ve ilişkiler soyut sözdizimi ile belirlendikten sonra Atlas Dönüşüm Dili ("*Atlas Transformation Language*"– ATL) (Eclipse ATL, 2007) ortamında hazırlanan SEA_ML üstmodel varlıkları ile bunların grafiksel gösterimleri arasındaki eşleşmeler sağlanmış ve görsel somut sözdizimi GMF üzerinde oluşturulmuştur. Somut sözdizim sayesinde örnek modelin SEA_ML kullanılarak oluşturulan MAS üstmodeline göre geçerliliği sağlanmaktadır. Somut sözdizim tanımı görsel ve metinsel olmak üzere iki şekilde yapılmıştır. Bu bölümde üstmodelde belirtilen her bir bakış açısı için oluşturulmuş görsel bir somut sözdizimine yer verilmiştir. Görsel somut sözdizimin nasıl oluşturulduğunu anlatmak amacıyla, Anlamsal Web etmeni içsel bakış açısının GMF ortamında gerçekleştirimi detaylı olarak anlatılmıştır. Diğer bakış açılarının geliştirimi de benzer şekilde gerçekleştirilmiştir.

Görsel sözdizimin oluşturulması için ilk önce SEA_ML kavramlarının grafiksel notasyonları belirlenmiştir. Tablo 5.1'de tüm elemanlara karşılık gelen grafiksel gösterimler bulunmaktadır.

Tablo 5.1. SEA_ML somut sözdizim elemanları.

Kavram	Gösterim	Kavram	Gösterim
Semantic Web Agent		Service	
Semantic Service Matchmaker Agent		Web Service	
Belief		Semantic Web Service	
Goal		Grounding	
Role		Process	
Capability		Interface	
Fact		Input	
Plan		Output	
Semantic Service Register Plan		Precondition	
Semantic Service Finder Plan		Effect	
Semantic Service Agreement Plan		ArchitectureRole	
Semantic Service Executor Plan		Ontology Mediator Role	
Send		Semantic Web Organization	
Receive		Role Ontology	
Task		Organization Ontology	
Action		Service Ontology	
Message		Interaction	
Message Sequence		Environment	
DomainRole		RegistrationRole	

Tablodaki tek sütunlar SEA_ML kavramlarına, çift sütunlar ise kavramların somut gösterimlerine karşılık gelmektedir. Oluşturulan aracın (Getir et al., 2011a) tüm kullanıcılara hitap edebilmesi amacıyla elemanlar ve ilişkiler İngilizce isimleri ile modellenmiştir. Kavramlara ait grafiksel gösterimler belirlenirken, model elemanlarından yakın türde olanların veya birbirinden türeyen elemanların benzer şekilde oluşturulmasına dikkat edilmiştir. Örneğin plan elemanından kalıtım yoluyla türeyen SS_RegisterPlan, SS_AgreementPlan ve SS_ExecutorPlan elemanları için plan zeminine ait gösterim kullanılmıştır ve ne tür bir plan olduklarını ifade eden kelimelerin baş harfleri bu zemine eklenmiştir.

Kavramlar için grafiksel gösterimler belirlendikten sonra bu grafiksel gösterimler GMF kullanılarak bir araya getirilmiştir.

5.1 Anlamsal Web Etmeni İçsel Bakış açısı için Görsel Somut Sözdizimin Gerçekleştirimi

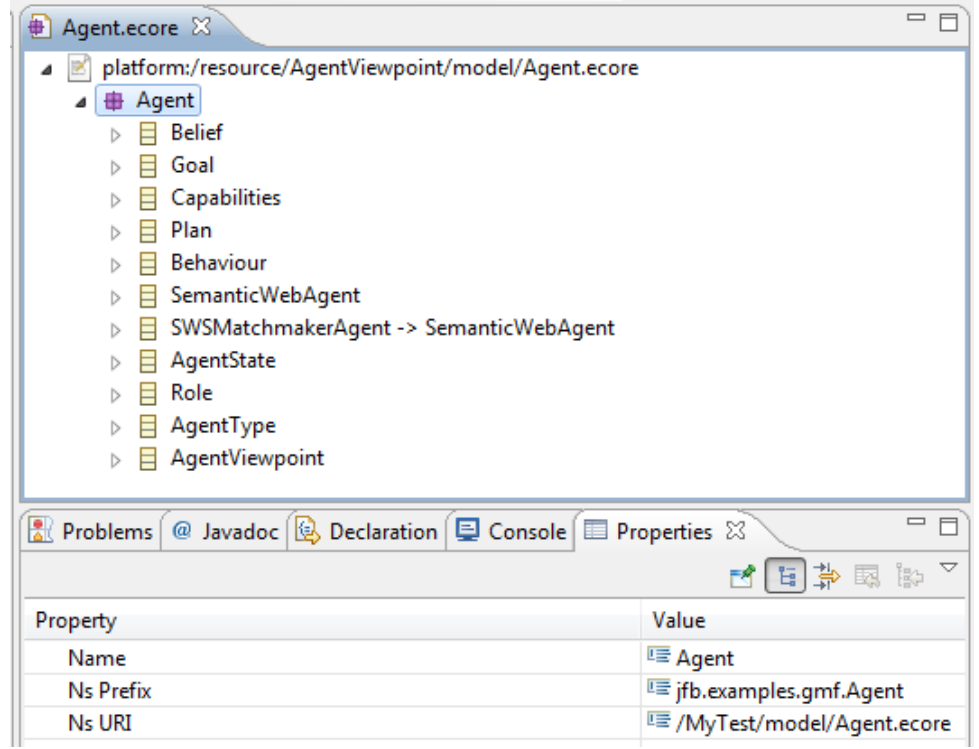
Görsel somut sözdizimin gerçekleştirimi için izlenen yolu detaylı olarak anlatmadan önce yapılan işlemler genel olarak şu şekilde özetlenebilir:

Öncelikle soyut sözdizimin yer aldığı Ecore dosyasında modele ait tüm elemanların ve ilişkilerin özelliklerinin atanması işlemi tamamlanmıştır. Aksi takdirde boş değer (“*null*”) atamaları tüm model oluşumlarında hata vermektedir. Ecore dosyası GMF tabanlı araç üretimi için ihtiyaç duyulan diyagram dosyasına, Eclipse GMF ortamının sunduğu imkanlar ile dahil edilmiştir. Böylece elemanların ve ilişkilerin tüm özelliklerinin görülmesi sağlanmıştır. Devamında ise, modele ait Ecore dosyasını sistemin kendi iç yapısına dönüştüren ve kod üretimini gerçekleştiren “EMF generating model” elde edilmiştir. Bu modeli kullanarak yazılımcıların kullanacağı araçları oluşturmak için “GMF Simple Tooling Definition” modeli üretilmiştir. Bu modelde, etmen yazılımı geliştiricilerinin kullanacağı palette bulunması istenen elemanlar belirtilmektedir.

Üretici model, modelde tanımlanacak elemanların paletteki biçim ve renk ayarlarının yapılabilmesini ve etiketlerin tanımlanacağı grafik modelinin üretilmesini sağlamaktadır. Bu sayede üretici model aracılığıyla “GMF Simple Graphical Definition Model” üretilerek palet oluşturulmuştur. Son aşamada, modeldeki temel elemanların ve elemanlar ile ilişkilerinin kısıtlarının oluşturulması için “Guide Mapping Model Creation” işlemi gerçekleştirilmiştir. Bunun hemen ardından model ve diyagram kodu üretimi gerçekleştirilmiştir.

Bu işlemlerin adım adım uygulanması için öncelikle Eclipse ortamına gerekli eklentiler (EMF, GMF) eklenmiştir. İhtiyaç duyulan ortam hazırlandıktan sonra, yeni bir GMF projesi oluşturularak projenin ismine “AgentViewpoint” adı verilmiştir. Oluşan projenin içerisinde “model” adlı dosya üzerinden sağa tıklanarak yeni bir “Ecore Model” oluşturulmuştur. Oluşturulan bu Ecore modele

“Agent” adı verilmiştir. Bu işlem tamamlandıktan sonra, “model” dosyasının içinde oluşan “Agent.ecore” isimli dosya üzerinde Anlamsal Web Etmeni İçsel bakışaısına ait üstvarlıklar ve ilişkiler tanımlanmıştır. “Agent.ecore” dosyası üzerine çift tıklandığında açılan pencere üzerinde üstmodel için belirtilen “Name”, “Ns Prefix” ve “Ns URI” özelliklerinin tanımlanması gerekmektedir. Aksi takdirde bir sonraki aşamada hata oluşacaktır. “Agent.ecore” için gerekli tanımlamalar yapıldıktan sonra, Ecore üzerinden “SemanticWebAgent” adı verilen üstvarlık oluşturularak “name”, “description” ve “properties” öznitelikleri tanımlanmıştır. Bu bakışaısında yer alan diğer üstvarlıklar da (amaç, rol, yetenek, plan ve kanı) benzer şekilde oluşturulmuştur. Şekil 4.2’de de görüldüğü gibi bir SWA bir ya da daha fazla rol oynayabilmektedir. Bu ilişkiyi tanımlamak için, SWA üstvarlığı üzerinden EReference seçimleri ile “plays” ilişkisi oluşturulmuştur. Bu ilişkinin “EType” özelliği rol olarak tanımlanmıştır. Ayrıca, bir SWA için birden fazla rol tanımlanabilmesi amacıyla “Upper Bound” özelliği -1 olarak atanmıştır. Şekil 4.2’deki tüm üstvarlıklar ve ilişkiler benzer şekilde tanımlandıktan sonra “Agent.ecore” üstmodeli tamamlanmıştır. Burada dikkat edilmesi gereken önemli bir nokta tüm üstvarlıkları içerecek bir “root eleman” oluşturmaktır. “Agent.ecore”da, “AgentViewpoint” adındaki bu üstvarlık tüm üstvarlıklarla ilişkilendirilmiştir. Şekil 5.1’de GMF üzerinde oluşturulan “Agent.ecore” gösterilmektedir.

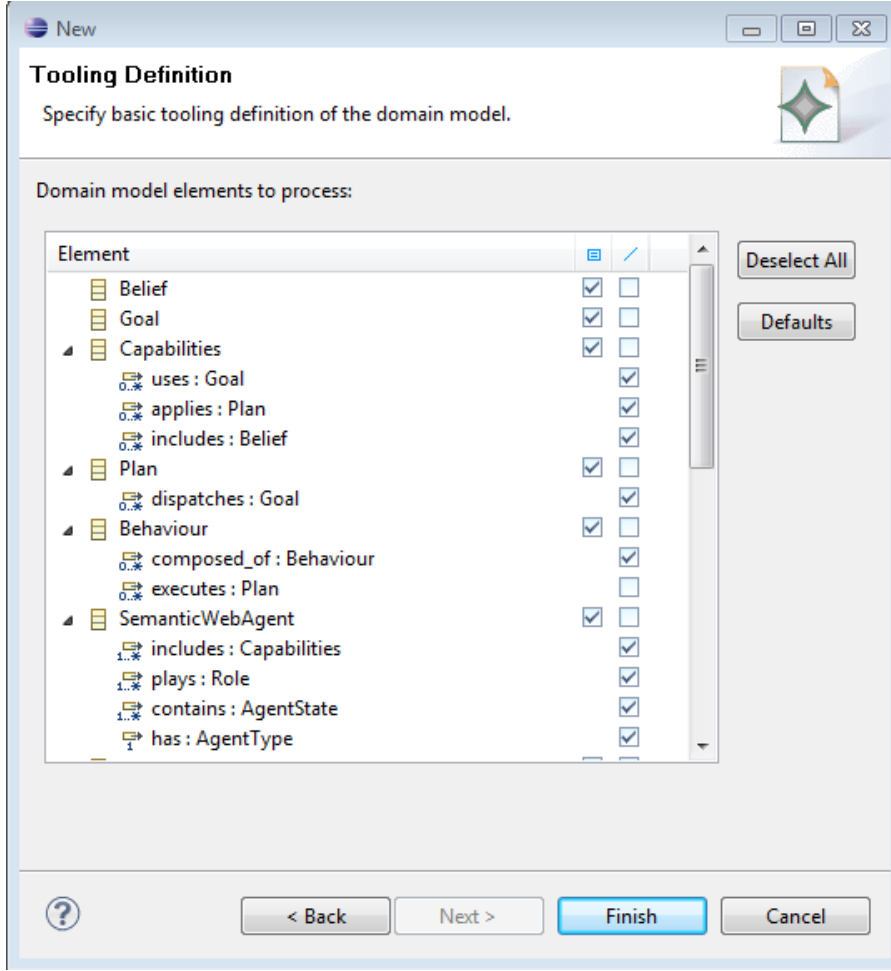


Şekil 5.1. “Agent.Ecore” üstmodeli.

“Agent.Ecore” üstmodeli oluşturulduktan sonra dosya üzerinden “Initialize ecore_diagram file” seçimi ile diyagram dosyası oluşturulmuştur.

Üstmodelin oluşturulmasından sonra, “Agent.ecore” üzerinden platformun kendi içerisinde kullanacağı “Agent.genmodel” dosyası otomatik olarak oluşturulmuştur. “.genmodel”in oluşumu hazırlanan Ecore dosyası üzerinden olduğu için Ecore üstmodelinin oluşumu sırasında yapılan herhangi bir hata burada gözükecektir. “Agent.genmodel” üzerinden “Generate Model Code” ve “Generate Edit Code” seçimleriyle Ecore dosyası sistemin kendi içyapısına dönüştürülmüş ve kod üretimi gerçekleştirilmiştir. Bu işlemler tamamlandıktan sonra grafiksel modelleme kısmına geçilmiştir.

Öncelikle “Agent.Ecore” dosyası üzerinden “Simple Tooling Definition Model” seçeneği seçilmiştir. Daha sonra üstmodel için belirlenen root eleman seçimi yapılmıştır ve ekrana çıkan tablo üzerinden araçta görülmesi istenilen üstvarlıklar ve ilişkiler Şekil 5.2’deki gibi seçilmiştir.



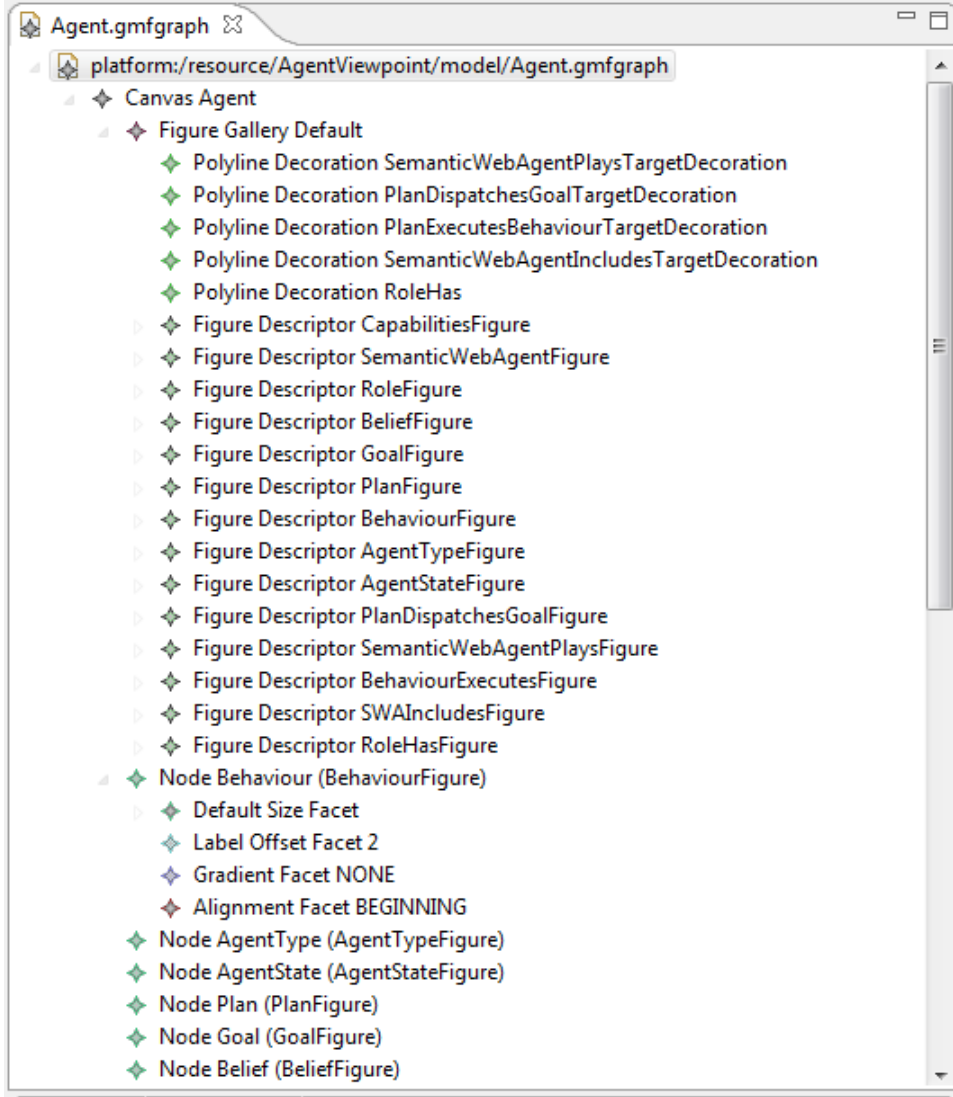
Şekil 5.2. “Simple Tooling Definition Model”de üstvarlıkların seçimi.

Araç için belirlenen görsel notasyonların yer alması için, oluşan “Agent.gmftool” açılarak “Tool Group Agent” dizininin altındaki her bir “Default Image” elemanları silinmiştir. Silinen bu elemanlar yerine “New Child” ve “Small Icon Default Image” seçenekleri ile oluşturulan paletin sağ tarafında gösterilecek olan üstvarlıkların küçük resimleri tanımlanmıştır. Palettteki bu elemanlar sürükleyip bırak yöntemi ile modellendiğinde beyaz ekranda gözükecek büyük resimler için, “Large Icon Default Image” seçeneği eklenmiştir. Resimler “AgentViewpoint.edit” dosyasının içerisindeki “icons” alt klasörüne kaydedilmiştir.

“Simple Tooling Definition Model” üzerindeki işlemler bittikten sonra yukarıda anlatılanlara benzer biçimde “Simple Graphical Definition Model” seçilerek “Agent.gmfgraph” oluşturulmuştur. “Simple Graphical Definition

Model” ile diyagramda gözükecek olan figürler, linkler, seçilen figürlerin modelleme yapılırken nasıl gözükeceği gibi grafiksel işlemler tanımlanır. Bu bölümde yapılan tanımlamalar bir ağaç yapısına benzemektedir. Örneğin bütün üstvarlık tanımlamaları “Canvas Agent” adındaki bir “root” elemanının altındadır. Yetenek üstvarlığı için biçimsel tanımlamaların yapıldığı “FigureDescriptionCapabilitiesFigure” un altında elemanın ekranda dikdörtgen bir geometrik şekle sahip olacağını belirten “Rectangle Capabilities Figure” sekmesi, onun altında ise dikdörtgen çerçevenin kalınlığının ve alanının belirlendiği sekmeler yer almaktadır. Bu bölümde, aracı oluşturan kişinin tercihinine göre elemanlar için açıklama, etiket gibi farklı görsel özellikler tanımlanabilir.

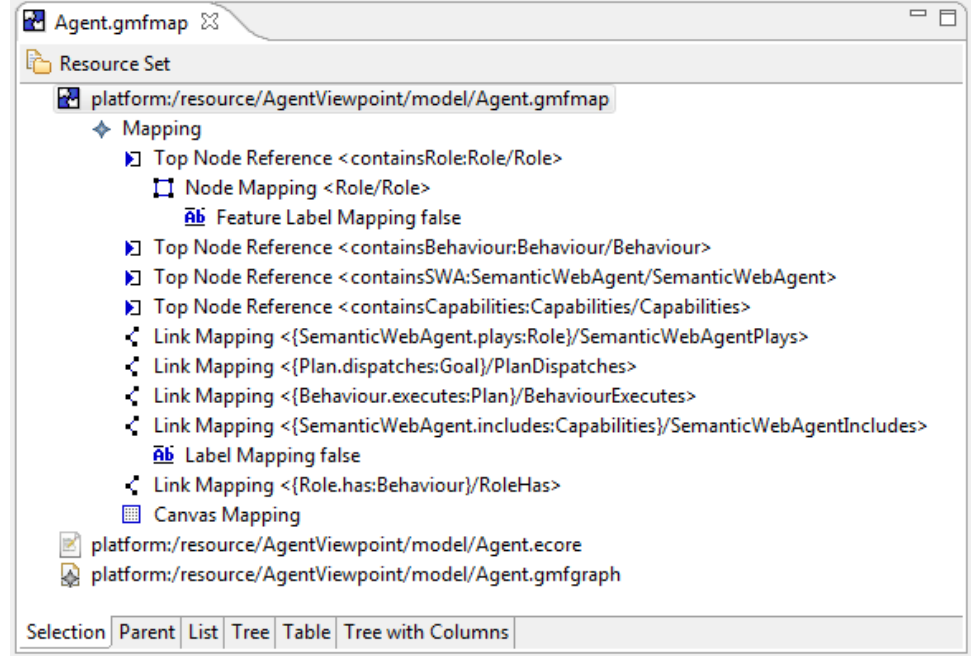
Şekil 4.2’deki Anlamsal Web etmeni içsel bakış açısına göre, yetenek üstvarlığı plan, kanı ve amaç üstvarlıklarını içerebilmektedir. Bu nedenle, araçta tanımlanan bir yetenek elemanı içerisine plan, kanı ve amaç elemanlarının konulabilmesi mümkün hale getirilmiştir. Bu özellik hücreleme (“*compartment*”) tanımlamalarıyla gerçekleştirilmiştir. Bunun için “Canvas Agent” üzerinden hücreleme seçeneği ile plan, kanı ve amaç elemanları için hücreler oluşturulmuştur ve özellikler kısmından adı, hangi eleman tarafından kullanılabileceği, hangi figür ile temsil edilebileceği gibi özellikler tanımlanmıştır. Bu tanımlamalardan sonra, “Figure Gallery Default” sekmesine gelinerek “Figure Descriptor CapabilitiesFigure” üzerinden plan, kanı ve amaç elemanları için “New Child” -> “Child Access” seçenekleri ile yeni sekmeler oluşturulmuştur. Burada dikkat edilmesi gereken nokta, oluşturulan yeni sekmelerin “accessor” ve “figure” özellikleri daha önce oluşturulan hücrelerin özellikleriyle aynı olmalıdır. Şekil 5.3’te Anlamsal Web servis etmeni içsel bakış açısı için oluşturulan “Agent.gmfgraph” gösterilmektedir.



Şekil 5.3. “Agent.gmfgraph”ın gösterimi.

Grafiksel tanımlamalar yapıldıktan sonra, üstvarlıklar arasındaki ilişki tanımlamalarının yapıldığı “Mapping Definition” bölümüne geçilmiştir. Bu bölümde, önceden tanımlanan “Agent.ecore”, “Agent.gmftool” ve “Agent.gmfgraph” modelleri bir araya getirilmiştir. Yeni bir “GMFMap” model oluştururken belirtilen üç model eklenerek birbirine bağlanmıştır. “Mapping” aşamasında üstmodelde belirtilen ilişkiler tanımlanmıştır ve kullanıcıların ilişkili olmayan elemanlar arasına link koymaları kısıtlanmıştır. “Agent.gmfmap” oluşturulduktan sonra, rol, davranış, yetenek, SWA, plan ve amaç üstvarlıkları için “New Child” -> “Top Node Reference” seçimleri ile “Top Node Reference”lar ve bu referansların altına aynı yöntemle “Node Mapping”ler ve “Feature Label Mapping”ler eklenmiştir. Her bir elemanın diğer elemanlarla olan

ilişkilerini tanımlamak için, “Link Mapping”ler eklenerek kaynak ve hedef elemanlarının belirtildiği “Source Feature” ve “Targer Feature” özelliklerinde ilgili elemanlar tanımlanmıştır.



Şekil 5.4. “Agent.gmfmap”ın gösterimi.

Son olarak, “Agent.gmfmap” modeli içerisindeki “Mapping” sekmesi sağa tıklanarak, “Create generator model” seçeneği seçilerek “Agent.gmfgen” adındaki “GMF generator model” oluşturulmuştur.

Oluşturulan somut sözdizimlerin üstmodel ile uyumlu olması için hem görsel somut sözdizim de hem metinsel somut sözdizimde bazı kısıtlar tanımlanmıştır. Görsel somut sözdizim için “Mapping” aşamasında tanımlanan bu kısıtlar (Getir ve ark., 2011b)’de de anlatıldığı gibi şu şekildedir:

Hücreleme kısıtı: Üst modelde oluşturulan elemanlar arasındaki bütünleştirme (“*aggregation*”) ilişkisinin örnek model seviyesinde birbirini içeren iki elemana dönüşmesini gerçekler. Bu ilişki türüne sahip olmayan iki eleman iç içe hücre şeklinde modellenemez. Örneğin “Agent.gmfgraph”ın oluşturulması sırasında da belirtildiği gibi, Anlamsal Web servis etmeni içsel bakış açısına ait üstvarlıklardan yetenek üstvarlığı plan, kanı ve amaç elemanlarından oluşur. Üst

modeldeki ilişkiler, bütünleştirme olarak tanımlandığından plan, kanı ve amaç, yetenek üstvarlığının bir hücresi olacaktır. Aynı bakış açısında SWA ve rol elemanları arasında ise böyle bir ilişki ve modelleme uygulanamaz.

İlişki sayısı kısıtı: İki eleman arasındaki bire bir (“one to one”), bire çok (“one to many”), çoktan çoğa (“many to many”) gibi ilişkiler “Ecore” aşamasında belirlendikten sonra, örnek modelde bu kısıtlar iki eleman arasındaki ilişkileri sınırlar. Örneğin bir etkileşim, birden fazla mesaj içerebilirken yalnızca bir mesaj dizisine sahip olabilir.

İlişki kaynağı ve hedefi kısıtı: Bir ilişkinin hangi elemandan hangi elemana doğru olduğu, o ilişkinin kaynağını ve hedefini belirler. Bu kısıt “Ecore” seviyesinde tanımlanmaya başlanır. Örneğin SWA ile rol arasındaki “plays” ilişkisi roleden SWA’ya doğru tanımlanamaz.

Kalıtım ilişkisi kısıtı: Üst seviyede tanımlanan kalıtım ilişkileri, örnek modeller yaratılırken kalıtımın doğası gereği bazı kısıtlar getirmektedir. Örnek modeldeki bir alt sınıf üst sınıfının tüm özneliliklerini ve ilişkilerini içerir. Buna güzel bir örnek etmen-Anlamsal Web servis etkileşim bakış açısı kapsamında verilebilir. Şekil 4.7’de gösterildiği gibi plan üstvarlığının dört tane alt sınıfı mevcuttur. Plan elemanının diğer elemanlarla direkt olarak ilişkileri var olsa bile, oluşturulan araçta alt sınıflar yer almış, planın tüm ilişkileri kalıtım yoluyla bu alt sınıflardan türeyen örneklerle aktarılmıştır.

İlişki eleman bütünlüğü: Örnek modelde oluşturulan bir elemanın kaldırılmasıyla, ona ait tüm ilişkiler de modelden silinmektedir.

Belirtilen aşamaların tamamlanmasıyla Anlamsal Web servis etmeni içsel bakış açısı için gerekli araç oluşmuştur. Bu aracın kullanımı Grafiksel Sözdizimin Değerlendirilmesi bölümünde anlatılacaktır.

6. METİNSEL SOMUT SÖZDİZİMİN GERÇEKLEŞTİRİMİ

SEA_L için geliştirilen metinsel somut sözdizim Xtext aracı ile her bir bakış açısı için gerçekleştirilmiştir. Bu bölümde etmen-Anlamsal Web servis etkileşim bakış açısı için gerçekleştirilen metinsel somut sözdizime yer verilmiştir. Diğer bakış açılarının gerçekleştirimleri de benzer şekilde yapılmıştır.

Belirtilen yöntemlerle yeni bir Xtext projesi oluşturulduktan sonra, ilk önce EBNF kuralları tanımlanmıştır. EBNF kuralları ile oluşturulmak istenen metinsel editörün sözdizimi belirlenmiştir. Öncelikle kullanıcının oluşturacağı örnek model “Agent-SemanticWebServiceInteractionViewPoint” etiketinin içinde olmalıdır. Örnek modellerin her birinde en az bir tane SWA ve SWS elemanı bulunmak zorundadır. Aksi halde editörde hata işareti oluşur. Etmen-Anlamsal Web servis etkileşim bakış açısına ait üstmodelde belirtildiği gibi, her bir SWS “Interface”, “Process” ve “Grounding” bileşenlerine sahiptir. Öncelikle bir “Interface” servisin yeteneklerini gösterir. Daha sonra “Process” ile servisin iç birleşimi ve çalıştırma dinamikleri tanımlanır. Son olarak ise, fiziksel altyapı kullanılarak servisin çalışma protokolü tanımlanır. Bu bakış açısıyla “Interface”, “Process”, “Grounding” ve SWS elemanları arasındaki ilişkiler tanımlanır. Oluşturulan metinsel sözdizimi bu elemanların etkileşimlerini belirtilen sıra ve belirtilen ilişki dışında tanımlanmasına izin vermez. Fakat “Interface” ve “Process” elemanlarının içerdiği girdi, çıktı, önkoşul ve etki elemanlarının sayısına herhangi bir kısıt getirilmemiştir. Örneğin, bir “Interface” birden fazla girdi ve çıktı elemanına sahip olabilmektedir. Bu bakış açısında dikkat çeken bir diğer kısıt plan üstvarlıklarının sırasıdır. Şekil 4.7’de gösterilen plan üstvarlıkları, SS_RegisterPlan, SS_FinderPlan, SS_AgreementPlan ve SS_ExecutorPlan sırasında kullanılabilmektedir. EBNF kuralları, bu sıralama dikkate alınarak oluşturulmuştur.

Etmen-Anlamsal Web servis etkileşim bakış açısına ait metinsel sözdizimin bir kısmı Şekil 6.1’de belirtilmiştir.

```

1 grammar org.xtext.example.mydsl2.AgentSWSInteraction with org.eclipse.xtext.common.Terminals
2 generate agentSWSInteraction "http://www.xtext.org/example/mydsl2/AgentSWSInteraction"
3
4 SWSInteractionViewpoint:
5   'Agent-SemanticWebServiceInteractionViewPoint' name=ID
6   '{'
7     (service+=Service|semanticwebagent+=SemanticWebAgent|sws+=SWS|(plan+=Plan)|(role+=Role)|
8     interface+=Interface|
9     registrationrole+=RegistrationRole|
10    process+=Process|owlowlclass+=ODM_OWLClass|grounding+=Grounding)*
11   '}';
12
13 Plan returns Plan:
14   ('Plan' name=ID (type=STRING)? (description=STRING)? (priority=INT)? ';' ) | PlanSequence;
15
16 PlanSequence returns Plan:
17   reg=SS_RegisterPlanDef find=SS_FinderPlanDef agree=SS_AgreementPlanDef exe=SS_ExecutorPlanDef;
18
19 SS_RegisterPlanDef returns SS_RegisterPlan:
20   'SS_RegisterPlan' name=ID ';' | SS_RegisterPlan;
21 SS_FinderPlanDef returns SS_FinderPlan:
22   'SS_FinderPlan' name=ID ';' | SS_FinderPlan;
23 SS_AgreementPlanDef returns SS_AgreementPlan:
24   'SS_AgreementPlan' name=ID ';' | SS_AgreementPlan;
25 SS_ExecutorPlanDef returns SS_ExecutorPlan:
26   'SS_ExecutorPlan' name=ID ';' | SS_ExecutorPlan;
27
28 SS_RegisterPlan returns SS_RegisterPlanDef:
29   plan=[SS_RegisterPlanDef] '{' ('advertisesInterface' adv+=[Interface] ';'
30   ('advertisesInterface' adv+=[Interface] ';')* '}'
31   'SS_FinderPlan returns SS_FinderPlanDef :
32   (plan=[SS_FinderPlanDef] '{' (('interacts_with' inter+=[SSMatchmakerAgent] ';'
33   ('interacts_with' inter+=[SSMatchmakerAgent] ';')*
34   ('discovers' disc+=[Interface] ';'
35   ('discovers' disc+=[Interface] ';')* '}'
36   'SS_AgreementPlan returns SS_AgreementPlanDef :
37   (plan=[SS_AgreementPlanDef] '{' (('interacts_with' inter+=[SSMatchmakerAgent] ';'
38   ('interacts_with' inter+=[SSMatchmakerAgent] ';')*
39   ('discovers' disc+=[Interface] ';'
40   ('discovers' disc+=[Interface] ';')* '}'
41   'SS_ExecutorPlan returns SS_ExecutorPlanDef :
42   (plan=[SS_ExecutorPlanDef] '{' (('interacts_with' inter+=[SSMatchmakerAgent] ';'
43   ('interacts_with' inter+=[SSMatchmakerAgent] ';')*
44   ('discovers' disc+=[Interface] ';'
45   ('discovers' disc+=[Interface] ';')* '}'

```

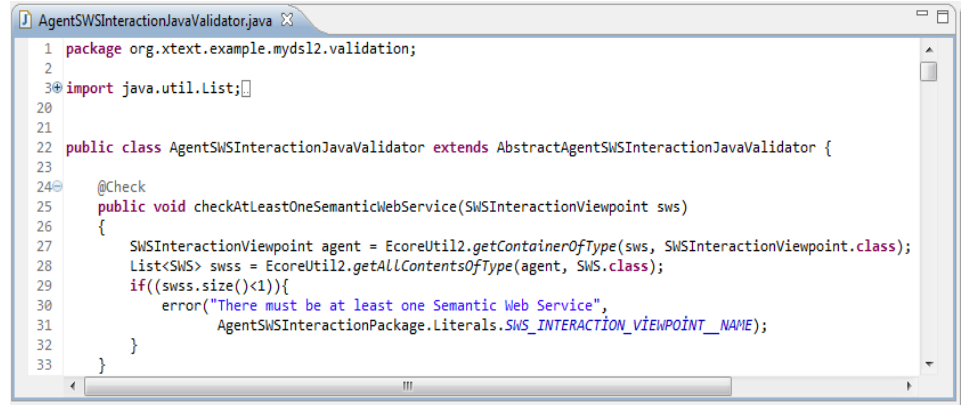
Şekil 6.1. Etmen-Anlamsal Web servis etkileşim bakış açısına ait metinsel sözdizimin bir kısmı.

Belirtilen sıralama kısıtlamalarına ek olarak, Görsel Somut Sözdizimde uygulanan tüm kısıtlar Metinsel Somut Sözdizimde de uygulanmıştır. Örneğin Agent-SWS Interaction bakış açısı için oluşturulan örnek bir modelde en az bir SemanticWebAgent ve en az bir SemanticWebService bulunmak zorundadır. Kullanıcı bu üstvarlıkları tanımlamadığı sürece sistemde hata oluşmaktadır.

Şekil 6.1’de gösterildiği gibi tüm üstvarlıklar belirli etiketler içerisinde tanımlanmaktadır. 7. ve 10. satırlar arasındaki tüm kurallar bu etiket içerisinde istenilen sırada ve istenilen miktarda kullanılabilir. Plan kuralının içeriği 13. satırda yer almaktadır. Kullanıcı doğrudan bir Plan tanımlayabilir ya da “PlanSequence” seçeneği sayesinde Anlamsal Web Servis Planlarını (SS_RegisterPlan, SS_FinderPlan, SS_AgreementPlan ve SS_ExecutorPlan) 17.satırda belirtilen sırada kullanabilir. Bir Plan için sadece “name” özniteliği tanımlanabileceği gibi tip (“type”), tanım (“description”) ve öncelik (“priority”) gibi öznitelikler de tanımlanabilir. 14.satırda yer alan “?” karakteri bu özniteliklerin seçimlik olduğunu belirtmektedir. EBNF kuralları ile kalıtım

özellği 14.satırdaki gibi sağlanmaktadır. Anlamsal Web Servis Planları, yalnızca Plan kuralından türetilir. Bir üstvarlık ve bu üstvarlığın ilişkili olduğu diğer üstvarlıkların tanımı Şekil 6.1'deki gibi yapılmaktadır. Kurallarda belirtilen ‘|’ karakteri “ya da” anlamına gelirken, ‘*’ karakteri kapsadığı elemanlardan istenilen miktarda kullanılabileceğini belirtmektedir. EBNF kuralları tanımlandıktan sonra oluşturulan dosya çalıştırılarak DSL için gerekli ilk adım tamamlanmıştır.

SEA_L üstmodeli dikkate alındığında, her bir örnek modelde en az bir SWA ve en az bir SWS olması gerekmektedir. Her bir SWS'nin en az bir “Grounding”, “Interface” ve “Process” elemanlarına ihtiyacı olduğu için (Şekil 4.7) örnek modelde bu üstvarlıkların her birinin birer örneği oluşturulmak zorundadır. Bu sebeple “Validation Package” içerisine bu kısıtları sağlayacak kurallar yazılmıştır. Her bir örnek modelde en az bir SWS olmasını sağlayan “checkAtLeastOneSemanticWebService” kuralı Şekil 6.2’de verilmektedir.

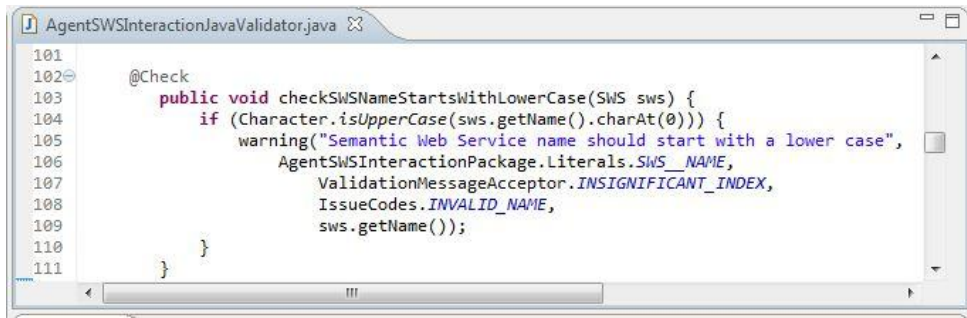


Şekil 6.2. “checkAtLeastOneSemanticWebService” kuralı.

Kısıt tanımlamalarının yapılacağını belirten “@Check” ifadesi, kendinden sonraki kodların örnek model üzerinden kontrol edileceğini belirten bir anahtar kelimedir. 27. ve 28.satırlarda örnek model üzerindeki her bir SWS elemanı bulunur ve bu elemanlar “swss” isimli bir diziye aktarılır. Daha sonra (29.satırdan itibaren) “swss” dizisinin eleman sayısı 1 den küçük olduğu takdirde kullanıcıya en az bir SWS elemanının bulunması gerektiğini belirten “There must be at least one Semantic Web Service” hatası verilir. Şekil 6.2’de belirtilen kontrol, SWA,

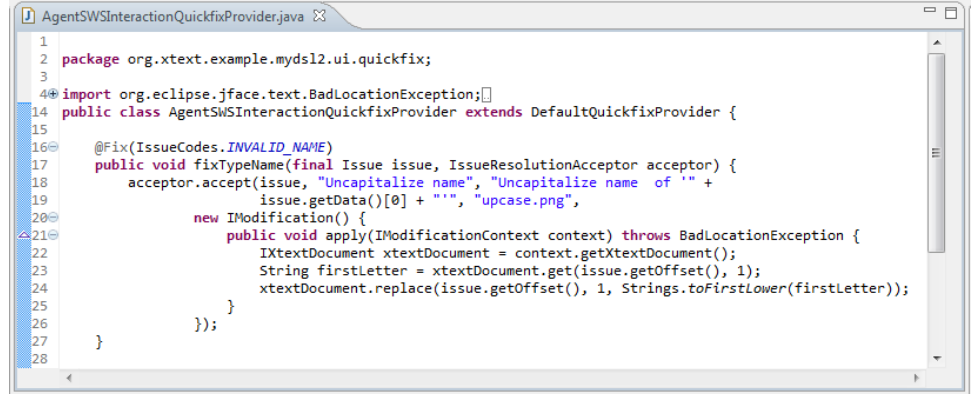
“Grounding”, “Process”, “Interface” ve “WebService” elemanları için de benzer şekilde gerçekleştirilmiştir.

Modelleme sırasında kullanıcıya getirilen bir diğer kısıt üstvarlıkların isimlendirilmesi ile ilgilidir. Üstvarlık adları büyük harfle başladığı için karışıklığı önlemek amacıyla kullanıcının büyük harfle başlayan değişkenler tanımlaması kısıtlanmıştır. Büyük harfle başlayan her bir değişkenin yanında uyarı (“warning”) işareti belirmektedir. Şekil 6.3’te bu kısıtlamanın SWS elemanı için gerçekleştirimi gösterilmektedir.



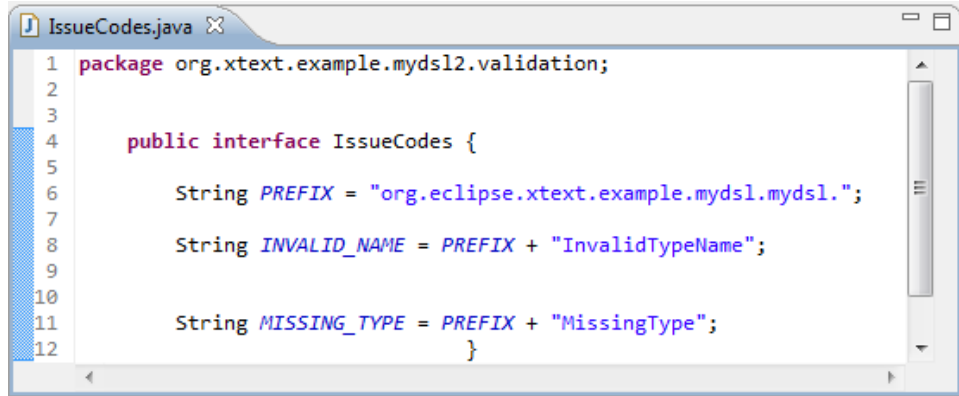
Şekil 6.3. “checkSWSNameStartsWithLowerCase” kuralı.

Şekil 6.3’te örnek eleman isminin ilk harfi kontrol edilmektedir. Elemanın ismi büyük harfle başladığı takdirde, bu eleman isminin küçük harfle başlaması gerektiğini belirten “Semantic Web Service name should start with a lower case” uyarı işareti belirecektir. Aracın kullanıcı için daha kullanışlı hale getirilmesi amacıyla bu tür kurallara hızlı düzeltme (“quick fix”) tanımlanmıştır. Bu sayede kullanıcı uyarı üzerine çift tıklayarak değişkeninin ilk harfini küçük harfe çevirir ve uyarı ortadan kalkar. Şekil 6.4’te büyük harfle başlayan değişken adlarını kolayca küçük harfe çevirmek için tanımlanan hızlı düzeltme verilmektedir.



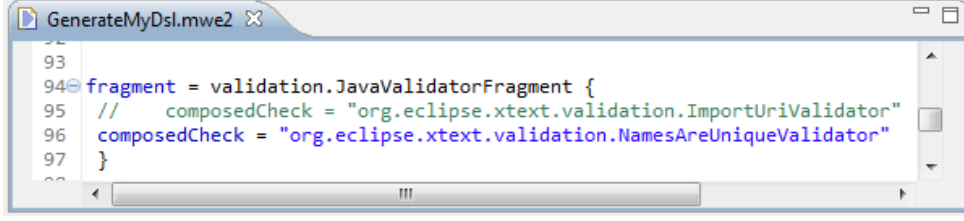
Şekil 6.4. “Name must be lower case” uyarısı için tanımlanan hızlı düzeltme.

Hızlı düzeltme tanımlamasına geçmeden önce, “IssueCodes” isimli bir java dosyası oluşturularak sıkça kullanılacak olan önek (“*prefix*”), geçersiz isim (“*invalid_name*”) ve eksik tip (“*missing_type*”) değişkenleri bu dosyada tanımlanmıştır. Değişken tanımlamaları, Şekil 6.5’te verilmiştir. Böylece proje içerisindeki diğer dosyalardan bu değişkenlere kolayca ulaşılmıştır.



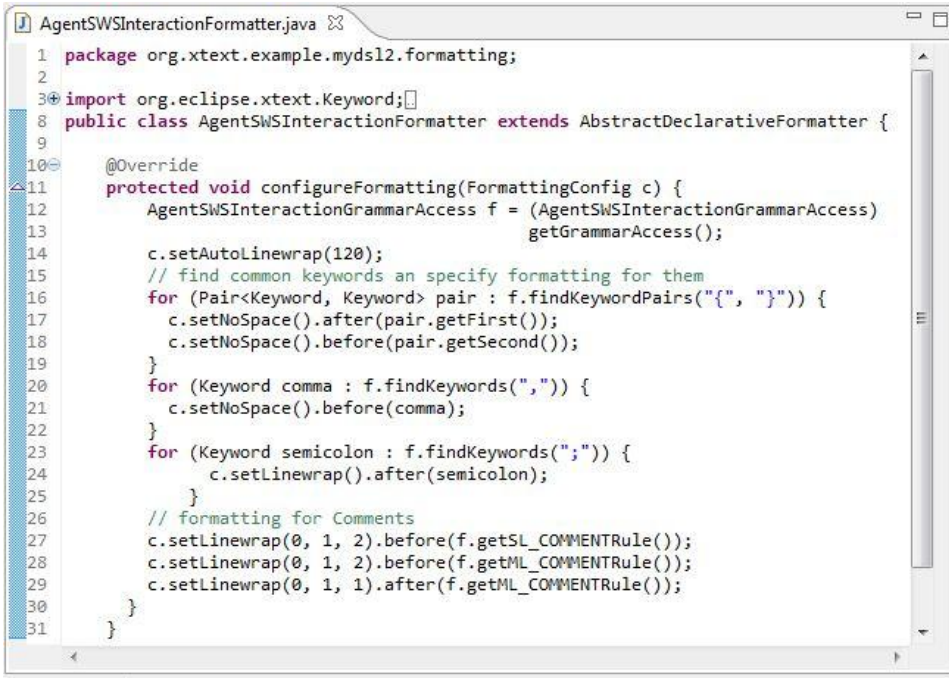
Şekil 6.5. “IssueCodes.java” dosyası.

SEA_L üstmodeline dayalı olarak farklı bakışaçıları için yukarıda anlatılanlara benzer kısıtlar tanımlanmıştır. Örneğin etkileşim bakışaçısı ele alındığında, bir etkileşim üstvarlığının en az bir mesaj üstvarlığını içerdiği görülmektedir. Bu nedenle bu bakışaçısı için örnek modelde bir etkileşim varsa mutlaka bir mesaj da olmalıdır. Üstmodel kısıtlamalarının dışında, her bir bakışaçısındaki elemanlar için isim özniteliklerine boş değer atanmaması gibi kurallar da tanımlanmıştır. Değişken isimlerinin birden fazla tanımlanmaması için Şekil 6.6’da belirtilen kural eklenmiştir.



Şekil 6.6. Bir değişkenin birden fazla kez tanımlanmaması için yazılan kod parçası.

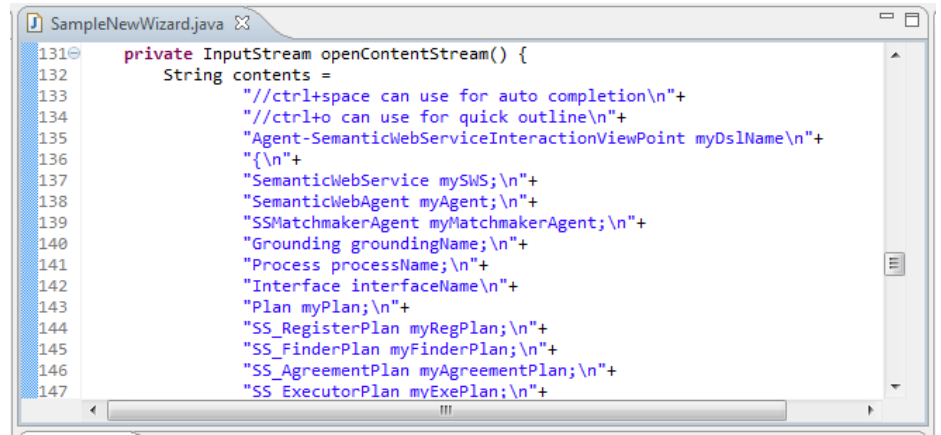
Model oluşturma sırasında kullanıcıya kolaylık sağlaması için yapılan format tanımlamaları Şekil 6.7’de verilmektedir. 16 ve 18.satırlar arasında araçta her bir ‘{’ karakterinin karşılığı olan ‘}’ karakterinin belirlenmesi, ‘{’ karakterinden sonra ve ‘}’ karakterinden önce boşluk olmayacak şekilde hizalanması sağlanmıştır. 20 ve 21.satırlar ‘,’ karakterinden önce boşluk olmaması, 23 ve 24.satırlarda ise ‘;’ karakterinden sonra 14.satırda belirtilen miktar kadar boşluk bırakılması tanımlanmıştır.



Şekil 6.7. “AgentSWSInteractionFormatter.java” dosyası.

Hazırlanan DSL’nin kullanımıyla örnek bir model oluşturulurken üstmodelde belirtilen kısıtlar ve elemanlar arasındaki ilişkiler dikkate alınmalıdır. Kullanıcı üstmodeli bilmek zorunda olmadığı için kendi modelini oluştururken bazı zorluklarla karşılaşabilir. Bu zorlukları önlemek amacıyla kullanıcı kendi

modelini oluşturacağı dosyayı açtığında karşısına örnek bir model çıkmaktadır. Böylece bu örnek modele eklemeler ya da model üzerinde değişiklikler yaparak kullanıcı kendi modelini kolaylıkla oluşturabilmektedir. Bu özelliği sağlamak için bir “New File Wizard” oluşturulmuştur. Kullanıcının oluşturacağı yeni dosyanın ismi “AgentSWSInteraction” olarak ve dosya uzantısı da “.sws” olarak belirlenmiştir. Oluşturulan dosyada ekrana gelecek olan örnek modeli tanımlamak için, “SampleNewWizard.java” dosyasına Şekil 6.8’deki tanımlamalar yapılmıştır. Tüm tanımlamalar yapıldıktan sonra, Xtext dosyası çalıştırılarak DSL’nin son hali oluşturulmuştur.



Şekil 6.8. “SampleNewWizard” dosyasının bir kısmı.

Xtext aracının kullanımıyla oluşturulan örnek modelden, Xpand (Xpand, 2009) aracının kullanımıyla kod üretilmiştir. Xpand EMF modellerini kullanarak metinsel çıktılar oluşturulmasına olanak tanır. Xpand ile modelden metine dönüşüm yapabilmek için EMF ile tanımlanmış bir üstmodele ve dönüşümü sağlamak amacıyla kullanılan en az bir taslak dosyasına ihtiyaç vardır. Xpand aracı Eclipse eklentisi olarak kullanılmaktadır.

Xpand dosyası Xtext projesinin üretilmesiyle birlikte otomatik olarak oluşur. Xpand dosyası, taslak dosyası olduğunu belirten (.xpt) uzantısına sahiptir. Xpand kod üretimi için güçlü bir araç olmasının yanında, anlamsal kontrollerin yapılması için kullanılacak olan Xtend ile de uyumludur.

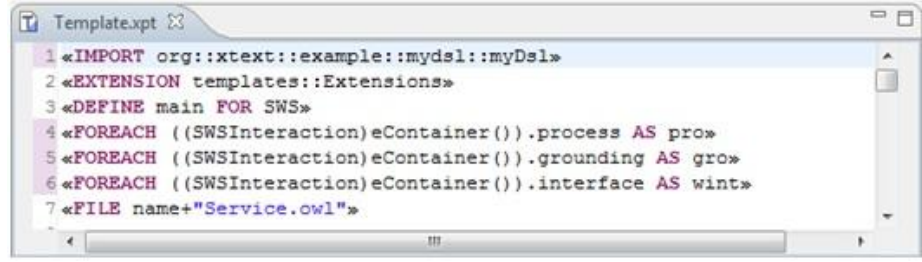
SEA_L üstmodeli göz önünde bulundurularak oluşturulan örnek modeller JADEX (Jadex, 2003), OWL (W3C, 2004), ve OWL-S (Martin et al., 2004) üst modellerine uygun hale getirilerek modelden metne dönüşüm işlemi sağlanmıştır. SEA_L üstmodeli ile JADEX, OWL ve OWL-S üstmodellerindeki elemanlar arasındaki eşleştirmeler Tablo 6.1’de verilmiştir.

Tablo 6.1. SEA_L üstmodel elemanlarının JADEX ve OWL-S üstmodel elemanları ile eşleştirilmesi.

SEA_L	JADEX	OWLS
SemanticWebAgent	Agent	
SSMatchmakerAgent	Agent	
Plan, Behavior	Plan	
Capabilities	Capability	
Goal	AchieveGoal	
Goal	QueryGoal	
Goal	PerformGoal	
SS_AgreementPlan	Plan	
SS_ExecutorPlan	Plan	
SS_FinderPlan	Plan	
SS_RegisterPlan	Plan	
SemanticWebService		Service
Interface		ServiceProfile
Process		ServiceModel
Grounding		ServiceGrounding
Input		Input
Output		Output
Precondition		Condition
Effect		ResultVar

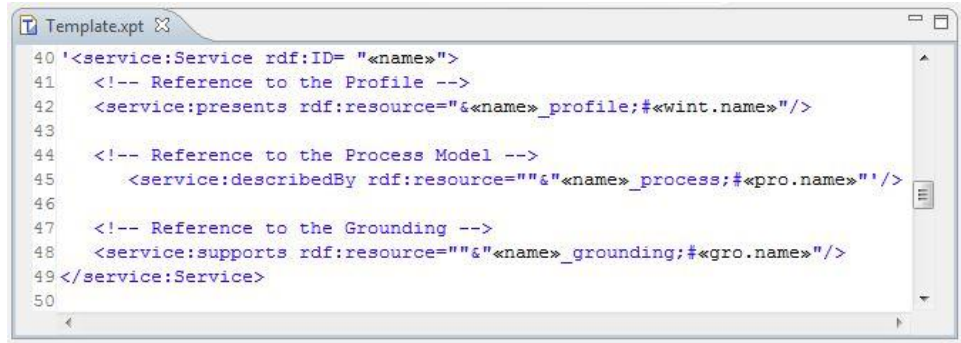
Hazırlanan araç üzerinden kod üretiminin nasıl gerçekleştirildiğini göstermek amacıyla etmen-Anlamsal Web servis etkileşim bakış açısı için oluşturulan taslak dosyası örneklendirilmiştir.

Kod üretimi için Xtext projesinin içerisinde bulunan “Template.xpt” dosyası modifiye edilmiştir. Bu dosya kendi içinde taslaklardan oluşur. Xpand’de tüm anahtar kelimeler ve tanımlamalar “«” ve “»” karakterlerinin arasında yapılmaktadır. “Define” kelimesinin kullanımıyla başlayan kurallar bir taslak çağırıldığını gösterir. Burada taslaklar Java dilindeki metodlar gibi düşünülebilir. Şekil 6.9’da görüldüğü gibi, taslak dosyasında yapılacak ilk iş üstmodelin editöre tanıtılmasıdır.



Şekil 6.9. “Service.owl” dosyasının oluşturulması.

Her bir SWS için bir WSDL dosyası ve dört OWL dosyası (service.owl, profile.owl, process.owl, grounding.owl) oluşturulacaktır. “Interface”, “Process” ve “Grounding” elemanları sırasıyla “profile.owl”, “process.owl” ve “grounding.owl” dosyalarında tanımlanmaktadır. Şekil 6.9’da taslak dosya içerisinde kullanılacak olan bu elemanlar sırasıyla “pro”, “gro” ve “wint” değişkenlerine atanmıştır. Şekil 6.10’da “Interface”, “Process” ve “Grounding” elemanları için “service.owl” dosyasından sırasıyla “service profile”, “process model” ve “service grounding” dosyalarına referans verilmiştir.



Şekil 6.10. Service profile, service process ve service grounding dosyalarına referansların tanımlanması.

Soyut sözdizimde belirtilen tüm bakışaçıları için taslak dosyaları hazırlanmıştır. Anlamsal Web etmeni içsel bakışaçısından oluşturulan bir örnek model üzerinden kod üretimi yapıldığında, her bir SWA için bir ADF dosyası ve her bir “Plan” için bir Java Plan dosyası oluşmaktadır. Bu nedenle “plans” ve “semanticwebagents” adı verilen taslaklar, aşağıda gösterildiği gibi ana taslaktan çağırılmaktadır.

```

01 «IMPORT
02   org::xtext::example::agentinternal::agentInternal»
03 «DEFINE main FOR AgentInternalViewpoint»
04 «EXPAND plans FOREACH plan»
05 «EXPAND semanticwebagents
06     FOREACH semanticwebagent»
07 «ENDDEFINE»

```

Aşağıda belirtilen kod bloğu, oluşturulan ADF dosyasında elemanların nasıl tanımlandığını “belief” üst elemanı örneği ile göstermektedir. “Belief”ler <beliefs> etiketinin içinde tanımlanmaktadırlar. “Belief” üstelemanının öznitelikleri 3, 4 ve 5. satırlarda gösterilmektedir.

```

01 «DEFINE beliefs FOR Belief»
02 <beliefs>
03   name=«this.name»
04   description=«this.description»
05   dynamic = «this.dynamic»
06 </beliefs>
07 «ENDDEFINE»

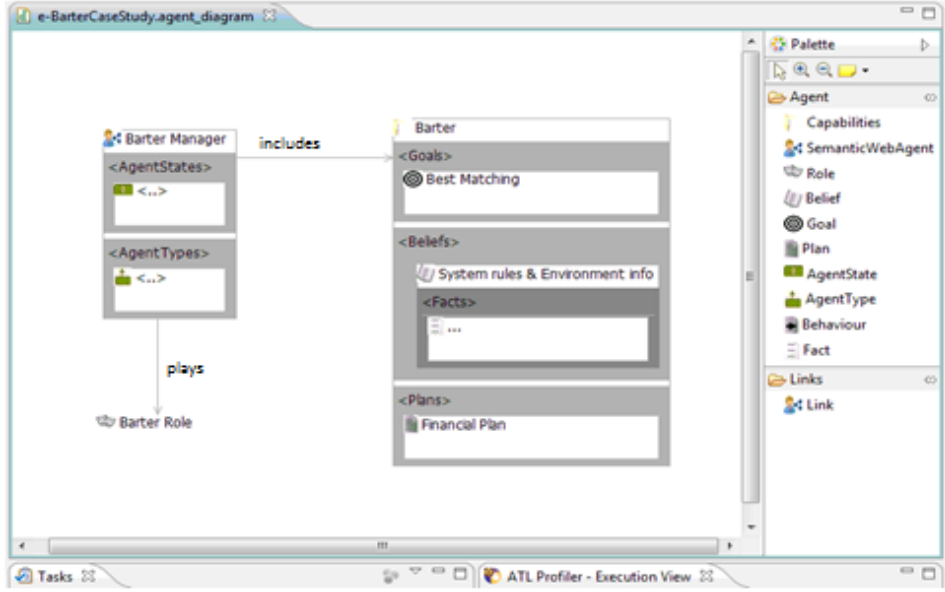
```

7. GRAFİKSEL SÖZDİZİMİN DEĞERLENDİRİLMESİ

Hazırlanan SEA_ML üstmodelinin ve modelleme aracının etmen tabanlı sistemlerin geliştirilmesinde kullanımını örneklemek amacıyla, araç üzerinde etmen tabanlı elektronik takas sistemi (“*electronic barter system*”) örneği modellenmiştir. Anlamsal Web etmeni içsel ve etmen-Anlamsal Web servis etkileşim bakışaçıları için geliştirilen modellerin ekran çıktıları verilmiştir.

Bir takas sistemi, bir malın veya hizmetin herhangi bir para kullanımı olmadan alışverişini sağlayan alternatif bir ticaret yaklaşımıdır. Para kullanımı yerine mal veya hizmetlerin takası söz konusudur. Etmen tabanlı bir elektronik takas sistemi, kullanıcıları adına ve kullanıcıların tercihleri doğrultusunda takas işlemini yürüten etmenleri içermektedir. Müşteri (“*customer*”) etmenleri takas önerisi sunmak veya değerlendirmekten sorumludur. Takas yöneticisi (“*barter manager*”) etmeni ise takas önerilerini toplamaktan, uygun takas önerilerini eşlemekten, alıcı ve satıcı etmenler arasındaki alışveriş protokolünden sorumludur (Kardaş et al., 2010). Alıcı ve satıcı rolündeki müşteri etmenleri arasında belirli kısıtlamalar doğrultusunda alışveriş protokolü gerçekleşmektedir. Bu kısıtlar, her etmenin kendine ait kanı-tabanında takas için belirlediği üst ve alt sınırları içerir. Belirtilen sınırlar arasındaki değere ulaşamayan müşteriler, alışveriş başarılı bir şekilde sonlandıramadıklarını takas yöneticisi etmenine haber verir. Etmen tabanlı takas sisteminin işleyişi daha ayrıntılı olarak (Demirkol et al., 2011)’de yer almaktadır.

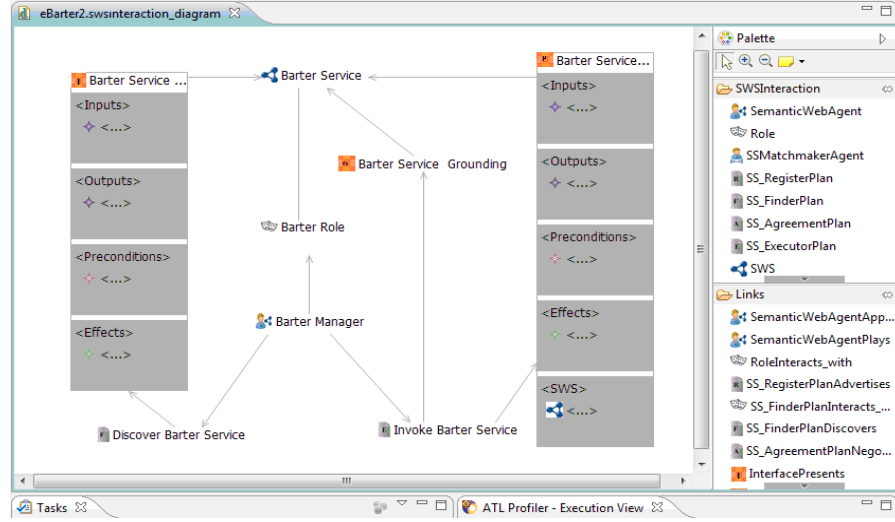
Anlamsal Web etmeni içsel bakışaçısı için oluşturulan örnek model Şekil 7.1’de gösterilmektedir. Anlamsal Web etmeninin bir örneği olan takas yöneticisi, üstmodele uygun olarak bir takas yeteneğini içermektedir. Takas yeteneği en iyi eşleme (“*best matching*”) amacını, finansal planı (“*financial plan*”) ve sistem kurallarını (“*rules*”) içerir. Takas yöneticisi, amacını gerçekleştirmek için takas rolünü (“*barter role*”) oynar.



Şekil 7.1. Takas yöneticisinin Anlamsal Web etmeni içsel bakış açısına göre GMF tabanlı editörde modellenmesi.

Şekil 7.2’de takas yöneticisi ile Anlamsal Web servis elemanının bir örneği olan takas servisi (“*barter service*”) arasındaki etkileşim gösterilmektedir. Palet üzerinde tüm üstvarlık elemanları yer alırken, örnek modelde takas sisteminin modellenmesi sırasında ihtiyaç duyulan Anlamsal Web etmeni, anlamsal servis bulucu plan, anlamsal servis çalıştırıcı plan, rol, “Interface”, “Process”, “Grounding” ve Anlamsal Web servis elemanları yer almaktadır.

Takas yöneticisi, servis bulucu plan örneği olan takas sistemini bul (“*discover barter service*”) ve anlamsal servis çalıştırıcı plan örneği olan takas servisini çağır (“*invoke barter service*”) planlarını çalıştırmaktadır. Anlamsal Web etmeni, takas işlemleri için takas servisini kullanır. Bu servisi kullanmak için Anlamsal Web servis arayüzüne ve çalıştırma mekanizmasına ihtiyaç duyar.

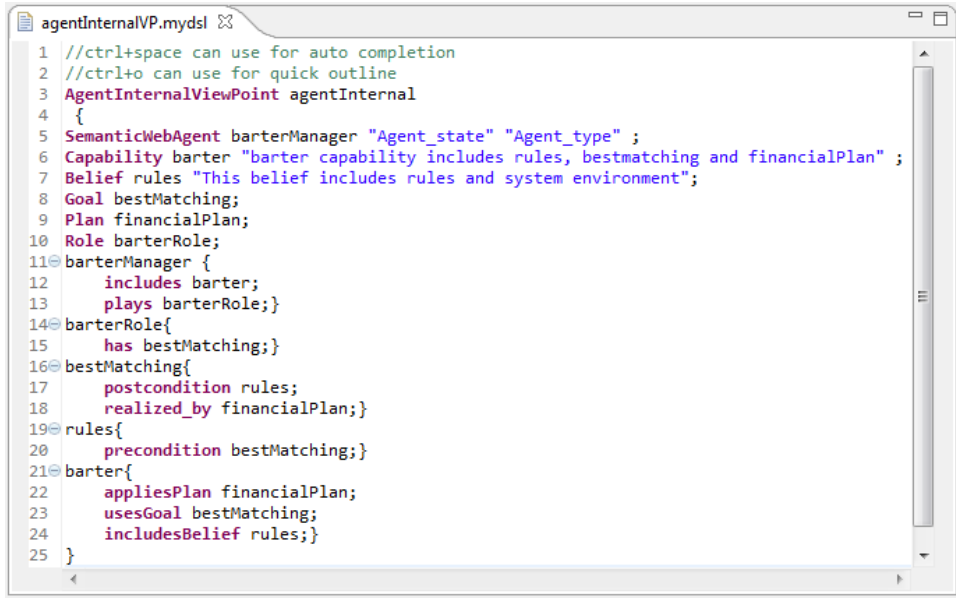


Şekil 7.2. Takas yöneticisinin etmen-Anlamsal Web servis etkileşim bakış açısına göre GMF tabanlı editörde modellenmesi.

TÜBİTAK projesi kapsamında, bu tezde hazırlanan grafiksel somut sözdizimiyle oluşturulan örnek model üzerinden, modelden modele ve modelden metine dönüşümler tanımlanarak kod üretimi gerçekleştirilmiştir.

8. METİNSEL SÖZDİZİMİN DEĞERLENDİRİLMESİ

Bir önceki bölümde anlatılan etmen tabanlı elektronik takas sistemi, Anlamsal Web etmeni içsel bakış açısı ve etmen-Anlamsal Web etmeni etkileşim bakış açısı için hazırlanan araç üzerinden metinsel olarak tanımlanmıştır. Anlamsal Web etmeni içsel bakış açısını oluşturmak için öncelikle, Xtext projesi üzerinden yeni bir Eclipse uygulaması açılarak “Sample Wizards” seçeneğinden kullanılacak olan bakış açısı seçilmiştir. Böylece, oluşturulacak örnek modelin taslağı ekrana gelmiştir. Taslak üzerinden etmen tabanlı takas sistemi için gerekli elemanlar ve ilişkiler tanımlanmıştır. Anlamsal Web etmeni içsel bakış açısı için oluşturulan örnek takas sistemi Şekil 8.1’de verilmiştir.



Şekil 8.1. Takas yöneticisinin Anlamsal Web etmeni içsel bakış açısına göre Xtext tabanlı editörde modellenmesi.

Şekil 8.1, bir takas yöneticisi etmeninin içsel yapısının Anlamsal Web etmeni içsel bakış açısına göre nasıl modellenebileceğini örneklemektedir. Bir takas yöneticisi, Anlamsal Web etmeninin bir örneğidir. Takas yöneticisi tarafından ihtiyaç duyulan amaç, rol, yetenek gibi bileşenlerin SEA_L somut sözdizim elemanlarından plan, yetenek, amaç, rol ve inanç ile ilişkileri belirlenmiştir. İlişkiler Şekil 4.2’de verilen üstmodele uygun olarak oluşturulmuştur. Örneğin, takas yöneticisi bir takas rolü oynar ve bir takas

yeteneği, en iyi eşleme gibi bir amaç ile ilişkilendirilebilirken bir takas rolü ile ilişkilendirilemez.

Örnek model oluşturulduktan sonra taslak dosyasının çalıştırılmasıyla otomatik kod üretimi gerçekleştirilmiştir. Anlamsal Web etmeninin bir örneği olan takas yöneticisi için bir JADEX ADF dosyası ve bu etmenin kullandığı finansal plan için bir Java dosyası otomatik olarak oluşturulmuştur. Oluşturulan bu kodlar JADEX ortamında doğrudan çalıştırılabilir kodlardır. Oluşturulan ADF dosyasının bir bölümü aşağıda verilmiştir:

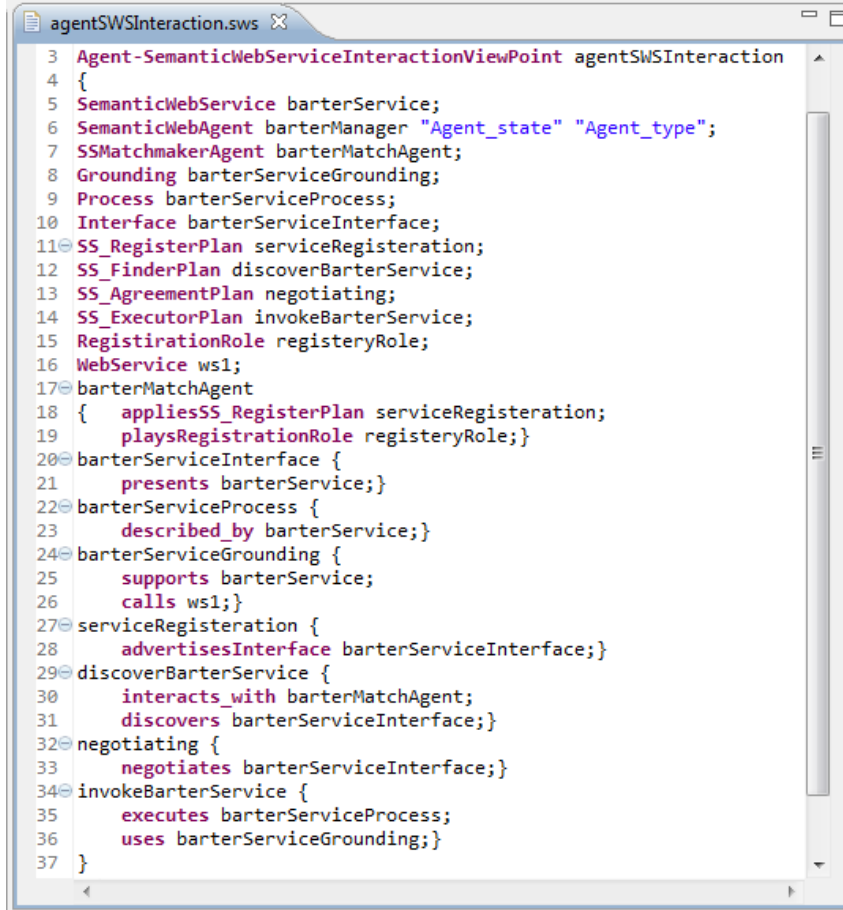
```

01 <agent xmlns="http://jadex.sourceforge.net/jadex"
02   xmlns:xsi="http://www.w3.org/2001/XMLSchema-
03   instance" xsi:schemaLocation="http://jadex.sourceforge.net
04   /jadex http://jadex.sourceforge.net/jadex-2.0.xsd"
05   name="barterManager" agent_state="Agent_state"
06   agent_type="Agent_type">
07   <capabilities>
08     <capability>
09       name="barter" file="" description="barter capability
10 includes rules, bestmatching and financialPlan"
11     </capability>
12   </capabilities>
13   <plans> name="financialPlan"
14     description="financial plan description" priority="1"
15   </plans>
16   <beliefs> name="rules"
17     description="This belief includes rules and system
18 environment"
19     dynamic="1"
20   </beliefs>
21   <goals>
22     <achievegoal name="bestMatching"
23       exclude="when_tried" recalculate="true" retry="true"
24       exported="false" posttoall="false" recurdelay="0"
25       randomselection="false"
26       ...
27       retrydelay="0">
28       <creationcondition> <!-- Write Creation Condition -->
29     </creationcondition>
30     ...
31   </achievegoal>
32 </goals>
33 <events> </events>
34 <properties> </properties>
35 <configurations> </configurations>
36 </agent>

```

Elektronik takas sisteminin bir bakış açısı da, Anlamsal Web etmeni içsel bakış açısı için hazırlanan örneğe benzer şekilde etmen-Anlamsal Web servisi etkileşim bakış açısı için oluşturulmuştur. Şekil 8.2’de takas yöneticisinin Anlamsal Web servisleri ile olan etkileşiminin sunulan araç ile etmen-Anlamsal

Web servis etkileşim bakış açısına uygun bir biçimde nasıl modellenebileceği gösterilmiştir. Takas sisteminde, takas yöneticisi etmeni takas için önerilen malların eşlenmesi ve değerinin belirlenmesi amacıyla bir Anlamsal Web servisi ile etkileşime geçmektedir.



Şekil 8.2. Takas yöneticisinin etmen-Anlamsal Web servis etkileşim bakış açısına göre Xtext tabanlı editörde modellenmesi.

Sözü edilen tüm etkileşim, öne sürdüğümüz SEA_L’in metinsel somut sözdizimi ile oluşturulmuştur ve bu örnek model üzerinden kod üretimi gerçekleştirilmiştir. Taslak dosyasının çalıştırılması ile otomatik olarak oluşan OWL-S dosyasının bir kısmı aşağıda verilmiştir. 23, 27 ve 31. satırlarda sırasıyla “profile.owl”, “process.owl” ve “grounding.owl” dosyaları için tanımlanan anlamsal web servisi belirtilmiştir.

```

01 <?xml version="1.0"?>
02 <rdf:RDF xmlns:rdf= "&rdf;#" xmlns:rdfs="&rdfs;#"
03   xmlns:owl= "&owl;#" xmlns:service= "&service;#"
04   ...
05   xml:base="&DEFAULT;" >
06   <owl:Ontology rdf:about="">
07     <owl:versionInfo>
08     </owl:versionInfo>
09     <rdfs:comment> "This ontology represents the OWL-S
10       service description for the"+ barterService +
11       "service example."
12     </rdfs:comment>
13     <owl:imports rdf:resource="&service;" />
14     ...
15   </owl:Ontology>
16   <service:Service rdf:ID= "barterService">
17     <!-- Reference to the Profile -->
18     <service:presents rdf:resource="&'barterService_profile;
19       # barterServiceInterface'"/>
20     ...
21   </service:Service>
22   <!-- Inverse links -->
23   <profile:Profile rdf:about=&
24     "barterService_profile;# barterServiceInterface">
25     <service:presentedBy rdf:resource=#"barterService"/>
26   </profile:Profile>
27   <process:AtomicProcess rdf:about=&
28     "barterService_process;# barterServiceProcess">
29     <service:describes rdf:resource=#"barterService"/>
30   </process:AtomicProcess>
31   <grounding:WsdIGrounding rdf:about=&
32     "barterService_grounding;# barterServiceGrounding">
33     <service:supportedBy rdf:resource=#"barterService"/>
34   </grounding:WsdIGrounding>
35 </rdf:RDF>

```

9. SONUÇ

Daha önce MAS'lar için geliştirilmiş olan dillerin Anlamsal Web yetenekli olmaması, MAS geliştiricileri için bir alana özgü dil ihtiyacını doğurmuştur. Bu tez çalışmasında, Anlamsal Web yetenekli MAS'lar için geliştirilen bir alana özgü dilin somut sözdizimi hem metinsel olarak hem grafiksel olarak tanımlanmıştır. Her iki somut sözdizim üzerinden örnek modeller oluşturulmuştur. Metinsel olarak oluşturulan örnek model üzerinden kod üretimi gerçekleştirilmiştir.

Bu tez çalışmasında hem grafiksel hem metinsel sözdizimin oluşturulması, her iki sözdizimin de kendine göre avantajlarının olmasından kaynaklanmıştır. Metinsel sözdizim, kısıtların daha kolay tanımlanabilmesi ve örnek bir taslak ile kullanıcıya kolaylık sağlaması açısından daha avantajlıyken, grafiksel sözdizimin görsel arayüzü sayesinde kullanıcı için daha kullanışlı olduğu düşünülebilir.

TÜBİTAK tarafından desteklenen bir proje bünyesinde ortaya konan tez çalışmasında, etmen sistemlerinin görsel ve metinsel bir şekilde tanımlanması için tam bir sözdizim oluşturulmuştur. Projenin devamında, dilin anlamsallığı sağlanarak model dönüşümleri aracılığıyla etmen tasarım modellerinden JADEX ortamında işletilebilir yazılım kodları otomatik olarak elde edilmiştir.

Bu tezin devamı olabilecek çalışmalar, bu tezde tanıtılan DSL ve DSML sözdizimine uygun geliştirilen örnek etmen modellerinden JADE (Bellifemine et al., 2001) ve JACK (Howden et al., 2001) gibi farklı etmen geliştirme çerçevelerine otomatik kod üretimi gerçekleştirilebilir.

KAYNAKLAR DİZİNİ

- Afşar, B., Kardaş, G., Topaloğlu, N.Y. ve Dikenelli, O.,** 2011, JADDEX Kanı-İstek-Hedef Etmenlerinin Model GÜdümlü Geliştirilmesi, Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 4:9-18ss.
- AOS,** 2001, Agent Oriented Software Pty. Ltd., JACK Environment, <http://www.aosgrp.com/products/jack/>, (Erişim tarihi: Haziran, 2012).
- Bauer, B., Muller, J.P. and Odell, J.,** 2001, Agent UML: A formalism for specifying multiagent software systems, International Journal of Software Engineering and Knowledge Engineering, 11(3):207-230pp.
- Bellifemine, F., Poggi, A and Rimassa, G.,** 2001, Developing Multi-Agent Systems with a FIPA-compliant Agent Framework, Software: Practice and Experience, 31(2):103-128pp.
- Bellifemine, F., Caire G. and Greenwood, D.,** 2007, Developing Multi-Agent Systems with JADE, A Practical Guide, John Wiley and Sons, 230p.
- Berners-Lee, T., Hendler, J. and Lassila, O.,** 2001, The Semantic Web, Scientific American, 284(5):34-43pp.
- Bernon, C., Cossentino, M., Gleizes, M-P., Turci, P. and Zambonelli, F.,** 2005, A Study of some Multi-Agent Meta-Models, Lecture Notes in Computer Science, 3382:62-77pp.
- Bernon, C., Gleizes, M-P., Peyruqueou, S. and Picard, G.,** 2003, ADELFE: A methodology for adaptive multi-agent systems engineering, Lecture Notes in Artificial Intelligence, 2577:70-81pp.
- Beydoun, G., Low, G.C., Henderson-Sellers, B., Mouratidis, H., Gomez-Sanz, J.J., Pavon, J. and Gonzalez-Perez, C.,** 2009, FAML: A Generic Metamodel for MAS Development, IEEE Transactions on Software Engineering, 35(6):841-863pp.
- Bratman, M.E.,** 1987, Intention, Plans, and Practical Reason, Harvard University Press, Cambridge, 200p.

KAYNAKLAR DİZİNİ (devam)

- Cervenka, R., Trencansky, I., Calisti, M. and Greenwood, D.,** 2005, AML: Agent Modeling Language – Toward Industry-Grade Agent-Based Modeling, Lecture Notes in Computer Science, 3382:31-46pp.
- Challenger, M., Getir, S., Demirkol, S. and Kardaş, G.,** 2011, A Domain Specific Metamodel for Semantic Web enabled Multi-agent Systems, Lecture Notes in Business Information Processing, 83:177-186pp.
- Cossentino, M. and Potts, C.,** 2002, A CASE tool supported methodology for the design of multi-agent systems, In proceedings of the International Conference on Software Engineering Research and Practice (SERP'02), Las Vegas, 315-321pp.
- Demirkol, S., Getir, S., Challenger, M. and Kardaş, G.,** 2011, Development of an Agent based E-barter System, In proceedings of 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), Istanbul, Turkey, IEEE Computer Society, 193-198pp.
- Depke, R., Heckel, R. and Kuster J.M.,** 2001, Agent-Oriented Modeling with Graph Transformations, Lecture Notes in Computer Science, 1957:105-119pp.
- Eclipse ATL,** <http://www.eclipse.org/atl/>, (Erişim tarihi: Haziran, 2012).
- Eclipse EMF,** <http://www.eclipse.org/modeling/emf/>, (Erişim tarihi: Haziran, 2012).
- Eclipse GMF,** <http://www.eclipse.org/modeling/gmf/>, (Erişim tarihi: Haziran, 2012).
- Erdur, R.C.,** 2001, Yazılım Etmeni Teknolojisinin İnternet Tabanlı Yazılım Yeniden Kullanımına Uygulanması, Doktora Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, 213s.

KAYNAKLAR DİZİNİ (devam)

- Ferber, J. and Gutknecht, O.**, 1998, A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems, In proceedings of the Third International Conference on Multi-Agent Systems, Paris, France, 128-135pp.
- FIPA**, 2002a, FIPA Standartları, <http://www.fipa.org>, (Erişim tarihi: Mayıs, 2012).
- FIPA**, 2002b, FIPA Agent Communication Language Message Structure Specification, <http://www.fipa.org/specs/fipa00061/>, (Erişim tarihi: Mayıs, 2012).
- FIPA**, 2002c, FIPA Agent Communication Language Specifications, <http://www.fipa.org/repository/aclspecs.html>, (Erişim tarihi: Mayıs, 2012).
- Finin, T., Labrou, Y. and Mayfield, J.**, 1997, KQML as an agent communication language, In Bradshaw (Ed): Software Agents, AAAI Press/MIT Press, 291-316pp.
- Fowler, M.**, 2011, Domain-specific Languages, Addison-Wesley Professional, New Jersey, 640p.
- Gascuena, J.M., Navarro, E. and Fernandez-Caballero, A.**, 2012, Model-Driven Engineering Techniques for the Development of Multi-agent Systems, Engineering Applications of Artificial Intelligence, 25(1):159–173pp.
- Georgeff, M.P. and Lansky, A.L.**, 1987, Reactive reasoning and planning. In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), Seattle, WA, 677-682pp.
- Getir, S., Demirkol, S., Challenger, M. and Kardaş, G.**, 2011b, Anlamsal Web Yetenekli Etmenler için Alana Özgü bir Modelleme Diline ait Görsel Sözdizim, 5. Ulusal Yazılım Mühendisliği Sempozyumu (UYMS 2011), Ankara, Türkiye, 93-100ss.

KAYNAKLAR DİZİNİ (devam)

- Getir, S., Demirkol, S., Challenger, M. and Kardaş, G.,** 2011a, The GMF-based Syntax Tool of a DSML for the Semantic Web enabled Multi-Agent Systems, In proceedings of the Workshop on Programming Systems, Languages, and Applications based on Actors, Agents, and Decentralized Control (AGERE!2011), held at the 2nd Systems, Programming, Languages and Applications: Software for Humanity Conference (SPLASH 2011), Portland, USA, ACM Press, 235-238pp.
- Gray, J., Tolvanen, J-P., Kelly, S., Gokhale, A., Neema, S. and Sprinkle, J.,** 2007, Domain-Specific Modeling, In Fishwick (Ed): Handbook of Dynamic System Modeling, CRC Press, Florida, 1-7pp.
- Haag, S., Cummings, M. and McCubbrey, D.J.,** 2003, Management Information Systems for the Information Age – 4th edition, Mc Graw Hill Publication, 550p.s
- Hahn, C. and Fischer, K.,** 2009, The Formal Semantics of the Domain Specific Modeling Language for Multi-agent Systems, Lecture Notes in Computer Science, 5386:145-158pp.
- Hahn, C.,** 2008, A Domain Specific Language for Multiagent Systems, In proceedings of the Seventh Autonomous Agents and Multiagent Systems Conference (AAMAS'08), Estoril, Portugal, 233-240pp.
- Hahn, C., Madrigal-Mora, C. and Fischer, K.,** 2009, A platform-independent metamodel for multiagent systems, Autonomous Agents and Multi-agent Systems, 18(2):239-266pp.
- Hahn, C., Nesbigall, S., Warwas, S., Zinnikus, I., Fischer, K. and Klusch, M.,** 2008, Integration of Multiagent Systems and Semantic Web Services on a Platform Independent Level, In proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008), Sydney, Australia, 200-206pp.

KAYNAKLAR DİZİNİ (devam)

- Howden, N., Ronnquista, R., Hodgson, A. and Lucas, A.,** 2001, Jack intelligent agents: Summary of an agent infrastructure, In Proceedings of the 2nd International Workshop on Infrastructure for Agents, MAS, and Scalable MAS at the 5th International Conference on Autonomous Agents, Montreal, Canada.
- Huget, M-P.,** 2005, Modeling Languages for Multiagent Systems, In proceedings of the 6th International Workshop on Agent-Oriented Software Engineering (AOSE 2005), Utrecht, The Netherlands.
- JADEx,** 2003, JADEx BDI Agent System, <http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/About/Overview>, (Erişim tarihi: Haziran, 2012).
- Kardaş, G.,** 2008, Anlamsal Web Ortamında Çalışan Çok Etmenli Sistemlerin Model Güdümlü Geliştirilmesi, Doktora Tezi, Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü, 205s.
- Kardaş, G., Demirezen, Z. and Challenger, M.,** 2010, Towards a DSML for Semantic Web enabled Multi-agent Systems, In proceeding of the International Workshop on Formalization of Modeling Languages, held in conjunction with the 24th European Conference on Object-Oriented Programming (ECOOP), Maribor, Slovenia, 1-5pp.
- Kardaş, G., Goknil, A., Dikenelli, O. and Topaloglu, N. Y.,** 2007, Modeling the Interaction between Semantic Agents and Semantic Web Services using MDA Approach, Lecture Notes in Artificial Intelligence, 4457: 209-228pp.
- Kardaş, G., Goknil, A., Dikenelli, O. and Topaloglu, N.Y.,** 2009, Model Driven Development of Semantic Web Enabled Multi-agent Systems, International Journal of Cooperative Information Systems, 18(2):261-308pp.
- Kulesza, U., Garcia, A., Lucena, C. and Alencar, P.,** 2005, A Generative Approach for Multi-agent System Development, Lecture Notes in Computer Science, 3390:52-69pp.

KAYNAKLAR DİZİNİ (devam)

- Martin, D., Burstein, M., Hobbs, J. Lassila, O., McDermott, D., McIlraith, S. Narayanan, S. Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N. and Sycara, K.,** 2004, OWL-S: Semantic Markup for Web Services, W3C Member Submission, <http://www.w3.org/Submission/OWL-S/>, (Erişim tarihi: Mayıs, 2012).
- Mernik, M., Heering, J. and Sloane, A.,** 2005, When and how to develop domain-specific languages, ACM Computing Surveys, 37(4):316-344pp.
- Molesini, A., Denti, E. and Omicini, A.,** 2005, MAS Meta-models on Test: UML vs. OPM in the SODA Case Study, Lecture Notes in Artificial Intelligence, 3690:163-172pp.
- Odell, J., Nodine, M. and Levy, R.,** 2005 A Metamodel for Agents, Roles and Groups, Lecture Notes in Computer Science, 3382:78-92pp.
- OMG,** 2000, Unified Modeling Language Specification, <http://www.uml.org/>, (Erişim tarihi: Haziran, 2012).
- OMG,** 2002, Meta Object Facility (MOF), <http://www.omg.org/spec/MOF/>, (Erişim tarihi: Mayıs, 2012).
- OMG,** 2004, UML 2.0 Superstructure Specification, <http://www.omg.org/technology/documents/formal/uml.htm>, (Erişim tarihi: Mayıs, 2012).
- Omicini, A.,** 2000, SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems, Lecture Notes in Computer Science, 1957:185-193pp.
- OWL-S Coalition,** 2004, OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.1/overview/>, (Erişim tarihi: Mayıs, 2012).
- Padgham, L. and Winikoff, M.,** 2004, Developing Intelligent Agent Systems: A Practical Guide, John Wiley and Sons, 230p.

KAYNAKLAR DİZİNİ (devam)

- Padgham, L. and Winikoff, M.**, 2004, Developing Intelligent Agent Systems: A Practical Guide, John Wiley and Sons, 230p.
- Pokahr, A, Braubach, L, Lamersdorf, W.**, 2005, Jadex: A BDI Reasoning Engine. In Bordini et al. (Eds): Multi-Agent Programming Languages, Platforms and Applications, Springer, 149-174pp.
- Rao, A. and Georgeff, M.**, 1995, BDI Agents: From Theory to Practice, In proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, CA, USA., 312-319pp.
- Rougemaille, S., Migeon, F., Maurel, C. and Gleizes, M-P.**, 2007, Model Driven Engineering for Designing Adaptive Multi-Agent Systems, Lecture Notes in Artificial Intelligence, 4995:318-332pp.
- Russell, S.J. and Norvig, P.**, 2003, Artificial Intelligence: A Modern Approach, second ed., Pearson Education, USA, 1080p.
- Schmidt, D.C.**, 2006, Guest Editor's Introduction: Model-Driven Engineering, IEEE Computer, 39(2):25-31pp.
- Smith, R.G.**, 1980, The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver, IEEE Transactions on Computers, 29(12):1104-1113pp.
- Sprinkle, J., Mernik, M., Tolvanen, J.-P. and Spinellis, D.**, 2009, Guest Editors' Introduction: What Kinds of Nails Need a Domain-Specific Hammer?, IEEE Software, 26(4):15-18pp.
- Sycara, K., Paolucci, M., Ankolekar, A. and Srinivasan, N.**, 2003a, Automated discovery, interaction and composition of Semantic Web Services, Journal of Web Semantics, 1(1):27-46pp.
- The Eclipse Foundation**, 2009 JET, <http://www.eclipse.org/modeling/m2t/?project=jet#jet>, (Erişim tarihi: Haziran, 2012).

KAYNAKLAR DİZİNİ (devam)

- van Deursen, A., Klint, P. and Visser, J.**, 2000, Domain-specific languages: an annotated bibliography, ACM SIGPLAN Notices, 35(6):26-36pp.
- Varanda-Pereira, M.J., Mernik, M., Da-Cruz, D. and Henriques, P.R.**, 2008, Program comprehension for domain-specific languages, Computer Science and Information Systems, 5(2):1-17pp.
- W3C**, 2004, World Wide Web Consortium Resource Description Framework (RDF), <http://www.w3.org/RDF/>, (Erişim tarihi: Mayıs, 2012).
- Warwas, S. and Hahn, C.**, 2008, The concrete syntax of the platform independent modeling language for multiagent systems, In proceedings of the Agent-based Technologies and applications for enterprise interoperability, held in conjunction with the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal.
- Weiss, G.**, 1999, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, USA, 619p.
- WSMO Working Group**, 2005, Web Service Modeling Ontology, <http://www.wsmo.org/index.html>, (Erişim tarihi: Haziran, 2012).
- Xpand**, 2009, Eclipse Xpand Model Transformation Framework, <http://www.eclipse.org/modeling/m2t>, (Erişim tarihi: Haziran, 2012).
- Xtext**, <http://www.eclipse.org/Xtext/>, (Erişim tarihi: Haziran, 2012).
- Zambonelli, F., Jennings, N.R. and Wooldrige, M.**, 2003, Developing multiagent systems: The Gaia methodology, ACM Transactions on Software Engineering and Methodologies, 12(3):317-370pp.
- Zdun, U. and Strembeck, M.**, 2009, Reusable Architectural Decisions for DSL Design: Foundational Decisions in DSL Development, Proceedings of the 14th Annual European Conference on Pattern Languages of Programming (EuroPLoP), Aachen, Germany, 1-37pp.

ÖZGEÇMİŞ

Sebla DEMİRKOL

Adres: Uluslararası Bilgisayar Enstitüsü 35100 Bornova/İZMİR

e-mail: sebla.demirkol@ege.edu.tr, seblademirkol@gmail.com

Kişisel Bilgiler

Uyruk: Türkiye Cumhuriyeti

Doğum Yeri: İzmir

Doğum Tarihi: 22.02.1988

Eğitim Durumu

Yüksek Lisans : 2009 – , Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü

Lisans : 2006 – 2009, Ege Üniversitesi Matematik Bölümü (Yatay Geçiş)

Lise : 2002 – 2005, Özel Ege Lisesi

Yabancı Dil

Türkçe : anadil

İngilizce : iyi derecede

Bilgisayar dilleri ve program deneyimi

- C/C++, C#, Java
- Microsoft Visual Studio 2008 Professional, Eclipse IDE, NetBeans IDE, Microsoft Windows Office

Projeler

- JADEX ile elektronik takas sistemi
- JADEX ile uzman bulma sistemi
- JADE ile elektronik takas sistemi
- C ile öğrenci bilgi sistemi takip programı
- JAVA ile konferans yönetim sistemi
- Anlamsal Web Yetenekli Çok-Etmenli Sistemler İçin Alana Özgü Modelleme Dili – Tübitak Proje No: 109E125 (2011 - ,)

Yayınlar

- Demirkol, S., Challenger, M., Getir, S., Kosar T., Kardas, G and Mernik, M. (kabul edildi) “SEA_L: A Domain-specific Language for Semantic Web enabled Multi-agent Systems”, 2nd Workshop on Model Driven Approaches in System Development (MDASD), Wroclaw, Poland, IEEE Computer Society.
- Getir, S., Challenger, M., Demirkol, S. and Kardas, G. (2012) "The Semantics of the Interaction between Agents and Web Services on the Semantic Web", In proceedings of the 7th IEEE International Workshop on Engineering Semantic Agent Systems (ESAS 2012), held in conjunction with the 36th IEEE Signature Conference on Computers, Software, and Applications (COMPSAC 2012), Izmir, Turkey, IEEE Computer Society.
- Getir, S., Demirkol, S., Challenger, M. and Kardas, G. (2011) "The GMF-based Syntax Tool of a DSML for the Semantic Web enabled Multi-Agent Systems", In proceedings of the Workshop on Programming Systems, Languages, and Applications based on Actors, Agents, and Decentralized Control (AGERE! 2011), held at the 2nd Systems, Programming, Languages and Applications: Software for Humanity Conference (SPLASH 2011), Portland, USA, ACM Press, 235-238pp.
- Challenger, M., Getir, S., Demirkol, S. and Kardas, G., A Domain Specific Metamodel for Semantic Web enabled Multi-agent Systems, In Salinesi, C. and Pastor, O. (Eds.): Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, Springer-Verlag, 83:177-186pp.
- Getir, S., Demirkol, S., Challenger, M. and Kardaş, G., 2011b, Anlamsal Web Yetenekli Etmenler için Alana Özgü bir Modelleme Diline ait Görsel Sözdizim, 5. Ulusal Yazılım Mühendisliği Sempozyumu (UYMS 2011), Ankara, Türkiye, 93-100ss.
- Demirkol, S., Getir, S., Challenger, M. and Kardas, G. (2011) "Development of an Agent based E-barter System", In proceedings of 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), Istanbul, Turkey, IEEE Computer Society, 193-198pp.

EKLER

Ek 1 Türkçe - İngilizce Terimler Sözlüğü

Al	Receieve
Alan	Domain
Alan Rolü	Domain Role
Alana Özgü Dil	Domain Specific Language
Alana Özgü Modelleme	Domain Specific Modeling
Alana Özgü Modelleme Dili	Domain Specific Modeling Language
Algılayıcı	Sensor
Alman Yapay Zeka Araştırma Merkezi	German Research Centre for Artificial Intelligence
Altyapı	Grounding
Amaç	Goal
Amaç-Yönelimlilik	Goal-Orientedness
Anlamsal	Semantics
Anlamsal Servis Anlaşma Planı	Semantic Service Agreement Plan
Anlamsal Servis Arabulucu Etmeni	Semantic Service Matchmaker Agent
Anlamsal Servis Bulucu Plan	Semantic Service Finder Plan
Anlamsal Servis Çalıştırıcı Plan	Semantic Service Executor Plan
Anlamsal Servis Kayıt Planı	Semantic Service Register Plan
Anlamsal Web Etmeni	Semantic Web Agent
Anlamsal Web Organizasyonu	Semantic Web Organization
Anlamsal Web Servis	Semantic Web Service
Aracı	Matchmaker
Arayüz	Interface
Atlas Dönüşüm Dili	Atlas Transformation Language
Basit Nesne Erişim Protokolü	Simple Object Access Protocol
Biçimsel	Formal
Bilgi Sorgulama ve İşleme Dili	Knowledge Query and Manipulation Language
Bilgilendirme	Inform
Bire bir	One to One
Bire çok	One to Many
Birleşik Modelleme Dili	Unified Modeling Language

Ek 1 (devam)

Birlikte İşlerlik	Interoperability
Boş Değer	Null
Bütünleştirme	Aggregation
Çevre	Environment
Çıkarsama	Inference
Çıktı	Output
Çok-Etmenli Sistemler	Multi-Agent Systems
Çoklu Etmenler	Multi Agents
Çoktan Çoğa	Many to Many
Davranışsal	Behavioural
Denklik	Equality
Dünya Çapında Ağ Birliği	World Wide Web Consortium
Dünya Geneli Ağ	World Wide Web
Düşünülmüş Tavrılar	Deliberative Attitudes
Eclipse Grafiksel Modelleme Ortamı	Eclipse Graphical Modeling Framework
Eclipse Modelleme Ortamı	Eclipse Modeling Framework
Ecore Üstmodeli	Ecore Metamodel
Eksik Tip	Missing Type
Elektronik Takas Sistemi	Electronic Barter System
En İyi Eşleme	Best Matching
Etki	Effect
Etkileşim	Interaction
Etkileyici	Effector
Etmen Durumu	Agent State
Etmen İletişim Dili	Agent Communication Language
Etmen Modelleme Dili	Agent Modeling Language
Etmen Sınıf Üstyapısı	Agent Class Superstructure
Üstmodeli	Metamodel
Etmen Tabanlı Yazılım Geliştirme	Agent Oriented Software Development
Etmen Tabanlı Yazılım Mühendisliği	Agent Oriented Software Engineering
Etmen Tanımlama Dosyası	Agent Definition File

Ek 1 (devam)

Etmen Tipi	Agent Type
Etmen UML	Agent UML
Eylem	Action
Finansal Plan	Financial Plan
Formal Anlam	Formal Semantics
Geçersiz İsim	Invalid Name
Genişletme	Extension
Gerçek	Fact
Gerçekleştirici	Performative
Girdi	Input
Gönder	Send
Görev	Task
Hedef	Intention
Hızlı Düzeltme	Quick Fix
Hücreleme	Compartment
İstek	Desire
İşbirliği	Cooperates
Kalıcı Süreklilik	Permanent Continuity
Kanı	Belief
Kanı Tabanı	Beliefbase
Kanı-İstek-Hedef	Belief-Desire-Intention
Karşıt Eylemli Etmen Mimarisi	Reactive Architecture
Karşıt-Eylemlilik	Reactivity
Kayıt Rolü	Registration Role
Kaynak	Resource
Kaynak Tanımlama Çerçevesi	Resource Description Framework
Kod Formatlama	Code Formatting
Kontrat Net Protokolü	Contract Net Protocol
Kural	Rule
Mesaj	Message
Mesaj Dizisi	Message Sequence
Mimari Rol	Architecture Role
Model Güdümlü Geliştirme	Model Driven Development

Ek 1 (devam)

Müşteri	Customer
Müzakere	Deliberation
Nesne Yönetim Grubu	Object Management Group
Nicelik	Cardinality
Onay	Acknowledgement
Ontoloji Arabulucu Rolü	Ontology Mediator Role
Ontoloji Tanımlama Üstmodeli	Ontology Definition Metamodel
Ontoloji Web Dili	Ontology Web Language
Organizasyon	Organization
Organizasyon Ontoloji	Organization Ontology
Otomatik Akıl Yürütme	Automated Reasoning
Otomatik Düzeltme	Auto Edit
Otomatik Tamamlama	Auto Completion
OWL Servisleri	OWL for Services
Öncelik	Priority
Önek	Prefix
Önkoşul	Precondition
Özellikler	Properties
Özerk	Autonomous
Özerklik	Autonomy
Özne-Yüklem-Nesne	Subject-Predicate-Object
Öznitelik	Attribute
Parantez Eşleme	Bracket Matching
Platform Bağımsız Üstmodel	Platform Independent Metamodel
Prosedürel Çıkarsama Sistemi	Procedural Reasoning System
Rol	Role
Rol Ontoloji	Role Ontology
Servis Ontoloji	Service Ontology
Servis Profili	Service Profile
Servis Zemini	Service Grounding
Sınıflar	Classes
Somut Sözdizim	Concrete Syntax
Sosyal Yetenek	Social Ability

Ek 1 (devam)

Soyut Sözdizim	Abstract Syntax
Sözdizim	Syntax
Sözdizimi Renklendirme	Syntax Coloring
Süreç	Process
Süreç Modeli	Process Model
Takas Rolü	Barter Role
Takas Servisi	Barter Service
Takas Sistemini Bul	Discover Barter Service
Takas Sistemini Çağır	Invoke Barter Service
Takas Yöneticisi	Barter Manager
Talep	Request
Tanım	Description
Taslak	Template
Taslak Gösterim	Outline View
Tekbiçimli Kaynak Tanımlayıcısı	Uniform Resource Identifier
Tepki	Reaction
Tip	Type
Tümleşim	Composition
Uyarı	Warning
Üstmodel	Metamodel
Üstyapı	Superstructure
Web Servis Modelleme Ontolojisi	Web Service Modeling Ontology
Web Servis Tanımlama Dili	Web Service Description Language
Yazılım Etmenleri	Software Agents
Yeniden Adlandırma	Rename Refactoring
Yerleşik	Built-in
Yetenek	Capability
Yetenek Tanımlama Dosyası	Capability Definition File