

**Eclipse Platformu İin
Yüksek Düzeyli Mimari
Yazılım Modelleme Aracı**

**The High Level Architecture
Software Modeling Tool
for Eclipse Platform**

OĐUZ BAKTIR

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

BİLGİSAYAR MÜHENDİSLİĐİ Anabilim Dalı İin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2006

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Üye (Danışman) :.....
(Yrd. Doç. Dr. Kayhan İMRE)

Üye :.....
(Yrd.Doç.Dr. Harun ARTUNER)

Üye :.....
(Yrd.Doç. Dr. Ali Ziya ALKAR)

ONAY

Bu tez/...../..... tarihinde Enstitü Yönetim Kurulunca belirlenen yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

...../...../.....

Prof. Dr. Ahmet R. ÖZDURAL
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

ECLIPSE PLATFORMU İÇİN YÜKSEK DÜZEYLİ MİMARİ YAZILIM MODELLEME ARACI

Oğuz Baktır

ÖZ

Yüksek Düzeyli Mimari (HLA - *High Level Architecture*), benzetim sistemlerinin birlikte çalışabilirliği ve tekrar kullanılabilirliğini sağlamak amacıyla, Amerikan Savunma Bakanlığı'na bağlı Modelleme ve Benzetim Ofisi tarafından tanımlanmış bir belirtimdir. Bu belirtime göre, HLA uyumlu benzetim sistemleri geliştirilirken izlenmesi gereken adımlar da Federasyon Çalışma ve Geliştirme Süreci (FEDEP) ile tanımlanmıştır. FEDEP adımlarını gerçekleştirirken yardımcı araçların kullanımı etkinliği artırır. Bu amaca yönelik çeşitli araçlar geliştirilmiştir, fakat bu araçların eksiklikleri bulunmaktadır. Mevcut modelleme araçlarının tespit edilen üç önemli eksiği, yeterli görsel modelleme desteklerinin olmaması, davranış modellemesine olanak sağlamamaları ve bütünlük bir platformun bir parçası olmamalarıdır. Bütünlük bir platformun parçası olarak gerçekleştirilmiş modelleme araçlarının bir kısmı ise çalıştıkları platformun lisansı ile kullanılabildiklerinden pahalıya mal olmaktadır.

Mevcut araçların eksikliklerini taşımayan, açık kaynak kodlu ve tamamıyla ücretsiz olarak dağıtılan bir platform olan Eclipse platformu üzerinde özgün bir HLA araç setinin tasarımı yapılmıştır. Araç setinin modelleme, otomatik kod üretme, izleme ve sınamaya araçlarını içermesi planlanmış fakat bu tez kapsamında sadece IEEE 1516.2 OMT belirtimine uyan bir HLA FOM modelleme aracı gerçekleştirilmiştir.

Anahtar Kelimeler: Eclipse, Eclipse Modelleme Çerçevesi (EMF), Eclipse Grafiksel Betimleme Çerçevesi (GEF), Eclipse Eklenti Geliştirme Ortamı (PDE), Java Yayınlayıcı Şablonları (JET), Davranışsal Modelleme, Federasyon Geliştirme ve Çalıştırma Süreci (FEDEP), Yüksek Düzeyli Mimari (HLA), Federasyon Nesne Modeli (FOM), Modele Dayalı Mimari (MDA), Birleştirilmiş Modelleme Dili (UML)

Danışman: Yrd. Doç. Dr. Kayhan İMRE,

Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü

THE HIGH LEVEL ARCHITECTURE SOFTWARE MODELING TOOL FOR ECLIPSE PLATFORM

Oğuz Baktır

ABSTRACT

High Level Architecture (HLA) is developed by Defense Modeling and Simulation Office (DMSO) of U.S. Department of Defense (DoD) to improve interoperability and reusability of simulation systems. Federation Development and Execution Process (FEDEP) defines steps of HLA compliant simulation system development. It is easier to implement FEDEP steps with the help of automation tools. Such tools have been implemented and are available in the market for some time. However, they have either some shortcomings like lacking of efficient visual modeling, not having behavioral modeling capability, not being part of an IDE; or they are too expensive to own and use.

An HLA toolset that does not inherit shortcomings of existing HLA tools, and depends on an open-source and freely-available platform has been designed. Such a toolset should consist of modeling, code generation, monitoring and testing tools. In this thesis, a visual modeling tool has been implemented, which makes it possible to model IEEE 1516.2 OMT compliant HLA federation object models easier, quicker and dependable.

Keywords: Eclipse, Eclipse Modeling Framework, Eclipse Graphical Editing Framework, Eclipse Plug-in Development Environment (PDE), Java Emitter Templates (JET), Behavioral Modeling, Federation Execution Development Process (FEDEP), High Level Architecture (HLA), Federation Object Model (FOM), Model Driven Architecture (MDA), Unified Modeling Language (UML)

Advisor: Assistant Professor Kayhan İMRE, Hacettepe University, Department of Computer Science and Engineering

TEŞEKKÜR

Yazar, bu çalışmanın gerçekleşmesinde katkılarından dolayı, aşağıda adı geçen kişi ve kuruluşlara içtenlikle teşekkür eder.

Yrd. Doç. Dr. Kayhan İmre'ye, tez konusunun belirlenmesini sağladığı, tez çalışmasının gerçekleştirilmesindeki desteği ve anlayışı için,

Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü Başkanı Prof.Dr.Ersin Töreci'ye, savunma sanayiinde çalışırken tamamlamaya çalıştığım yüksek lisans çalışmalarım sırasında zaman zaman beni uyardığı ve yol göstererek yardımcı olduğu için,

Sevgili eşim Sadiye Baktır'a, çalışmalarım esnasında, huzur dolu bir ortam sağladığı, ayrıca sıkışık zamanlarımda işi zamanında ve uygun şekilde bitirebilmem için değerli tecrübelerini aktararak ve önerilerde bulunarak destek olduğu için,

Dedem Mht. Mehmet Baktır'a, sağlığıma dikkat etmemi her fırsatta hatırlattığı ve “ ‘Zor’ deme. Çalış, kolaylaştır!” diyerek öğüt verdiği, giriştiğim işlerde sonuca ulaşma azmimi artırdığı için,

Son olarak, annem Ecz.Ayşe Baktır'a ve babam Dr.Mehmet Baktır'a, tüm hayatım boyunca, varlıklarıyla ve ihtiyaç duyduğum zamanlarda her türlü destekleriyle bana güven aşıladıkları için; ayrıca kardeşlerim Selim,Elif,Zeynep ve Selçuk'a, hayatıma neşe kattıkları için.

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZ.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER DİZİNİ	iv
SİMGELER VE KISALTMALAR	vii
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	x
EKLER DİZİNİ	xi
1 GİRİŞ	1
2 TEMEL KAVRAMLAR	3
2.1 YÜKSEK DÜZEYLİ MİMARİ	3
2.1.1 HLA.....	3
2.1.2 HLA İhtiyacının Oluşumu	3
2.1.3 RTI.....	5
2.1.4 HLA Kuralları.....	5
2.1.5 Arayüz Belirtimi	6
2.1.6 HLA Nesne Modeli Şablonu	7
2.2 FEDEP	7
2.2.1 Federasyon Amaçlarını Belirleme	9
2.2.2 Federasyon Kavramsal Modelini Geliştirme.....	11
2.2.3 Federasyon Tasarımı	12
2.2.4 Federasyon Geliştirimi	13
2.2.5 Federasyon Bütünleştirme ve Sınama	14
2.2.6 Federasyonun Çalıştırılması ve Çıktıların Üretimi.....	15
2.2.7 Çıktı Analizi ve Sonuçları İnceleme.....	15
2.3 MODELE DAYALI MİMARİ	15
2.4 ECLIPSE PLATFORMU.....	18
2.4.1 Eclipse	18
2.4.2 Eclipse PDE	18
2.4.3 Eclipse Modelleme Çerçevesi.....	19
2.4.4 Grafiksel Düzenleme Çerçevesi.....	22

3	MEVCUT HLA MODELLEME ARAÇLARI.....	23
3.1	OMDT	23
3.2	VisualOMT	24
3.3	SIMplicity.....	25
3.4	RRTEI destekli HLA Modelleme Aracı	26
3.5	Modelleme Araçlarının Karşılaştırması	27
4	HLA İÇİN YENİ BİR MODELLEME ARACI GERÇEKLEŞTİRİMİ.....	28
4.1	Araç Setinin Sağladığı Avantajlar.....	29
4.2	Araç Setini Geliştirme Adımları	29
4.3	HLA Nesneleri İçin Geliştirilen Meta Model	31
4.3.1	HLA Meta Modeli Elemanları - HLA Nesne Sınıfı.....	34
4.3.2	HLA Meta Modeli Elemanları - HLA Nesne Sınıfı Niteliği.....	35
4.3.3	HLA Meta Modeli Elemanları - HLA Etkileşim Sınıfı.....	36
4.3.4	HLA Meta Modeli Elemanları - HLA Etkileşim Sınıfı Parametresi ...	37
4.3.5	HLA Boyutu.....	38
4.3.6	HLA Temel Veri Türleri	38
4.3.7	HLA Biçim İmi	39
4.3.8	HLA Zamanı.....	39
4.3.9	HLA Notu	39
4.3.10	HLA Federe.....	39
4.3.11	HLA Zaman Uyumlama Noktası.....	40
4.4	Araç Setinin Alt Bileşenleri	40
4.4.1	HLAM76B Model Eclipse Eklentisi	40
4.4.2	HLAM76B Edit Eclipse Eklentisi.....	41
4.4.3	HLAM76B Editör Eclipse Eklentisi	41
4.4.4	HLAM76B Grafiksel Editör Eclipse Eklentisi	42
4.4.5	HLAM76B MBGA Alt Bileşenlerinin Etkileşimi.....	43
4.5	Araç Setinin Kurulumu	44
4.6	Araç Setinin Kullanımı.....	45
4.6.1	HLA Nesne Modelinin Oluşturulması	45
4.6.2	Nesne Sınıflarının Tanımlanması.....	45
4.6.3	Nesne Sınıfı Niteliklerinin Tanımlanması	46
4.6.4	Veri Türü (<i>Datatype</i>)	47
4.6.5	Boyut (<i>Dimension</i>):	48

4.6.6	Cardinality.....	49
4.6.7	Birim (<i>Unit</i>).....	49
4.6.8	Duyarlılık (<i>Resolution</i>).....	49
4.6.9	Doğruluk (<i>Accuracy</i>).....	49
4.6.10	Doğruluk Koşulu (<i>Accuracy condition</i>).....	50
4.6.11	Günleme Türü.....	50
4.6.12	Günlenebilir (<i>Updateable</i>) / Yansıtılabilir (<i>Reflectable</i>) Olma.....	50
4.6.13	Devredilebilir (<i>Transferable</i>) / Kabul edilebilir(<i>Acceptable</i>) Olma.....	50
4.6.14	Teslim Türü (<i>Delivery type</i>).....	50
4.6.15	İleti Sıralama Türü (<i>Message ordering type</i>).....	51
4.6.16	Etkileşim Sınıflarının Tanımlanması.....	51
4.6.17	Karmaşık Veri Türü ve Alanlarının Tanımlanması.....	53
4.6.18	Numaralandırılmış Veri Türü ve Numaralandırıcılarının Tanımlanması.....	54
4.6.19	Model ve Geliştirici Kimlik Bilgilerinin Tanımlanması.....	55
4.6.20	HLA Modelinden FOM Kütüğünün Oluşturulması.....	56
4.6.21	Federe Modelinin Oluşturulması.....	57
5	SONUÇLAR VE ÖNERİLER.....	58
	KAYNAKLAR DİZİNİ.....	60

SİMGELELER VE KISALTMALAR

DDM	Data Distribution Management
DoD	Department of Defense
DMSO	Defense Modeling and Simulation Office
EMF	Eclipse Modelling Framework
EPL	Eclipse Public License
FED	Federation Execution Data
FEDEP	Federation Development and Execution Process
FOM	Federation Object Model
GEF	Graphical Editing Framework
HLA	High Level Architecture
MDA	Model Driven Architecture
MOF	Meta Object Facility
OMDT	Object Model Definiton Tool
OMG	Object Management Group
OMT	Object Model Template
PDE	Plug-in Development Environment
RID	RTI Initialization Data
SOM	Simulation Object Model
UML	Unified Modeling Language

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2-1 FEDEP Adımları Genel Görünümü.....	9
Şekil 2-2 OMG Model Driven Architecture.....	16
Şekil 2-3 OMG 4-Katmanlı Modelleme Yığını.....	17
Şekil 2-4 Eclipse Platformu Mimarisi (Griffin, [9]'den değiştirilerek).....	19
Şekil 2-5 EMF Ecore Modeli Nesneleri (Artifacts).....	21
Şekil 3-1 OMDT Ekran Görüntüsü.....	24
Şekil 3-2 Visual OMT Aracından Ekran Görüntüsü	25
Şekil 3-3 Simplicity Aracından Ekran Görüntüsü	26
Şekil 3-4 RRTEI destekli HLA Modelleme Aracı Ekran Görüntüsü.....	27
Şekil 4-1 Aracın HLA Çalışma Ortamındaki Yeri	29
Şekil 4-2 Modelleme Aracının Ekran Görüntüsü.....	30
Şekil 4-3 IEEE 1516.2 OMT HLA Elemanları Meta Modeli.....	33
Şekil 4-4 Meta model - HLA Nesne Sınıfı	35
Şekil 4-5 Meta model - HLA Etkileşim Sınıfı	37
Şekil 4-6 Meta model - HLA Veri Türleri	39
Şekil 4-7 HLAM76B HLA Model Sihirbazı	42
Şekil 4-8 HLAM76B Model - Eklenti Bağımlılıkları.....	43
Şekil 4-9 HLAM76B Edit - Eklenti Bağımlılıkları	43
Şekil 4-10 HLAM76B Editör - Eklenti Bağımlılıkları	44
Şekil 4-11 HLAM76B GEF - Eklenti Bağımlılıkları	44
Şekil 4-12 HLA Nesne Sınıfı Genel Özellikleri.....	46
Şekil 4-13 HLA Nesne Niteliği Genel Özellikleri	47
Şekil 4-14 HLA Veri Türleri Özelliklerini Düzenleme.....	48
Şekil 4-15 HLA Boyut Özelliklerini Düzenleme	48
Şekil 4-16 HLA Boyut Değeri Ayarları	49
Şekil 4-17 HLA Etkileşim Sınıfı Genel Özellikleri.....	52
Şekil 4-18 HLA Karmaşık Veri Türü Alanı, Birim, Duyarlık Bilgileri.....	53
Şekil 4-19 HLA Karmaşık Veri Türü Genel Özellikleri.....	54
Şekil 4-20 HLA Numaralandırılmış Veri Türü Genel Özellikleri.....	55
Şekil 4-21 Model Kimlik Bilgilerine Erişim İçin Menü Yapısı	55
Şekil 4-22 Model Kimlik Bilgileri Arayüzü	56

Şekil 4-23 FOM Dışarı/İçeri Aktarım (Export/Import) Arayüzü 57

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 2-1 FEDEP Adımları Alt İşlemleri.....	9
Çizelge 3-1 HLA için Modelleme Araçlarının Karşılaştırılması.....	28

EKLER DİZİNİ

	<u>Sayfa</u>
EK – 1 Eclipse Eklentileri Tanımlama Kütükleri.....	62
EK – 2 Hlam76b Ecore Meta Modeli	70
EK – 3 Hlam76b Ecore Kütüğü Örneği.....	75
EK – 4 Hlam76b Ecore Modeli GEF Kütüğü Örneği	76
EK – 5 IEEE 1516.2 HLA.DTD	80

1 GİRİŞ

Modelleme ve benzetim sistemleri, genellikle karmaşık mekanizmaların ve sistemlerin beklenmeyen davranışlarını önceden tesbit edebilmek ve beklenen davranışları da önceden sınavabilmek için geliştirilen; araştırma, geliştirme ve üretim safhalarında materyal, insan kaynakları ve zamandan tasarrufun esas gaye olduğu sistemlerdir. Modelleme ve benzetim sistemlerinin üretime sağladıkları fayda aşikardır. Geliştirilmeleri esnasında kendilerine ve geliştiricilerine sağladıkları fayda ise Amerikan Savunma Bakanlığı'na bağlı Modelleme ve Benzetim Ofisi (Defense Modeling and Simulation Office), Yüksek Düzeyli Mimari (HLA - *High Level Architecture*)'yi ortaya koyana kadar gözardı edilmiştir.

Gün geçtikçe zenginleşirken bir yandan da karmaşıklaşan hayatımızda, modelleme ve benzetim sistemlerinin kullanım alanları da giderek artmaktadır. Meteorolojik benzetimlerden, borsa tahminlerine, buzdolabı fabrikasında yapılan termodinamik modellemelerden, araba fabrikalarında yapılan çeşitli benzetim ve modellemelere, belediyelerin caddelerde oluşacak araç akışını tahmin etmek ve düzenlemek için, bakım inşaatlarına başlamadan önce yaptırdığı modelleme ve benzetimlere kadar hayatımızda sayısız noktada, modelleme ve benzetimler desteğiyle sürekli iyileştirmeler yapılmaktadır [10].

Görülüyor ki hayatımızda aldığı rolün büyümesi ve ödediğimiz vergilerden aldığı payın da artması kaçınılmaz olan modelleme ve benzetim sistemlerinin üretim maliyetlerinin düşürülmesi kaçınılmazdır.

Modelleme ve benzetim sistemlerinin üretimini kolaylaştırmak, yeniden kullanımını ve ihtiyaçlara göre yeni modelleme ve benzetimlerin biraraya getirilmesi sağlamak iddiasıyla ortaya çıkan HLA (High Level Architecture) mimarisi, esneklik, yeniden kullanılabilirlik ve maliyetlerle ilgili sıkıntıları şimdilik çözmüş görünmektedir [3;4].

HLA kullanılarak geliştirilen modelleme ve benzetim sistemlerinin hem bir arada çalışabilir hem de ayrı ayrı çalışabilir tasarlanması; ayrıca yeniden kullanılabilirliği kolaylaştırması nedediyle ölçeklenebilirlik sağlanmakta, dolayısıyla benzetim sistemi mimarilerinde esneklik ve ekonomi ortaya çıkmaktadır.

Benzetim bileşenlerinin birlikte çalışabilmeleri ve yeniden kullanılabilmeleri için önceden tanımlanmış birtakım kurallara uymaları gerekir. HLA bu ihtiyaçları karşılamak için DMSO (*Defense Modeling and Simulation Office*) tarafından

geliştirilmiştir. HLA'da benzetim bileşenleri federe, benzetim sistemi de federasyon olarak adlandırılır. HLA temelde, federasyon kuralları, arayüz belirtimi ve nesne modeli şablonundan (OMT – *Object Model Template*) oluşur [5]. Federasyon kuralları, federelerin ve federasyonun uyması gereken kurallardır. HLA'nın temel kurallarından biri federe etkileşimlerinin bir altyapı üzerinden sağlanmasıdır. Bu altyapı işletim zamanı altyapısı (RTI – *Runtime Infrastructure*) olarak adlandırılır. Arayüz belirtimi federelerin RTI üzerinden alacağı hizmetler ve RTI'nın federeler üzerinden kullanabileceği geridönüş (*callback*) çağrılarında oluşur. HLA belirtiminde federelerin işletim zamanında kullanacakları nesnelere ve gönderip alabilecekleri etkileşimler nesne modeli şablonları ile tanımlanır. Federasyondaki her federe için bir benzetim nesne modeli (Simulation Object Model - SOM), federasyon genelindeki nesne tanımları için bir federasyon nesne modeli (Federation Object Model - FOM) bulunur. RTI'nın kullanacağı nesne modeli federasyon çalışma bilgisi (Federation Execution Data - FED) olarak adlandırılır. HLA kavramları hakkında bilgi edinmek için [3], [4], [5], [13] incelenebilir.

HLA uyumlu benzetim sistemleri geliştirilirken izlenecek yol Federasyon Geliştirme ve Çalıştırma Süreci (FEDEP) ile tanımlanmıştır [4]. FEDEP temel olarak federasyonun amaçlarının belirlenmesi, kavramsal modelinin oluşturulması, tasarlanması, geliştirilmesi, alt birimlerinin birleştirilmesi ve sınanması, federasyonun işletilmesi ve sonuçlarının incelenmesi adımlarından oluşur. Federasyon gerçekleştirilirken FEDEP modelinin kullanılması önerilir. Uygulama geliştiriciden, FEDEP modelinde tanımlanan adımları doğru olarak uygulaması beklenir. FEDEP hakkında daha detaylı bilgi Temel Kavramlar bölümünde verilecektir.

Bu tez kapsamında modelleme ve benzetim sistemlerinin ortak nesne ve etkileşim yapılarını modellemek amacıyla Nesne Modeli Tasarım Aracı geliştirilmiştir.

Tez metninin ilerleyen kesimlerinde sırasıyla HLA, FEDEP ve MDA (Model Driven Architecture – Modele Dayalı Mimari) temel kavramları anlatılacaktır. HLA için geliştirilmiş mevcut modelleme araçları incelenmesi ve karşılaştırılmasının ardından, bu tez çalışmasında geliştirilen görsel modelleme aracı hakkında genel bilgi verilecektir. Ardından modelleme aracı detaylı olarak incelenecek ve sonuçlar anlatılacaktır.

2 TEMEL KAVRAMLAR

2.1 YÜKSEK DÜZEYLİ MİMARİ

2.1.1 HLA

HLA (High Level Architecture), benzetim sistemlerinin tasarlanmasına esneklik getirerek ileriki zamanlara ihtiyaç duyulacak olası değişiklikleri gerçekleştirmeyi kolaylaştıran; benzetim sistemlerinin yeniden kullanılabilirliğini ve diğer benzetim sistemleri ile ortak çalışabilirliğini sağlayan, Amerikan Savunma Bakanlığı öncülüğünde geliştirilmiş olan bir belirtimdir. HLA, benzetim sistemi geliştiricilerine uygulamalarını tanımlayıp gerçekleştirebilecekleri bir uygulama çatısı sunar. Aşağıda gösterildiği gibi HLA belirtimi üç kesimden oluşur.

1. HLA kuralları
2. Arayüz belirtimi (*Interface Specification*)
3. Nesne modeli şablonu (*Object Model Template*)

Hali hazırdaki en güncel HLA belirtimi IEEE tarafından geliştirilen 1516 kod numaralı belirtimdir. Başlangıçta, HLA askeri amaçlarla tanımlansa da günümüzde trafik benzetimleri, imalat sanayii [11], bilgisayar oyunları gibi sivil alanlarda da yaygın olarak kullanılmaktadır [12].

HLA belirtiminde benzetimin alt bileşenleri “federe” olarak adlandırılır. Federeler geliştirilirken HLA belirtiminin kuralları uygulanmalıdır. HLA benzetim sistemleri birlikte çalışan bir grup HLA federesinden oluşur ve buna “federasyon” adı verilir. Bir federasyonda, farklı federe türlerinden olgular bulunabileceği gibi, aynı federe türünden birden çok olgu da bulunabilir. Örneğin bir uçuş benzetiminde aynı anda birden fazla uçak federesi olabilir. Bunun yanında uçaksavar federeleri de bulunabilir. Federeler federasyonun çalışması esnasında herhangi bir anda federasyona katılabilir veya federasyonu terkedebilirler. Federasyonlarda benzetimi oluşturan temel federelerin yanısıra, izleme, veri toplama ve yönetim gibi işlerden sorumlu yardımcı federeler, donanım ve grafik arayüzler gibi bileşenler olabilir.

2.1.2 HLA İhtiyacının Oluşumu

Benzetim sistemleri kendi içlerinde alt benzetim sistemlerinden oluşabilirler. Bu alt benzetim sistemleri ise başka benzetim sistemlerinde de kullanılabilir denli belirli, test edilmiş ve olgunluğa erişmiş olabilir. Bu belirli ihtiyaçları karşılayan daha önce geliştirilmiş alt benzetimler fazla değişiklik yapılmadan tekrar kullanılabilirse benzetim sisteminin geliştirme süresi düşer. Ayrıca daha önceden sınanmış benzetimler olduklarından sınama maliyeti önemli ölçüde ortadan kalkar. Örneğin, yeni geliştirilen bir tank ve bu tankın silah sistemleri bir önceki tank modeliyle birçok ortak alt birim içerebilir. Bunun sebebi de yeni teknolojilerin ve ürünlerin genellikle eski teknolojilerden ilham alarak ortaya ve eskinin üzerine birikimsel olarak eklenen yeni özelliklerle geliştirilmesidir. Dolayısıyla eski ürünler için geliştirilmiş benzetim sistemleri ile yeni geliştirilecek ürünlerin ihtiyaç duyacağı benzetim sistemleri de ortak birçok alt benzetim kullanıyor olacaklardır. Örneğimizde eski model bir tankın, gereksinimlerinde değişiklik olmayan alt birimleri yeni bir tank geliştirilirken tekrar kullanılırsa, yeni tankın geliştirme maliyeti önemli ölçüde düşecektir. Benzer şekilde yeni tank için bir benzetim sistemi gerçekleştirilirken de eski benzetimler kullanılabilirse büyük kolaylık ve maliyetlerde düşüş ortaya çıkacaktır.

Farklı geliştirme grupları tarafından, farklı teknolojiler kullanılarak geliştirilmiş ve değişik platformlarda çalışan benzetim sistemlerinin birlikte çalışması benzetim sistemlerinin yetenekleri önemli ölçüde artırır. Örneğin C++ dili kullanılarak geliştirilen Unix platformu üzerinde çalışan bir uçak benzetim sistemi ile Java dili kullanılarak gerçekleştirilen, Windows platformunda çalışan bir uçaksavar benzetim sisteminin bir senaryo dahilinde birlikte çalışabilmesi gerekebilir.

Ürünler ve sistemler için benzetim sistemleri geliştirilirken, bu benzetim sistemlerinin yeniden kullanılabilirliğini ve birlikte çalışabilirliği sağlamak çok önemlidir. Özellikle birlikte çalışabilirlik için; geliştirilen benzetim sistemlerinde, diğer benzetim sistemleri ile ortak bir belitime uyulması, geliştirilen ürünlerin ileriki bir zamanda başka ürünlerle entegrasyonunu kolaylaştıracak ve entegrasyon maliyetlerini azaltacaktır.

Kolay entegre edilebilme, birlikte çalışabilme, kolay değişim yapabilme ve maliyetlerden tasarruf sağlama ihtiyaçlarından dolayı Amerikan Savunma

Bakanlığı'na bağı Modelleme ve Benzetim Ofisi bir takım kurallardan, arayüz belirtilerinden ve şablondan oluşan HLA belirtimini tanımlamıştır.

2.1.3 RTI

RTI (Runtime Infrastructure – Çalışma Zamanı Altyapısı), HLA uyumlu benzetim sistemleri için gerekli servis arayüzlerini, arayüz belirtimine uygun olarak, gerçekleştiren bir altyapısal yazılımdır. Federeler arası etkileşim için RTI kullanılır. Federeler birbirleriyle doğrudan etkileşimde bulunmazlar, fakat RTI üzerinden etkileşirler. RTI benzetim sistemlerinin dağıtılabiliğini ve ölçeklenebilirliğini artıran bir platformdur.

2.1.4 HLA Kuralları

HLA kuralları, federasyonu oluşturan federeler arası etkileşimin düzgün bir biçimde gerçekleşmesini garantilemek için federasyonun ve federelerin görevlerini tanımlar. HLA kuralları, federe kuralları ve federasyon kuralları olmak üzere toplam on kuraldan oluşur. Federelerin ve federasyonların HLA uyumlu sayılması için bu kurallara uymaları gerekir

2.1.4.1 Federe kuralları

1. Federenin HLA nesne modeli şablonuna (OMT) göre düzenlenmiş bir benzetim nesne modeli (SOM) olmalıdır.
2. Federeler SOM'da tanımlanan kurallara uyarak olgu niteliklerinin değerlerini günleyip / yansıtabilirler, etkileşim gönderip / alabilir.
3. Federeler SOM'da tanımlanan kurallara uyarak federasyon çalışması esnasında olgu niteliklerinin sahipliğini kabul edebilir / devredebilir.
4. Federeler SOM'da tanımlanan kurallara uyarak sahip oldukları olgu niteliklerinin değerlerini günleyerek federasyonun akışını değiştirebilir.
5. Federeler yerel zamanlarını federasyon üyesi diğer federelerle veri değişimini yönetebilecek şekilde kontrol edebilir.

2.1.4.2 Federasyon kuralları

1. Federasyonun HLA nesne modeli şablonuna (OMT) uygun bir federasyon nesne modeli (FOM) olması gerekir.

2. FOM'daki nesnelerin tanımları RTI tarafından değil, federeler tarafından yapılmalıdır.
3. Federasyon çalışması esnasında tüm FOM bilgi değişimi RTI üzerinden olmalıdır.
4. Federasyon çalışması esnasında federelerle RTI arası iletişim HLA arayüz belirtimine uygun olarak yapılmalıdır.
5. Federasyon çalışması esnasında bir olgu niteliğinin sahipliği aynı anda sadece bir federede olmalıdır. Sahiplik, çalışma zamanında federeler arasında el değiştirebilir..

2.1.5 Arayüz Belirtimi

IEEE 1516.1 olarak standartlaşmış olan arayüz belirtimi ile federeler ve RTI arasındaki işlevsel arayüzler tanımlanmıştır. Arayüz belirtimi, federasyondaki federelerin birbirleriyle etkileşirken RTI üzerinden hangi servisleri kullanacaklarını tanımlar.

HLA arayüz belirtiminin altı temel amacı vardır:

1. Benzetim sistemi ile iletişim altyapısı birbirinden ayırma
2. RTI, DIS, ALSP gibi eski standartların eksik yönlerini giderme
3. Federasyonların oluşturulup yok edilmesini sağlama
4. Nesne / etkileşim tanımlayıp yönetmeye olanağı sağlama
5. Federasyon zaman yönetimi ile uyumlu çalışma
6. Federeleri mantıksal gruplara ayırıp iletişim etkinliklerini artırma.

Arayüz belirtimi altı yönetim alanından oluşur:

1. Federasyon Yönetimi (Federation Management)
2. Tanımlama Yönetimi (Declaration Management)
3. Nesne Yönetimi (Object Management)

4. Sahiplik Yönetimi (Ownership Management)
5. Veri Dağıtım Yönetimi (Data Distribution Management)
6. Zaman Yönetimi (Time Management)

2.1.6 HLA Nesne Modeli Şablonu

HLA federelerini yeniden kullanılabilir ve bir araya getirildiklerinde birlikte çalışabilir kılmak için birbirleriyle etkileşimlerinde kullanılan tüm nesne ve etkileşim sınıflarını, bilinen ortak bir biçime uygun olarak, detaylı bir biçimde tanımlamak gerekir. HLA Nesne Modeli Şablonu (Object Model Template – OMT), HLA nesne modelinin belgelenmesi için kullanılan bir standarttır. Nesne modeli şablonuna uygun olarak değişik amaçlarla üç nesne modeli tanımlanmıştır. Bunlar federasyon nesne modeli, benzetim nesne modeli ve yönetim nesne modelidirler.

2.1.6.1 Federasyon Nesne Modeli

Federasyon Nesne Modeli (Federation Object Model – FOM) federasyon genelinde geçerli olan nesne modelidir. Federeler arasında paylaşılan nesnelere, etkileşimler, bunların veri türleri gibi ortak bilgilerin tamamını içerir.

2.1.6.2 Benzetim Nesne Modeli

Benzetim Nesne Modeli (Simulation Object Model – SOM), federelerin kendine has tanımlarını ve dışarıdan kullanabilecekleri nesne ve etkileşimlerin tanımlarını içerir. Her federenin bir benzetim nesne modeli vardır.

2.1.6.3 Yönetim Nesne Modeli

Yönetim Nesne Modeli (Management Object Model – MOM) federasyonun izlenmesi ve yönetilmesinde kullanılabilecek nesne ve etkileşim sınıflarından içerir. Federasyona özel değil; genel bir tanımdır.

2.2 FEDEP

FEDEP (Federation Execution and Development Plan – Federasyon Çalıştırma ve Geliştirme Planı), HLA uyumlu benzetim sistemleri gerçekleştirilirken kullanılması önerilen bir metodolojidir. HLA uyumlu benzetim sistemleri geliştirilirken izlenmesi

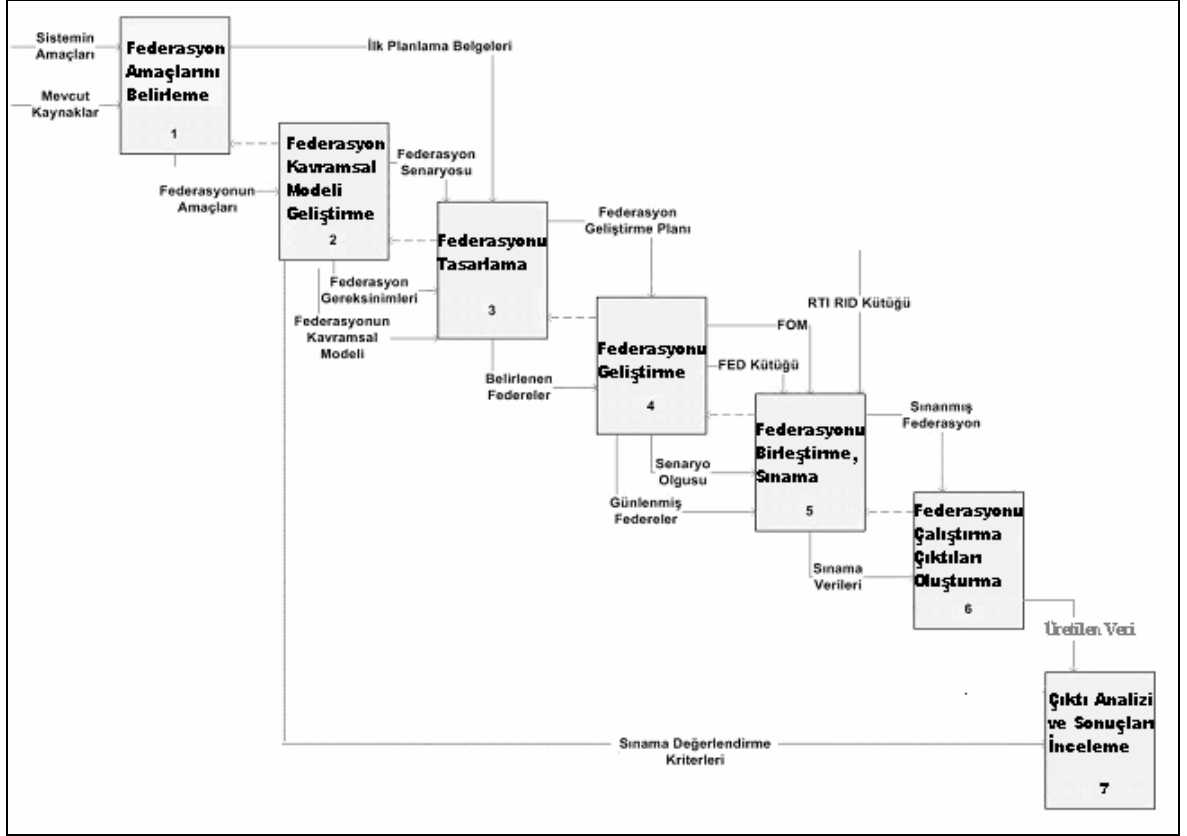
gereken temel adımları belirler. FEDEP federelerin HLA uyumlu olduğunu varsayarak, benzetim sistemini federasyon düzeyinde ele alır.

FEDEP'in yedi alt adımı vardır:

1. Federasyonun amaçlarını belirleme
2. Federasyonun kavramsal modelinin oluşturma
3. Federasyonu tasarlama
4. Federasyonu geliştirme
5. Federasyonun alt birimlerini birleştirme ve federasyonu sınaama
6. Federasyonun çalıştırma ve çıktıları oluşturma
7. Çıktıları analiz etme ve sonuçları değerlendirme

Bu yedi alt adımın gerçekleştirilmesi uygulamanın doğasına göre değişir.

Şekil 2-1'de, FEDEP'in yedi temel alt adımı ve bu adımlar arası geçişler gösterilmiştir. Şekilden süreçler arasında çok katı geçişler olmadığı ve geri dönüşlerin mümkün olduğu anlaşılabilir.



Şekil 2-1 FEDEP Adımları Genel Görünümü

FEDEP adımlarında gerçekleştirilen alt işlemler Çizelge 2-1’de gösterilmiştir.

Çizelge 2-1 FEDEP Adımları Alt İşlemleri

Federasyon Amaçlarını Belirleme	Federasyon Kavramsal Modeli Geliştirme	Federasyonu Tasarlama	Federasyonu Geliştirme	Federasyonu Birleştirme ve Sınama	Federasyonu Çalıştırma ve Çıktıları Oluşturma	Çıktı Analizi ve Sonuçları İnceleme
Amaçları Belirleme Amaca Yönelik İhtiyaçları Ortaya Koyma	Senaryo Geliştirme Federasyon Kavramsal Modelini Geliştirme Federasyon Gereksinimlerini Ortaya Koyma	Federeleri Belirleme Federasyonu Tasarlama Planı Hazırlama	FOM oluşturma Federasyonda Mutabık Kalma Belirlenen Federeleri Geliştirme Federasyon Altyapısını Gerçekleştirme	Federasyon Çalışmasını Planlama Federasyonu Bütünleştirme Federasyonu Sınama	Federasyonun Çalıştırılması Çıktının İncelenmesi Sonuçların Hazırlanması	Çıktı verilerini analiz etme Sonuçları değerlendirme ve geribildirim

2.2.1 Federasyon Amaçlarını Belirleme

Bu kesim iki alt adımdan oluşur. İlk adım, HLA federasyonunun geliştirilmesi ve işletilmesi aşamasındaki ihtiyaçların belirlenip belgelenmesidir. Bu işlem aşağıdaki dört adımı uygulayarak gerçekleştirilmelidir:

1. Problemin tamamen çözümlenip anlaşılması
2. Kritik sistemlerin üst düzey tanımının yapılması, benzetim parçalarının davranışlarının, çıktı verilerinin, gereksinimlerinin ve anahtar olayların belirlenmesi
3. Federasyon geliştirilmesine ayrılacak personel, bütçe, araç gibi kaynakların belirlenmesi
4. Federasyon geliştirme sürecini etkileyecek teslim tarihi, güvenlik gereksinimleri gibi kısıtlarının belirlenmesi.

Bu adım sonlandığı zaman, geliştiricilerin sistemin tüm gereksinimlerini anlamış olmaları ve gereksinimler üzerinde uzlaşıp yanlış anlaşılımların tamamen ortadan kaldırılmış olduğundan emin olunmalıdır.

İkinci adım, bir önceki adımda belirlenen ihtiyaçların daha detaylı federasyon amaçlarına dönüştürülmesidir. Bu işlem aşağıdaki beş adım uygulanarak gerçekleştirilmelidir:

1. İhtiyaçların federasyonun özel amaçlarını oluşturmak üzere iyileştirilmesi
2. Sonuçta ortaya çıkan amaçların ihtiyaçları karşıladığının kontrol edilmesi
3. Federasyonun olurluğunun ve olası risklerin incelenmesi
4. Araçların seçimi
5. Elemanların belgelenmesi, beş alt adımdan oluşur:
 - a) Federasyonun amaçlarının önceliklendirilmiş bir listesinin çıkarılması
 - b) Federasyonun temel özelliklerinin (tekrarlanabilirlik, taşınabilirlik, zaman yönetimi yaklaşımı vs.) üst düzey tanımının yapılması

- c) Federasyon geliştirme takviminin çıkarılması ve aşamaların (*milestone*) belirlenmesi
- d) Federasyonların ve coğrafi alanların sınırlarının ve kısıtların belirlenmesi
- e) Benzetim parametrelerinin sınırlarının ve gizliliklerinin belirlenmesi
- f) Federasyon parametreleri için gerekli malzemelerin, olanakların, veri sınırlandırmalarının ve kısıtların belirlenmesi
- g) Sonuçların gizliliğinin belirlenmesi
- h) İlk plan belgesinin oluşturulması

2.2.2 Federasyon Kavramsal Modelini Geliştirme

Bu kesim dört alt adımdan oluşur. İlk adım gerçek problemlerin irdelenip federasyon senaryosunun belirlenmesidir. Bu adımda amaçlarda belirlenen kısıtları ve yeniden kullanılabilir senaryolar içeren mevcut senaryo veritabanları kullanılarak federasyon senaryosunun işlevsel belirtimi oluşturulur. Oluşturulan bu belirtimin içeriği aşağıdaki gibidir:

- Anahtar olguların tanımları
- Temel varlıkların davranışları ve aralarındaki ilişkiler
- İlişkin çevre koşulları
- İlk durum, sonlanma koşulları
- Özel coğrafi alanlar

İkinci adımda bu kesimin esas amacı olan, federasyonun kavramsal modeli oluşturulur. Kavramsal modelin temel özellikleri aşağıdaki gibidir:

- Amaçların işlevsel ve davranışsal yeteneklere dönüştürülmesinde araç olarak kullanılacak gerçekleştirimden bağımsız bir gösterimdir.

- Federasyonun amaçlarıyla tasarımın gerekleřtirmi arasında tutarlılıđı sađlamak iin bađlantı sađlar.
- Tm federasyon amalarının sađlanması iin gerekli olan varlık ve iřlem tanımlarını ierir.

Kavramsal model tanımlandıktan sonra, nc adımda bařta belirlenen federasyon gereksinimleri dikkatlice incelenerek gerekirse deđiřiklikler yapılır.

Drdnc adımda federasyon amaları, federasyon sınaması esnasında bařarı kriteri olarak kullanılmak zere zel federasyon gereksinimlerine evirilir. İlk etapta federasyonun detaylı gereksinimleri federasyonun kavramsal modeli kullanılarak belirlenmelidir. Bu gereksinimler ařađıdaki zellikleri tařımalıdır:

- Test edilebilir olmalıdır.
- Federasyonun tasarlanması ve gerekleřtirilmesi srelerine rehberlik edebilmelidir.
- Dođruluk (*fidelity*) gereksinimlerini iermelidir. Federasyonun katılımcıları bu dođruluk kriterleri kullanılarak seilmelidir.
- Kısıtların detaylı tanımlarını iermelidir.

2.2.3 Federasyon Tasarımı

Bu kesim  temel adımdan oluřur. İlk adımda federasyonu oluřturacak federeler belirlenir. İlk nce aday federeler, ařađıdaki kısıtlara gre de federeler belirlenir:

- Uygunluk, gvenlik, olanaklar gibi ynetimsel kısıtlar
- VV&A (*verification, validation and accreditation*) durumu, tařınabilirlik gibi teknik kısıtlar. VV&A, DMSO'nun programının kısaltmasıdır.

İkinci adımda federelerin sorumlulukları belirlenir. Bu adım gerekleřtirilirken ařađıdaki  iřlem yapılır:

- Federeler, yeni federelerin gerekliliđinin tespit edilmesi ve mevcut federeler zerinde deđiřiklik ihtiyalarının belirlenmesi iin incelenir.

- Federasyonun tasarımına başlanır.
- Güvenli federeler için güvenlik hususlarının gerçekleştirilmesine başlanır.

Üçüncü adımda federasyon geliştirilmesi ve gerçekleştirimiyle ilgili detaylı bir plan çıkarılır. Bu planın temel hedefleri aşağıdaki gibidir:

- Federasyon amaçları ve gereksinimler üzerinde ortak bir anlayışın sağlandığının garantilenmesi
- Sistem geliştirilirken kullanılacak metodolojiler üzerinde anlaşma sağlanması
- Her federe için özel işlemlerin ve aşamaların belirlenmesi
- Her iş için hedeflenen bitiş süresinin belirlenmesi

2.2.4 Federasyon Geliştirimi

Bu kesim dört alt adımdan oluşur. İlk adımda federasyon amaçlarını gerçekleştirmek için federeler arası veri değişiminde kullanılacak Federasyon Nesne Modeli (*Federation Object Model-FOM*) geliştirilir. FOM geliştirilirken üç farklı yaklaşım kullanılabilir:

1. Aşağıdan yukarıya (*bottom-up*) geliştirme yaklaşımı kullanılabilir. Bu durumda OMDD (*Object Model Data Dictionary*), federasyon senaryosu ve federasyonun kavramsal modeli kullanılarak tamamen yeni bir FOM geliştirilir.
2. Tüm katılımcı federelerin SOM'larını birleştirerek FOM oluşturulabilir. Bu durumda SOM'ların uygulama alanı ile ilgili olmayan kısımları elenir.
3. Hedeflenen FOM'a en yakın olan SOM temel olarak alınıp, ilgisiz kesimleri çıkarılır ve FOM'u tamamlamak için diğer SOM'ların ilgili kesimleri eklenebilir .

FOM geliştirilirken, otomatik üretim araçlarının kullanılması geliştirme sürecini etkinleştirir [17]. HLA OML(*Object Model Library*), tekrar kullanılabilir nesne modeli kütüphanelerine erişim sağlar. OMDT (*Object Model Development Tool*) araçları

kullanarak da mevcut nesne modelleri deęiştirilebilir, genişletilebilir veya yeni modeller geliştirilebilir. Yine OMDT araçları kullanılarak tutarlılık, sözdizim kontrolleri yapılabilir, FED kütüğü oluşturulabilir, dięer nesne modeli geliştirme araçlarına arayüzler sağlanabilir ve çevrimiçi yardım adreslerine bağlantılar sağlanabilir.

İkinci adımda federasyon kuralları üzerinde anlaşma sağlanır. Bu adımın temel amacı, tümüyle tutarlı birlikte çalışabilir benzetimlerin elde edilmesidir. Bu adımda temel olarak aşağıdaki iki işlem gerçekleştirilir:

1. FOM'da belirtilemeyen, fakat federelerin uyması gereken kurallar belirlenir.
2. Federe düzeyinde ele alınacak ilk durum atama yordamları (*initialization procedures*), zamanuyumlama noktaları, kaydetme/geri yükleme koşulları gibi işlemsel hususlar üzerinde anlaşma sağlanır.

Üçüncü adımda federelerin aşağıdaki görevleri yerine getirecek şekilde düzenlenmesi gerekir:

- Nesnelerin ve bunlarla ilişkili davranışların federasyonun kavramsal modeline uygun olarak gösterilmesi
- FOM'da belirtilen kurallara uygun şekilde federasyon verisi üretilmesi ve deęişimi
- Zamanuyumlama, kaydetme/geri yükleme noktaları gibi federasyon kurallarına uyma

Dördüncü ve son adımda federasyon, bütünleştirme ve sınamaya hazırlanır.

2.2.5 Federasyon Bütünleştirme ve Sınama

Bu kesimde ilk olarak aşağıdaki adımları izleyerek federasyon çalışması planlanır.

- HLA federasyonunun çalışması için gerekli tüm çalışma planının tanımlanıp üretilmesi
- Test ve VV&A planlarının gözden geçirilmesi

- FEPW'da (Federation Execution Planning Workbook – Federasyon Çalışmasını Planlama Alıştırma Kitabı) tanımlanan şablonun doldurulması. FEPW, federasyonun performans gereksinimlerinin belirlenmesi için ortak yapısal bir mekanizma sağlar. Ayrıca HLA federasyonlarının federe performansı, *host* ve ağ gereksinimleri gibi diğer özelliklerini de tanımlar. FEPW'daki çizelgeler, bir federasyon geliştiricisinin federasyonu test ederken ve çalıştırırken ihtiyaç duyacağı tüm çalışma bilgilerini içerir. FEPW, FOM ve ilişkili FED kütüğü ile birlikte ele alındığında federasyon geliştirilmenin sınama ve bütünleştirme aşamalarına geçişte temeli oluştururlar.
- Güvenli federeler için güvenlik testlerinin ve değerlendirme planlarının oluşturulması
- Eğer performans artışı gerekiyorsa, RID (*RTI Initialization Data*) kütüğünün güncellenmesi

İkinci adımda federeler arası gerekli olan bağlantılar kurulur. Üçüncü ve son adımda ise, federasyon çalışmadan önce son kez sınanır.

2.2.6 Federasyonun Çalıştırılması ve Çıktıların Üretimi

Bu kesimde öncelikle federasyon çalıştırılır, federasyon çalışması esnasında çıktı verileri toplanır. Toplanan veriler sonraki adımda değerlendirileceklerdir.

2.2.7 Çıktı Analizi ve Sonuçları İnceleme

Bu kesimde amaç, federasyon çalışması esnasında oluşturulan çıktıları değerlendirerek, modelleme ve benzetim sisteminin kullanıcılarına (sponsor) raporlar oluşturmaktır. Bu raporlar ve geribildirimler, federasyonun kullanıcının ihtiyaçlarını karşılayıp karşılamadığına karar vermede çok önemlidir. İhtiyaçlar karşılanmamışsa gerekli bazı FEDEP adımları tekrarlanır.

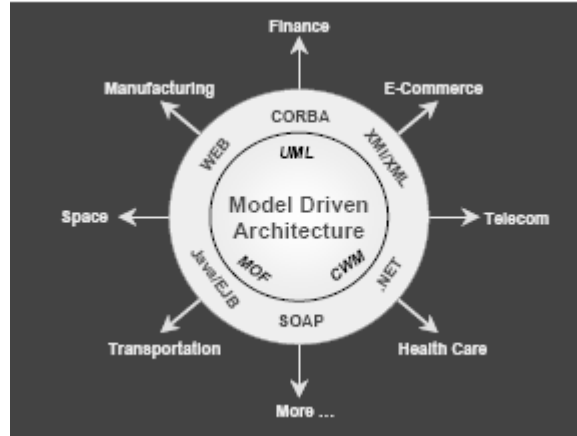
2.3 MODELE DAYALI MİMARİ

OMG (*Object Management Group*) yazılım dünyasında pek çok yeni kavramın ve ürünün oluşumunda fikir kaynaklığını yapmış çok üyeli, çok uluslu bir kuruluştur. Amacı küresel bir bilgi aracını mümkün kılmak olan bu kurum, altyapı standardı

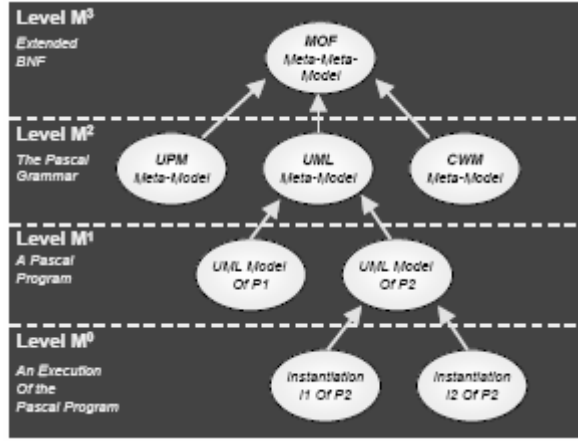
CORBA (Common Object Request Broker Architecture) ve modelleme standardı UML'nin ardından, meta modelleme standardı olarak MDA'yı (Model Driven Architecture) yazılım dünyasına tanıtmıştır. MDA, programlama dilinden, platformdan, ara katman altyapılarından bağımsız olarak sistem tasarlama amacıyla, tanımlanmıştır. Şekil 2-2'de [14] görüleceği gibi Modele Dayalı Mimari'nin temelinde OMG'nin diğer bazı modelleme standartları yer almaktadır. Bunlar UML (Unified Modelling Language), MOF (Meta-Object Facility ve CWM (Common Warehouse Meta-model)'dir.

MDA, UPM (Unified Process Model), UML ve CWM'yi ortak bir modelle ifade ederek birleştirir. Böylece yazılım ürünlerinin merkezinde yer alan MDA kullanımı ile, Şekil 2-3'de görülen çeşitli seviyeler için platformdan bağımsız sistem modelleri ve kaynak kodunu üretebiliriz.

MDA mimarisi kullanılarak geliştirilen uygulamalarda, öncelikle sağlam bir model geliştirilir. Bu model programlama dilinden, üreticiden ve çalışma ortamından bağımsız olmalıdır. Bu model uygulamadaki kavramların meta modelidir. MDA bu meta modeli geliştirmemiz için kavramları sunar [18].



Şekil 2-2 OMG Model Driven Architecture



Şekil 2-3 OMG 4-Katmanlı Modelleme Yiğini

MDA'nın temel amacı, sistem için detaylı, platformdan bağımsız modelin oluşturulması, daha sonra bu modelden farklı platformlar için sistem modelinin üretilmesidir. MDA mimari yaklaşımını uygularken ilk adım olan platformdan bağımsız modelin (PIM – *Platform Independent Model*) oluşturulurken sistemin tüm iş kurallarını, güvenlik ve performans gibi özellikleri ele alınır. Sonraki adımda, seçilen bir platform için PIM'den bir veya daha fazla platforma bağlı modele (PSM – *Platform Specific Model*) geçiş yapılır [18]. Örneğin IEEE 1516.2 OMT'ye uygun nesne modelleri geliştirmeye yarayan bir araç için oluşturulacak PSM, ilgili belirtimde geçen DTD'ye (hla.dtd) uygun olacaktır. Son geliştirme adımı da her bir PSM'nin uygulama koduna dönüştürülmesidir.

Bu tez kapsamında geliştirilen MBGA (Modelleme ve Benzetim Geliştirme Aracı) aracının Rational Rose ile hazırlanan ilk modeli PIM; EMF ile hazırlanan Ecore modeli ise PSM'dir. Geliştirilen Ecore modeli IEEE 1516.2 OMT'ye büyük ölçüde sadık kalınarak geliştirilmiştir. IEEE 1516.2'den farklı olduğu noktalar programın eksikleri olarak değerlendirilmelidir. Zira, bu eksiklikler kullanıcıya müsaade edilen grafiksel tasarlama imkanları noktasındadır. Ecore modelinin FOM'a çevrimi esnasında, görsel olarak modellenmesine izin verilmeyen HLA nesnelere de kabul edilen değerleri ile FOM kütüğünde yer almaktadırlar. Bu değerler, gerek görüldüğü takdirde, herhangi bir metin düzenleyici yazılımla değiştirilebilir.

MDA ile yazılım geliştirme, genel geliştirim yazılım süreçlerine benzemekle beraber, PIM'den nihai ürüne kadarki her bir adım sonucunda oluşan ürünler

farklılık göstermektedir. MDA'da geliştiriciden beklenen, öncelikle sistemi çok iyi tanımlayan bir modeli oluşturmaktır.

2.4 ECLIPSE PLATFORMU

2.4.1 Eclipse

Eclipse, açık kaynaklı bir geliştirme platformu projesidir. Eclipse, en bilinen şekliyle IDE (Integrated Development Environment) olmasının yanısıra aynı zamanda ileri seviyede entegre olmuş yeni geliştirme araçlarının gerçekleştirimini mümkün kılacak bazı temel nesne modelleri ve soyutlamalardan oluşan bir çerçeve sunar. Eclipse platformu projesi dört ana projeden oluşur. Bunlar temel Eclipse platformu ve Eclipse geliştirme araçları (Eclipse), Eclipse Araçları (Eclipse Tools), Eclipse araştırma ve eğitim faaliyetleri (Eclipse Technology) ve web uygulama sunucusu geliştirme araçlarından (Eclipse Web Tools Platform) oluşur [9].

Eclipse projesi, IBM, Borland, Oracle, SAP, OMG gibi 40'ı aşkın yazılım geliştirme araçları üreticilerinin üyeliğiyle oluşmuş olan, Eclipse Birliği (Eclipse consortium) tarafından yürütülür. Eclipse projesi, Eclipse platformu üzerine geliştirilmiş olan, Java programlama dili ile entegre geliştirme ortamı Eclipse Yazılım Geliştirme Alet Takımı'nı (SDK – Software Development Kit) üreterek Eclipse tabanlı araçların üretimi için komple bir geliştirme ortamı sunuyor. Dolayısıyla Eclipse platformu genişletilebilir platformdur.

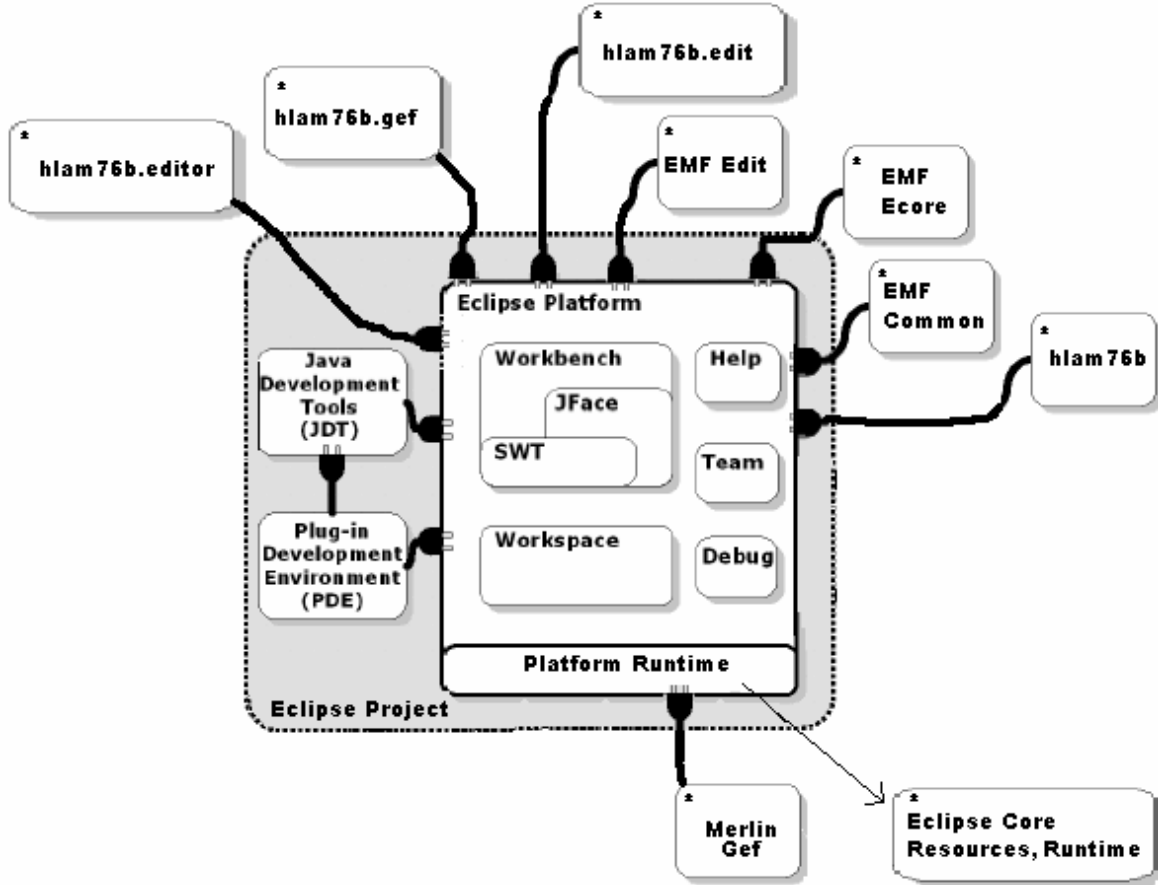
Bu tez kapsamında geliştirilen modelleme aracı da Eclipse IDE'si kullanılarak, Eclipse platformuna bir eklenti olarak geliştirilmiştir. Takip eden bölümde Eclipse eklenti mekanizması anlatılacaktır.

2.4.2 Eclipse PDE

Eclipse tabanlı araçlar Eclipse platformuna entegre olmuş Eclipse eklentilerinden (plug-ins extensions) oluşur. Eclipse plug-in mimarisine göre, Eclipse eklentileri genişleme noktaları (extension points) tanımlarlar. Bu genişleme noktaları başka Eclipse eklentilerince genişletilerek (extend) yeni Eclipse eklentileri oluşturulabilir.

Eclipse platformunun kendisi, küçük bir çekirdek bir çalışma zamanınının (a small core runtime) üzerine geliştirilmiş Eclipse eklentilerinden oluşur [9].

Şekil 2-4’de Eclipse eklentilerinin Eclipse platformu çalışma zamanı ile ilişkileri ve birbirleri ile ilişkileri görülmektedir.



Şekil 2-4 Eclipse Platformu Mimarisi (Griffin, [9]’den değiştirilerek)

2.4.3 Eclipse Modelleme Çerçevesi

EMF (Eclipse Modeling Framework – Eclipse Modelleme Çerçevesi) yapısal bir modele dayanan yazılım araçları ve çeşitli uygulama yazılımları geliştirmek için kullanılan, bir modelleme ve kod üretme çerçevesidir. Nesneye dayalı olarak hazırlanan modeller, EMF ile kısa zamanda verimli, düzgün ve güncellenebilen Java kodlarına dönüştürülebilir. EMF’nin meta modeli Ecore olarak adlandırılır. Kullanıldığında üretkenliği artıran EMF, ayrıca aşağıdaki faydaları da sunmaktadır:

- Modelde değişikliği bildirim (Model Change Notification)

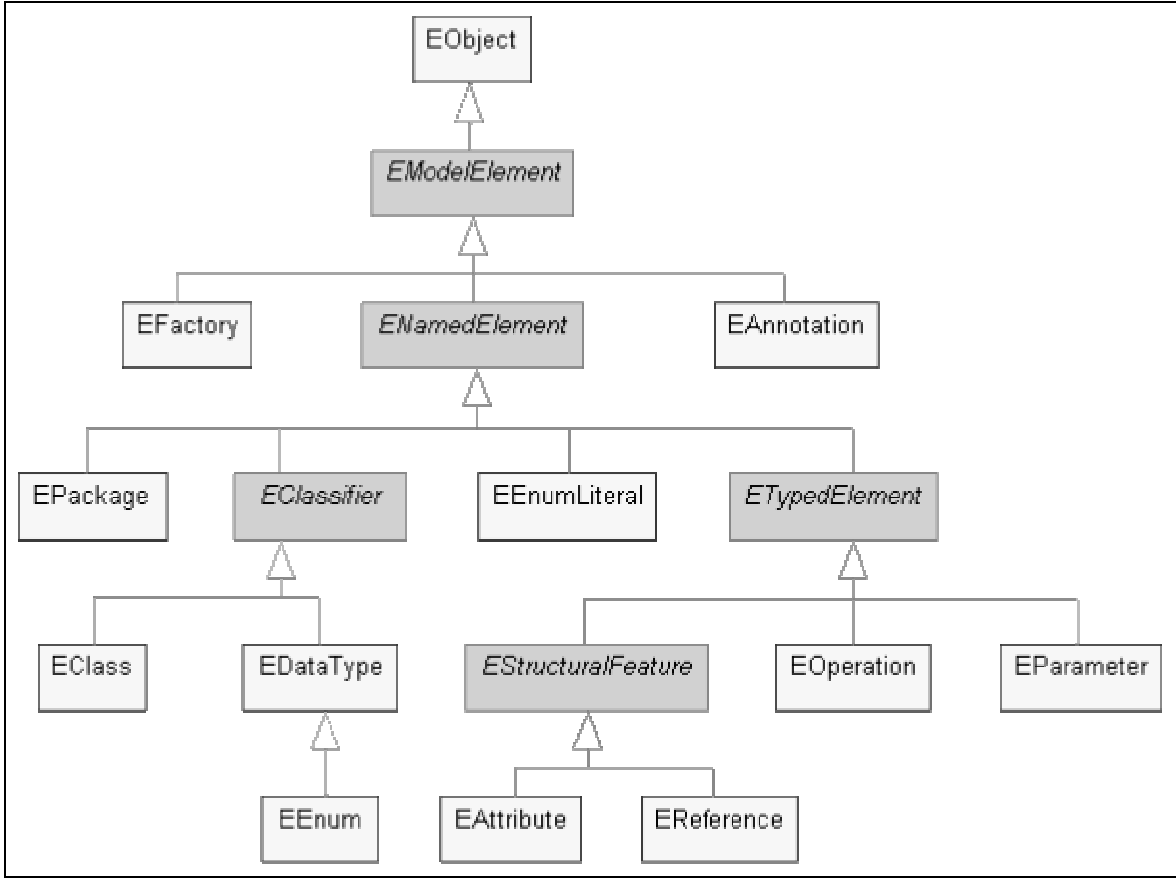
- XMI biçiminde kütükleme desteği (XMI-based persistence support)
- Şema tabanlı XML kütükleme desteği (Schema-based XML serialization)
- Model geçerliğini denetleme çerçevesi (Model validation framework)
- EMF nesnelerini oldukça soysal şekilde işlemeyi sağlayan reflective API(a very efficient reflective API for manipulating EMF objects generically)

EMF iki temel çerçeveden oluşmaktadır:

- Core Framework
- EMF.Edit

Core Framework, EMF modelinin Java sınıflarının üretimi için çalışma zamanı destek sağlar. EMF.Edit ise, EMF Core Framework üzerine gelişmiştir ve, EMF Ecore modelinin gösterilmesini ve komutlarla gerialınabilir olarak düzenlenmesini sağlayan adaptör Java sınıflarının üretimini yapar.

Şekil 2-5'de Ecore sınıf hiyerarşisinin tamamı görülmektedir. Koyu renkli kutular soyut sınıfları göstermektedirler [6].



Şekil 2-5 EMF Ecore Modeli Nesneleri (Artifacts)

2.4.3.1 EMF - MOF İlişkisi

EMF başlangıçta OMG'nin MOF belirtiminin bir gerçekleştirimi olarak düşünülmüş olsa da, bugün EMF ayrı bir modelleme çerçevesidir. OMG'nin MOF çekirdek API'sının (Application Programming Interface) Java ile gerçekleştirimi olan EMF'de MOF'a benzeyen meta model Ecore olarak adlandırılır.

EMF ile MOF 2.0 belirtiminde EMOF(Essential MOF) olarak adlandırılan MOF modelinin bir alt kümesi oldukça benzerdir. Sadece isimlendirmede bazı farklılıklar vardır. EMF, EMOF kütüklerini sorunsuzca okuyabilmektedir.

2.4.3.2 Uygulama Geliştirirken EMF Modelinin Hazırlanması

EMF kullanarak uygulama geliştirilirken, geliştirilecek uygulamanın platformdan bağımsız modelini (PIM), EMF'nin modeli olan Ecore modeline çevirmek gerekmektedir. Bu işlem çeşitli şekillerde yapılabilir. Uygulama geliştiricinin

tecrübe seviyesine ve elindeki mevcut araçlara göre herhangi bir yol seçilebilir. Aşağıda EMF Ecore modeli oluşturma seçenekleri sıralanmıştır.

- EMF Ecore Modelin formal ifade biçimi olan XMI dökümanını doğrudan bir XML düzenleyici uygulama ile oluşturmak
- XMI dökümanını Rational Rose gibi bir modelleme aracından ihraç ederek elde etmek
- EMF Ecore Modeli özelliklerini açıklamalı Java arabirimleri ile hazırlamak
- Modelin kütüklere aktarım şeklini tanımlayan XML şemaları kullanmak

Bu tez kapsamında seçilen metod, Rational Rose ile hazırlanan UML modelinden EMF Ecore modelini oluşturmak olmuştur.

2.4.4 Grafiksel Düzenleme Çerçevesi

Grafiksel Düzenleme Çerçevesi (Graphical Editing Framework - GEF), mevcut bir uygulama meta modelinden hızlı şekilde zengin grafiksel editörler üretmeyi mümkün kılan bir altyapıdır. GEF iki adet Eclipse eklentisinden oluşur. Bunlar *org.eclipse.draw2d* ve *org.eclipse.gef* 'dir. *org.eclipse.gef*, *org.eclipse.draw2d*'ye gereksinim duyar.

org.eclipse.draw2d, koordinat sistemi, saydam çizim katmaları, çizme alanı, imleç, yazıcıdan çıktı alma gibi işlevleri yapar. *org.eclipse.gef* ise meta modele göre modellenecek olan nesnelerin paletini oluşturur. Paletten sürükle-bırak yoluyla nesneleri görsel olarak oluşturmayı sağlar ve oluşturulan görsel modeli iki boyutlu grafiksel olarak gösterir. Aynı zamanda ağaç yapısında gösterme de yapabilir. Ekranda gösterilen grafiksel olarak modellenmiş model görüntüsü (View) ile uygulama modelini (Meta Model) birbirine bağlayan bir denetleyici (Controller) vazifesi görür.

MVC (Model View Controller) tasarım örüntüsünü kullanan GEF, model görüntüsündeki (View) değişikliklerin modele aktarılmasını sağladığı için, pek çok görsel modelleme aracının geliştirilmesinde kullanılabilecek bir altyapıdır. Kullanıcı

arayüzü geliştiriciler (GUI builders), sınıf çizeneği düzenleyiciler (class diagram editors), durum makinaları (state machines) GEF kullanılarak geliştirilebilecek uygulama türlerinden sadece bir kısmıdır [7].

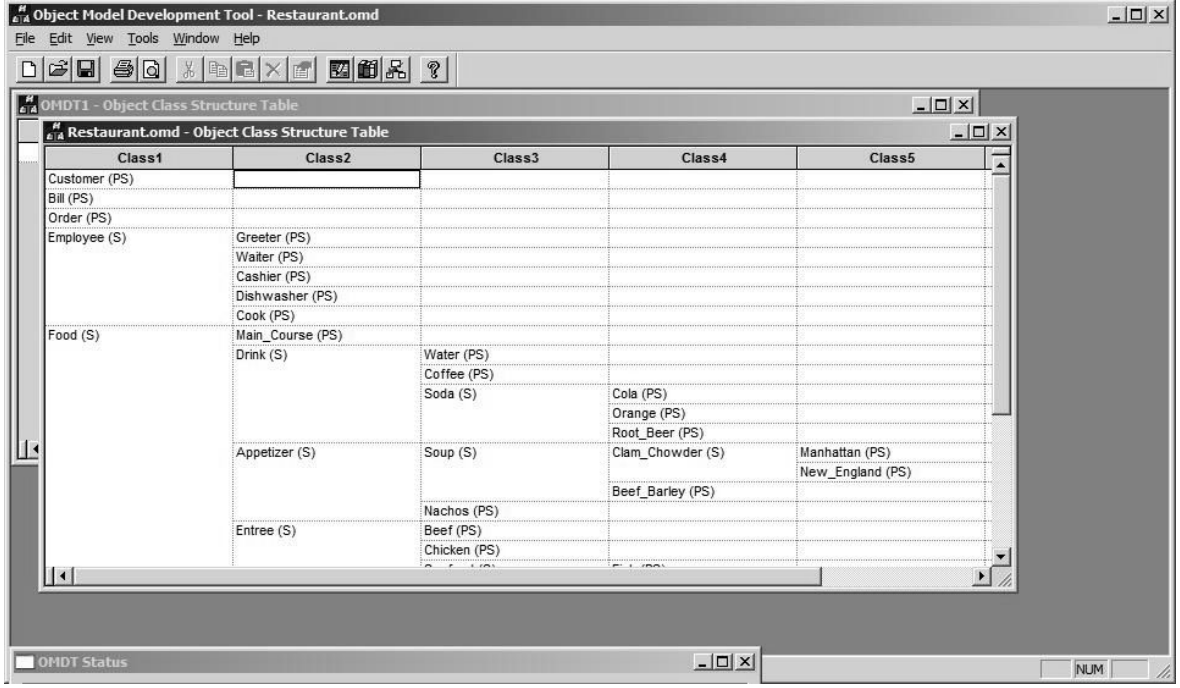
Bu tez çalışması kapsamında da GEF kullanılarak HLA nesnelere görsel olarak modellenmesi yapılmıştır. Tez çalışmasının sonraki bölümünde HLA modelleme için geliştirilmiş araçlar incelenerek kıyaslanacaktır. Bunun ardından da bu tez çalışması kapsamında geliştirilen HLA için görsel modelleme aracı incelenecektir.

3 MEVCUT HLA MODELLEME ARAÇLARI

HLA için modelleme ve benzetim sistemleri tasarlanıp gerçekleştirilirken; bir yandan da bu süreci kolaylaştırmak, hızlandırmak ve kusursuz hale getirmek için DMSO öncülüğünde, HLA için geliştirme araçları da üretimi de başlamıştır. OMDT (*Object Model Development Tool*), Visual OMT ve SIMplicity yazılımları modelleme ve benzetim sistemleri geliştirenlerin kullandığı başlıca ürünlerdir olarak öne çıkmaktadırlar. İlerleyen bölümlerde bu araçları incelenecek ve ardından bu araçlarla bu tez kapsamında geliştirilen modelleme aracının karşılaştırması yapılacaktır. Çizelge 3-1'de karşılaştırma sonuçları gösterilmiştir.

3.1 OMDT

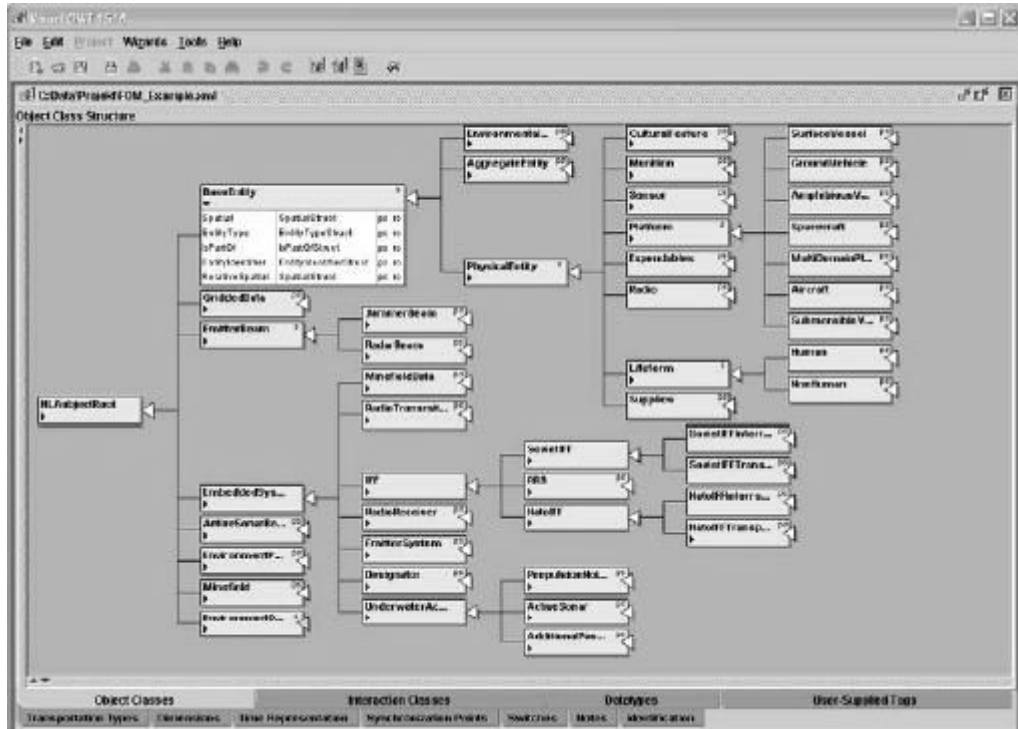
OMDT (Object Model Design Tool), DMSO tarafından geliştirilmiştir ve OMT oluşturma sürecini iyileştirmektedir. Bu araç kullanılarak FOM/SOM tasarlanabilir, modellerin doğruluğu sınanabilir. OMDT'de modelleme çizelgeler üzerinden yapıldığından modelleme işlemi çizelgelere alan ekleyerek gerçekleştirilir. Modelleme sadece nesne modeli oluşturma düzeyinde desteklenmektedir. Aracının genel görünümü Şekil 3-1'deki gibidir.



Şekil 3-1 OMDT Ekran Görüntüsü

3.2 VisualOMT

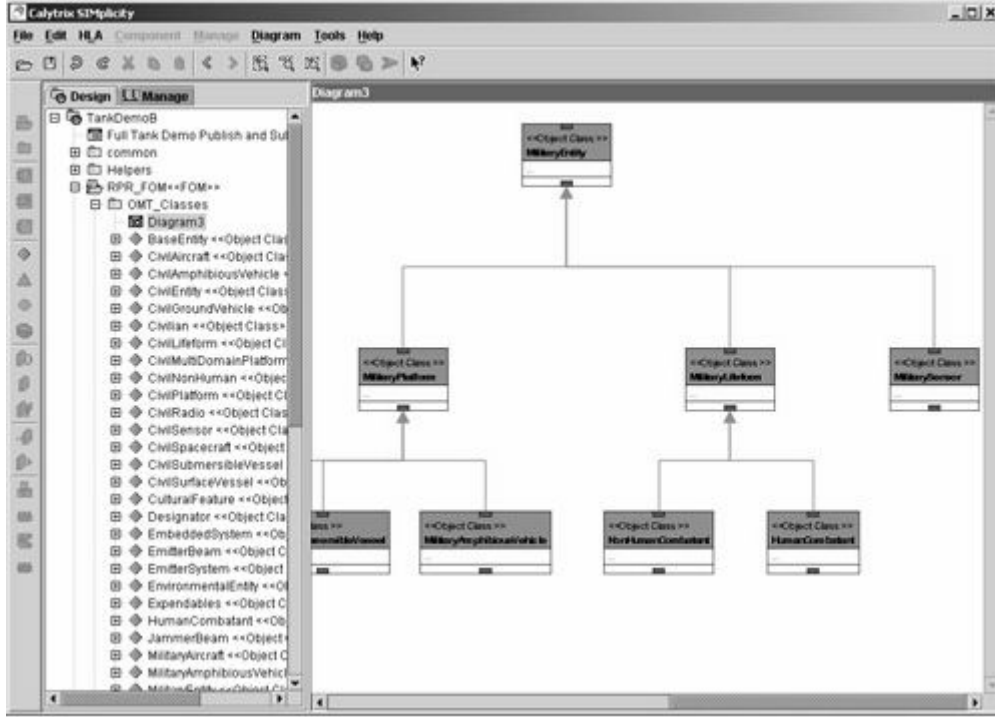
Visual OMT, *Pitch* firması tarafından geliştirilmiştir. UML türevi bir modelleme dili kullanır ve nesne modeli tasarımı yapabilir. OMDT'ye göre avantajı görsel modellemeye olanak sağlamasıdır. Nesne ve etkileşim sınıfları UML sınıf çizimlerdeki sınıf gösterimine benzer bir şekilde gösterilmiştir. Bu araçla numaralandırılmış ve karmaşık veri türleri, etkinlik alanları görsel olarak modellenemez, formlardan verilerini girmek suretiyle modellenebilirler. Visual OMT aracının görünümü Şekil 3-2'deki gibidir. Aracın yeteneği nesne modeli oluşturmayla sınırlıdır [16].



Şekil 3-2 Visual OMT Aracından Ekran Görüntüsü

3.3 SIMplicity

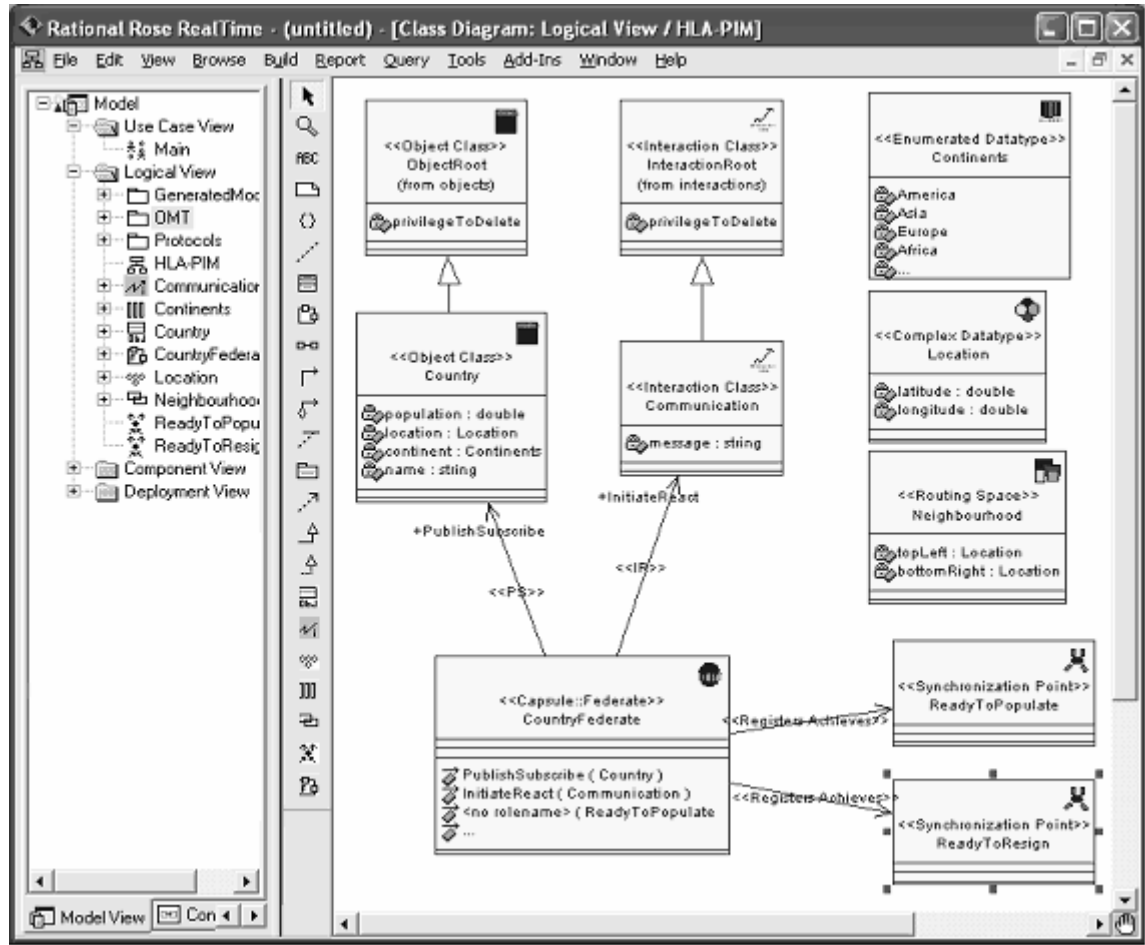
SIMplicity, tasarım, otomatik kod üretme, bütünleştirme ve çalıştırma süreçlerini ele alır. Tüm HLA kavramlarını bileşen olarak modeller. Federelerin birbirleriyle ve nesne/etkileşim sınıflarıyla ilişkilerini modellemeye olanak sağlar. Modelleme esnasında nesne/etkileşim sınıflarının ilişkileri sınıf çizeneğiyle, federelerin bu sınıflarla ilişkisi ise çizelge şeklinde gösterilmiştir. Ayrıca federelerin genel özellikleri de modelleme esnasında belirtilir. SIMplicity, oluşan modellerden, seçilen platform için uygulama kodunun bir şablonunu oluşturur. Ancak bu uygulama kodu sadece RTI ile etkileşimi sağlayan kesimi içermektedir. Uygulama geliştirici iş mantığını bu kodun içinde doğru yere yazmak zorundadır. Aracın görünümü Şekil 3-3'deki gibidir [1;15].



Şekil 3-3 Simplicity Aracından Ekran Görüntüsü

3.4 RTEI destekli HLA Modelleme Aracı

Rational Rose RealTime'a eklenti olarak geliştirilmiştir [2]. Nesne modeli oluşturma ve federe modelleme kabiliyetleri vardır. UML'i etkin olarak kullanır ve federe davranışını modelleyebilir. *Rational Rose RealTime*'ın tüm özelliklerini kullanabilmesi bir avantaj olarak görünse de, ticari bir ürün olan *Rational Rose RealTime* aracına bağımlı olması dezavantajdır. Şekil 3-4'da bu aracın ekran görüntüsü görülmektedir.



Şekil 3-4 RRTEI destekli HLA Modelleme Aracı Ekran Görüntüsü

3.5 Modelleme Araçlarının Karşılaştırması

Çizelge 3-1’de mevcut HLA modelleme araçları ve bu tez çalışmasında geliştirilmiş olan modelleme aracı kıyaslanmıştır. Bu çizelgede önceki bölümlerde incelenen araçların geliştirme mimarileri, modelleme ve benzetim sistemlerini görsel ya da çizelgelerle modelleyebilme kabiliyetleri, kendi başına bir uygulama olarak ya da entegre bir araç setinin parçası olmaları hakkında bilgiler verilmiştir. Ayrıca araçların açık kaynak kodlu olanları ile ilgili bilgiler de verilmiştir.

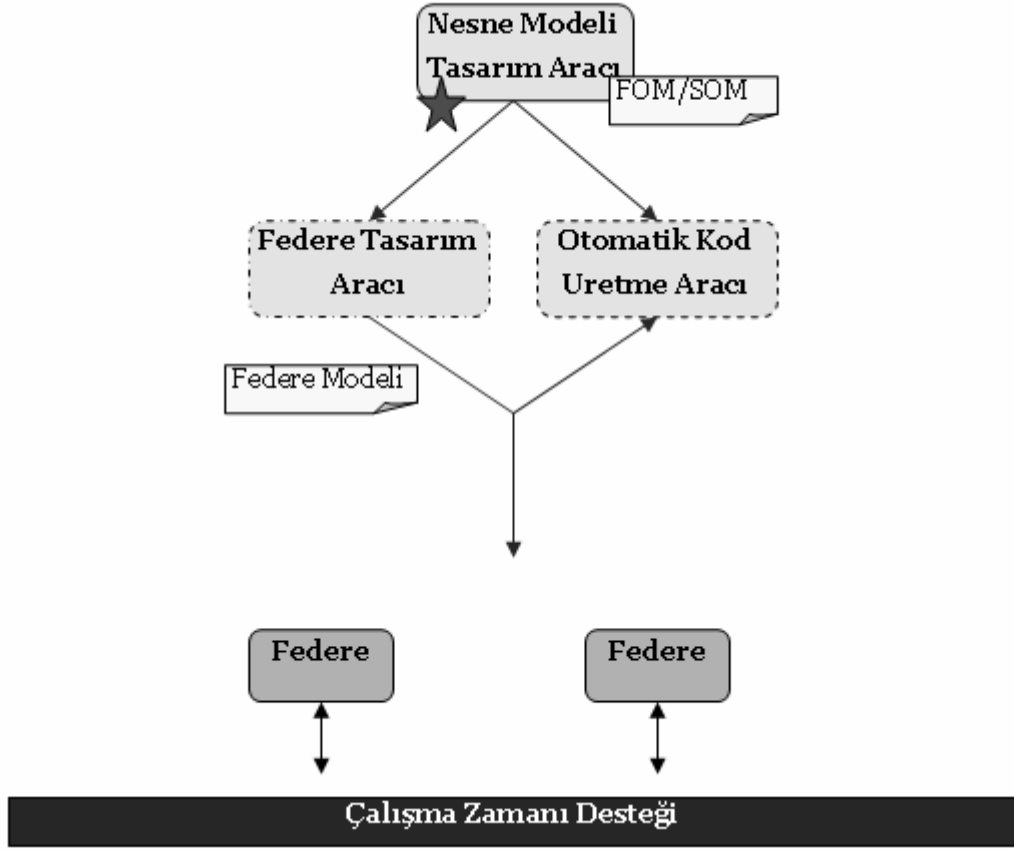
Çizelge 3-1 HLA için Modelleme Araçlarının Karşılaştırılması

ÖZELLİKLER	OMDT	VisualOMT	SIMplicity	RRTEI	HLAM76B
Görsel Modelleme	-	+	+	+	+
Çizelgelerle Modelleme	+	+ ^F	+	+ ^F	+ ^P
Bağımsız Uygulama	+	+			
Entegre Uygulama:			+	+	+
Tasarım			+	+	+
Davranış Modelleme			-	+	- ^G
Kod Üretme			+ ^D	-	- ^G
Bütünleştirme			+	+	- ^G
Çalıştırma			+	+	- ^G
MDA Mimarisi	?	?	+	+	+
Açık Kaynak	-	-	-	-	+

4 HLA İÇİN YENİ BİR MODELLEME ARACI GERÇEKLEŞTİRİMİ

Bu tez kapsamında geliştirilen HLA için modelleme aracı HLAM76B MBGA (Modelleme ve Benzetim Geliştirme Aracı), incelenen örneklerden en çok VisualOMT'ye benzemektedir. Fakat kullanılan mimarinin MDA olması ve kullanılan EMF çerçevesinin esnek ve gelişmelere açık olması sayesinde ürün federe modelleme, federe davranış modelleme ve federe kodu üretme özelliklerinin de eklenmesiyle SIMplicity uygulamasına yetenekler olarak yaklaşacaktır.

Aracın HLA için modelleme ve benzetim sistemleri tasarlama bağlamındaki konumu Şekil 4-1'da yıldızlı işaretlenen araç olarak gösterilmiştir.



Şekil 4-1 Aracın HLA Çalışma Ortamındaki Yeri

4.1 Araç Setinin Sağladığı Avantajlar

Geliştirilen araç setinin mevcut araçlara göre sağladığı avantajlar şunlardır:

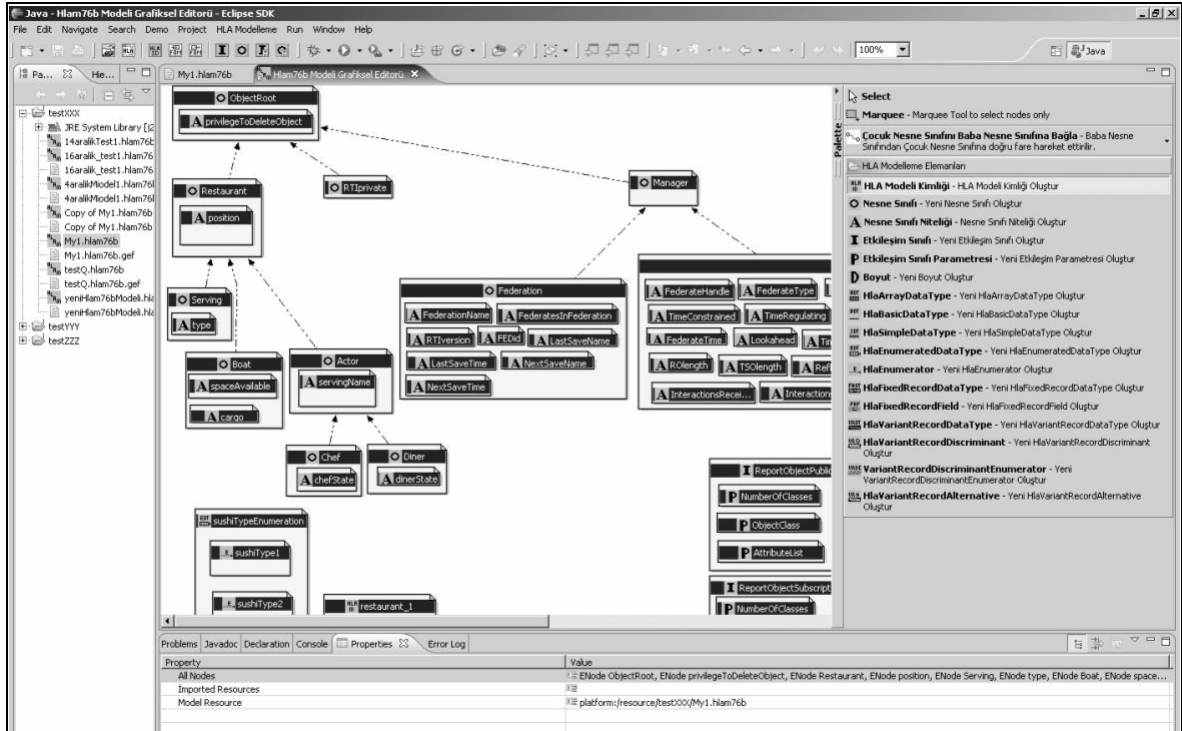
- Sürekli geliştirilen, açık ve ücretsiz bir platform olan Eclipse üzerinde geliştirilmiş olması
- MOF türevi olan EMF Ecore modelini kullanması
- IEEE 1516.2 OMT FOM export/importuna imkan vermesi

4.2 Araç Setini Geliştirme Adımları

Bu tez çalışması kapsamında geliştirilen modelleme ve benzetim geliştirme aracının HLA için nesnelerin görsel olarak tanımlanmasını temin etmesi gerekmektedir. Bu ana amaca uygun olarak HLA nesnelerinin tamamını içerecek şekilde bir meta model gerçekleştirimi bir ön koşuldur. MDA uygulayarak geliştirilen aracın tasarımı ve gerçekleştirimi de meta modelin hazırlanmasıyla başlar.

Geliştirilen araç setinin modelleme aşamalarında bir görsel desteğine ihtiyacı vardır. Gerçekleştirim ortamı olarak Eclipse platformuna ve Java programlama diline karar verilmiştir. Görsel modelleme desteği için de EPL lisanslı bir teknoloji olan GEF kullanılacaktır.

Şekil 4-2'de modelleme aracının GEF yardımıyla oluşturulmuş grafiksel arayüzü görülmektedir.



Şekil 4-2 Modelleme Aracının Ekran Görüntüsü

Araç seti gerçekleştirilirken izlenen yol aşağıdaki gibidir:

1. HLA'daki nesne sınıfı, etkileşim sınıfı, numaralandırılmış veri türü, karmaşık veri türü, boyut gibi HLA yapılarının modellendiği meta model olan hlam76b.mdl dosyasından EMF modeli ve EMF Model Eklentisi oluşturulur.
2. EMF Model Eklentisi oluşturulduktan sonra EMF Edit, EMF Editor ve EMF Gef Eclipse eklentileri de EMF, GEF ve Merlin Generator altyapıları kullanılarak geliştirilir.
3. Fonksiyonel tasarım nesne modelinin görsel modellemesine ve FOM/SOM üretmeye imkan verecek şekilde tamamlandıktan sonra, görsel ortamın menüyle ve düğmelerle zenginleştirilmesi ve aynı zamanda ekranda görüntülenecek simgelerin ve yazıların sade ve doğru bir Türkçe ile hazırlanması
4. Görsel gerçekleştirimin ardından Nesne modelinden FOM / SOM üretme yeteneği gerçekleştirilmesi. Modellenen HLA nesnelere dahili veri biçiminde kütüklerde tutulabilecekleri gibi, IEEE 1516.2 OMT'ye uygun FOM biçiminde dışarıya da aktarılabilirler. Aynı şekilde dışarıdan alınan FOM da dahili veri biçimine çevrilebilir, üzerinde düzenleme yapılabilir.

4.3 HLA Nesnelere İçin Geliştirilen Meta Model

Tez kapsamında geliştirilen HLA için modelleme ve benzetim aracının, MDA mimarisine uyularak geliştirildiği için esas aldığı meta model olmalıdır. Bu meta model, modelleme ve benzetim aracı ile modellenebilecek HLA nesnelere, veri yapılarını ve bunların ilişkilerini platformdan bağımsız bir şekilde, UML biçiminde tanımlar.

HLA için gerekli olan model elemanları aşağıda sıralanmıştır:

- Nesne Sınıfı ve Nesne Sınıfı Niteliği
- Etkileşim Sınıfı ve Etkileşim Sınıfı Parametresi
- Karmaşık Veritürü ve Karmaşık Veri Türü Alanı
- Numaralandırılmış Veritürü ve Numaralandırılmış Veri Türü Elemanı
- Sabit Kayıt Veritürü ve Sabit Kayıt Veri Türü Elemanı

- Deęişken Kayıt Veri Türü ve Deęişken Kayıt Veri Türü Elemanı
- Boyut

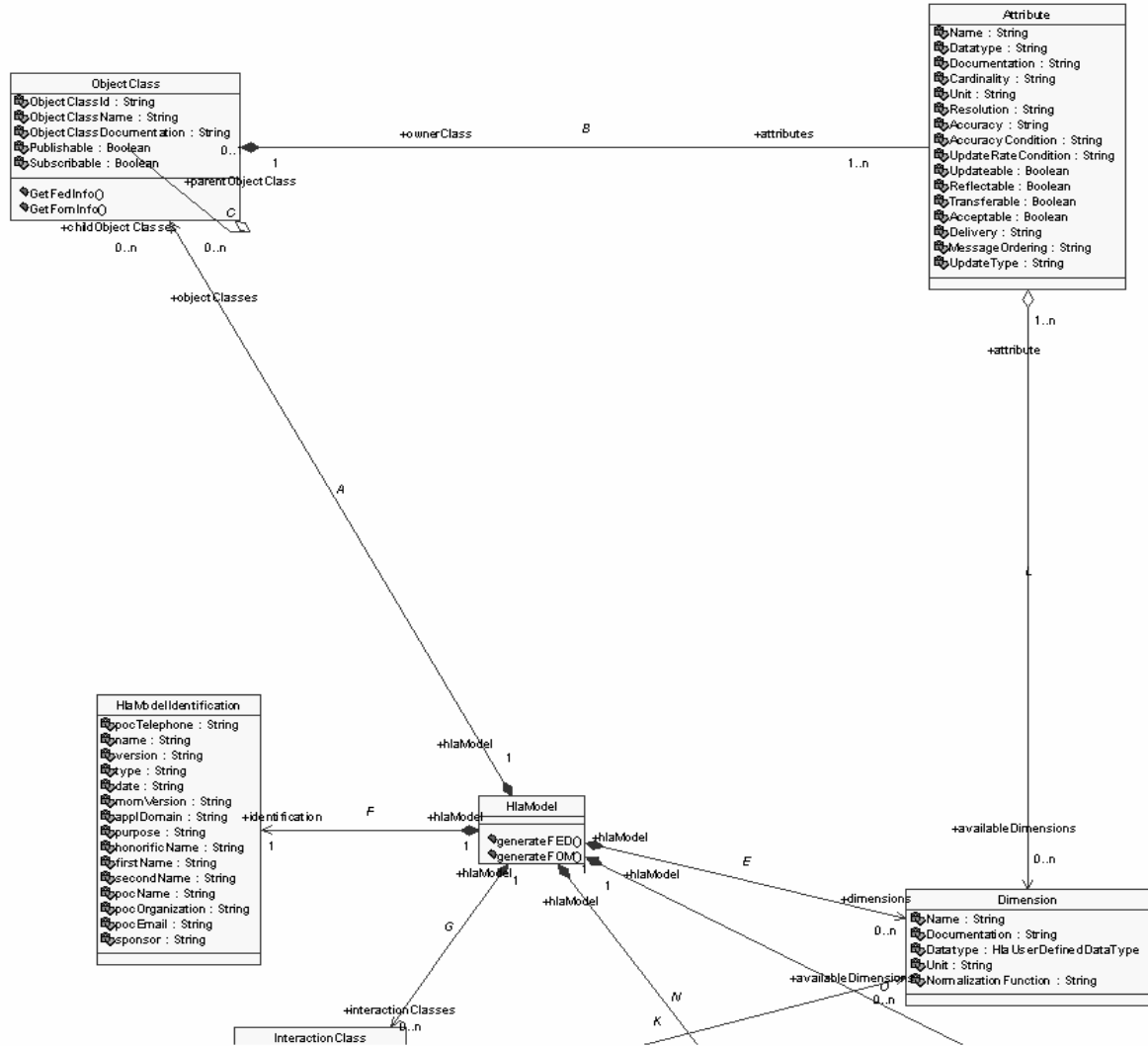
Yukarıda sıralanan her HLA model elemanı için bir *Rose* model elemanı tanımlanmıştır. Bu *Rose* model elemanları birleştirilerek *h1am76b.mdl* *Rose* modeli oluşur. Bu *Rose* modeli, geliştirilen grafiksel modelleme aracı ile hangi HLA nesnelerinin modellenebileceğini belirler (Bkz. Şekil 4-3).

4.3.1 HLA Meta Modeli Elemanları - HLA Nesne Sınıfı

HLA nesne sınıfı, hlam76b paketinde bulunan aşağıdaki arayüzü gerçekleştirir:

```
public interface ObjectClass extends EObject {
    String getObjectClassId();
    void setObjectClassId(String value);
    String getObjectClassName();
    void setObjectClassName(String value);
    String getObjectClassDocumentation();
    void setObjectClassDocumentation(String value);
    Boolean getPublishable();
    void setPublishable(Boolean value);
    Boolean getSubscribable();
    void setSubscribable(Boolean value);
    EList getAttributes();
    ObjectClass getParentObjectClass();
    void setParentObjectClass(ObjectClass value);
    EList getChildObjectClasses();
    void GetFedInfo();
    void GetFomInfo();
}
```

Şekil 4-4'de HLA nesne sınıfının diğer HLA model elemanları ile ilişkileri görülmektedir.



Şekil 4-4 Meta model - HLA Nesne Sınıfı

4.3.2 HLA Meta Modeli Elemanları - HLA Nesne Sınıfı Niteliği

HLA nesne sınıfı niteliği, hlam76b paketinde bulunan Attribute arayüzünü gerçekleştirir:

```
public interface Attribute extends EObject {
    String getName();
    void setName(String value);
    String getDatatype();
    void setDatatype(String value);
    String getDocumentation();
    void setDocumentation(String value);
    String getCardinality();
    void setCardinality(String value);
    String getUnit();
```

```

void setUnit(String value);
String getResolution();
void setResolution(String value);
String getAccuracy();
void setAccuracy(String value);
String getAccuracyCondition();
void setAccuracyCondition(String value);
String getUpdateRateCondition();
void setUpdateRateCondition(String value);
Boolean getUpdateable();
void setUpdateable(Boolean value);
Boolean getReflectable();
void setReflectable(Boolean value);
Boolean getTransferable();
void setTransferable(Boolean value);
Boolean getAcceptable();
void setAcceptable(Boolean value);
String getDelivery();
void setDelivery(String value);
String getMessageOrdering();
void setMessageOrdering(String value);
String getUpdateType();
void setUpdateType(String value);
ObjectClass getOwnerClass();
void setOwnerClass(ObjectClass value);
EList getAvailableDimensions();
}

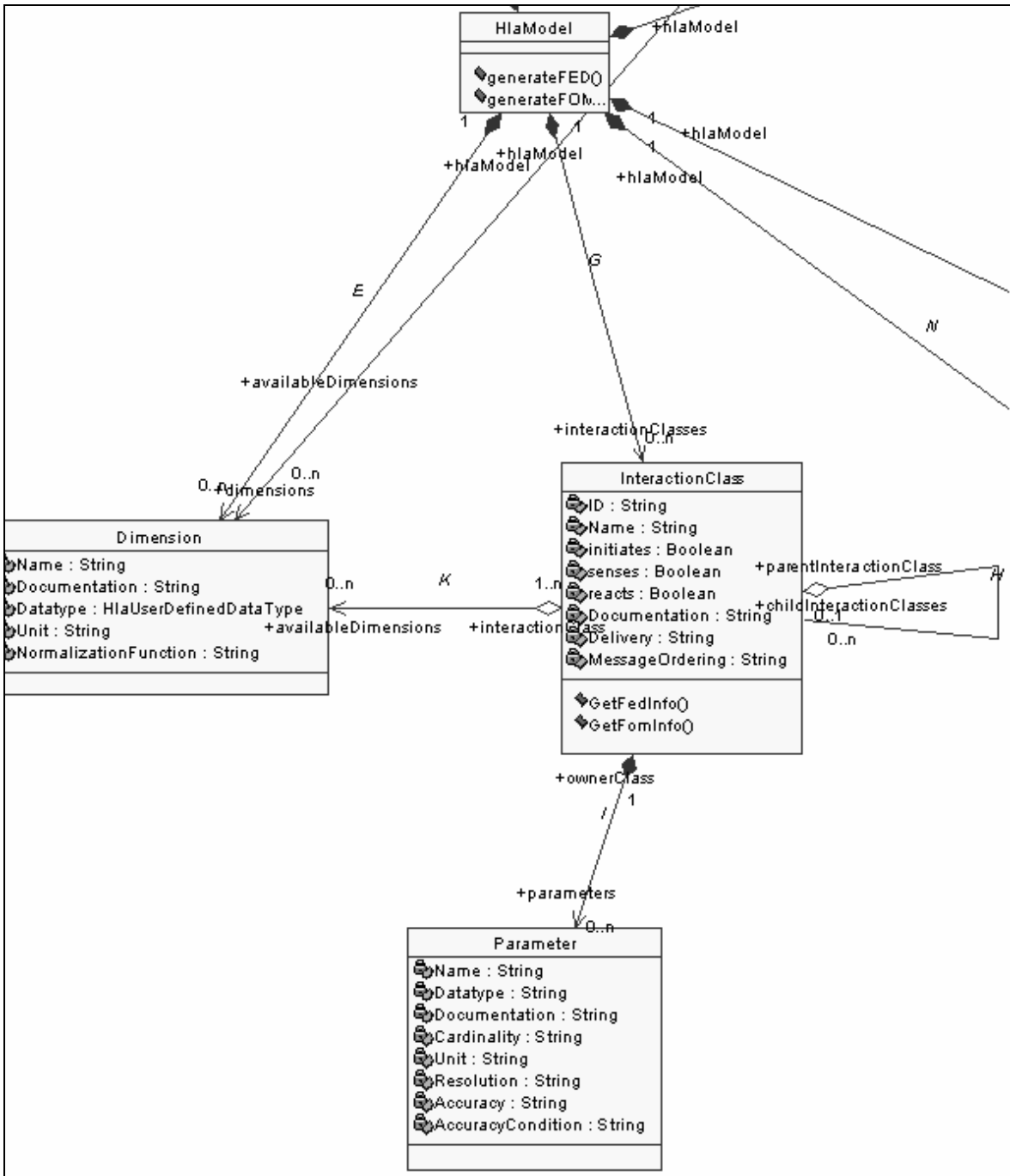
```

Diğer HLA nesneleri için de benzer arayüzler IEEE 1516.2 OMT'ye göre gerçekleştirilmiştir. İlerleyen bölümlerde bu HLA nesnelere ve diğer nesnelere ilişkilerini gösteren UML model tasarımları verilecektir.

Şekil 4-4'de HLA nesne sınıfı niteliğinin diğer HLA model elemanları ile ilişkileri görülmektedir.

4.3.3 HLA Meta Modeli Elemanları - HLA Etkileşim Sınıfı

Şekil 4-5'de HLA etkileşim sınıfının arayüzleri görülmektedir. HLA etkileşim sınıfının gerçekleştirim sınıfı da hlam76b pakedindedir.



Şekil 4-5 Meta model - HLA Etkileşim Sınıfı

4.3.4 HLA Meta Modeli Elemanları - HLA Etkileşim Sınıfı Parametresi

Şekil 4-5’de HLA etkileşim sınıfı parametresinin diğer HLA model elemanları ile ilişkileri görülmektedir.

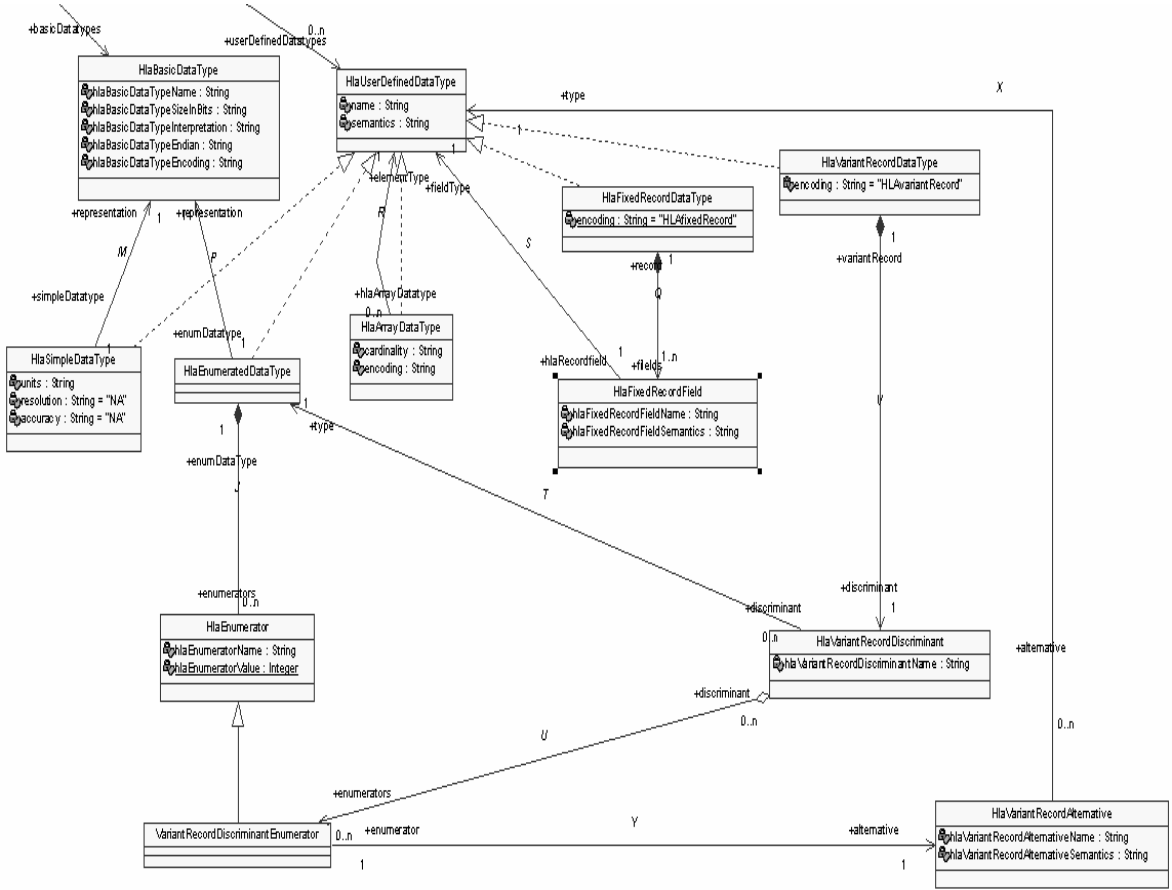
4.3.5 HLA Boyutu

Şekil 4-4 ve

Şekil 4-5’te HLA model elemanlarının HLA boyutu ile ilişkileri görülmektedir.

4.3.6 HLA Temel Veri Türleri

Şekil 4-6’de, temel veri türlerini tanımlamak ve diğer veri türleri ile ilişkilerini oluşturmak için hazırlanmış olan ve EMF Ecore modeline esas teşkil eden şema görülebilir.



Şekil 4-6 Meta model - HLA Veri Türleri

4.3.7 HLA Biçim İmi

HLA biçim imi (tag) elemanı modellenmemiştir. Fakat HLAM76 Ecore modelinden FOM oluşturulurken bu HLA elemanı kabul edilen ilk değerlerle FOM'da yer alır.

4.3.8 HLA Zamanı

HLA zamanı elemanı modellenmemiştir. Fakat HLAM76 Ecore modelinden FOM oluşturulurken bu HLA elemanı kabul edilen ilk değerlerle FOM'da yer alır.

4.3.9 HLA Notu

HLA notu elemanı modellenmemiştir. Fakat HLAM76 Ecore modelinden FOM oluşturulurken bu HLA elemanı kabul edilen ilk değerlerle FOM'da yer alır.

4.3.10 HLA Federe

Bu tez çalışması kapsamında federe modelleme ve federe davranışı modelleme ele alınmamıştır. Bu çalışmaya temel teşkil eden meta model federeyi, federe davranışını, federenin nesne ve etkileşim sınıfları ile ilişkini de modelleyecek şekilde geliştirilirse, federe modellenebilir, SOM üretilebilir.

4.3.11 HLA Zaman Uyumlama Noktası

Zaman uyumlama noktaları da HLA federelerinin modellenmesinde ele alınmaktadır. Bu açıdan bu çalışmaya dahil edilmemişlerdir.

4.4 Araç Setinin Alt Bileşenleri

HLAM76B Modelleme ve Benzetim Geliştirme Aracı, Eclipse platformuna eklenti olarak Eclipse PDE ve Java programa dili kullanılarak geliştirilmiştir. Hlam76b model, Hlam76b edit, Hlam76b editör ve Hlam76b grafiksel editör olmak üzere dört tane Eclipse eklentisinin birarada çalışmasından oluşur. Bu eklentiler aşağıda anlatılmaktadır.

4.4.1 HLAM76B Model Eclipse Eklentisi

Eclipse IDE'si kullanılarak hlam76b adıyla oluşturulan boş Eclipse EMF projesinin içinde bir EMF modeli oluşturulur. Bu EMF projesine hlam76b.mdl adlı Rational Rose class modeli eklenir. Bu işlem yapılırken alınan hatalar, hlam76b.mdl Rational Rose modelinde değişiklikler yapılarak giderilir. Sonuçta, Rational Rose modeli EMF'in kabul edeceği seviyeye getirilir ve hlam76b adlı Eclipse plug-in projesinin altında iki dosya oluşur:

- hlam76b.genmodel (Adı proje adı ile aynı olacak şekilde verilir.)
- halm76b.ecore (Adını Rational Rose modelinden alır.)

hlm76b.genmodel kullanılarak başlangıç için EMF Model kodu üretilir. Bu model kodları kullanıcının grafiksel arayüz üzerinden girdiği tüm verileri tutan Java nesnelere erişimi sağlayan diğer yardımcı sınıfları içerir.

hlm76b projesinin özelliklerine bakıldığında Java Build Path'te aşağıdaki kütüphaneler ve seçenekler olmalıdır:

JRE System Library (jre 1.4.12)

ECLIPSE_CORE_RUNTIME (eclipse\plugins\org.eclipse.core.runtime_3.2.0.v20060327-1200.jar)

ECLIPSE_CORE_RESOURCES (eclipse\plugins\org.eclipse.core.resources_3.2.0.v20060328.jar)

EMF_COMMON (eclipse\plugins\org.eclipse.emf.common_2.1.0.jar)

EMF_ECORE (eclipse\plugins\org.eclipse.emf.ecore_2.1.0.jar)

Java Compiler --> JDK Compliance

Compiler compliance level 1.4

Use default compliance settings

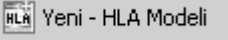
Generated class file compatibility 1.2

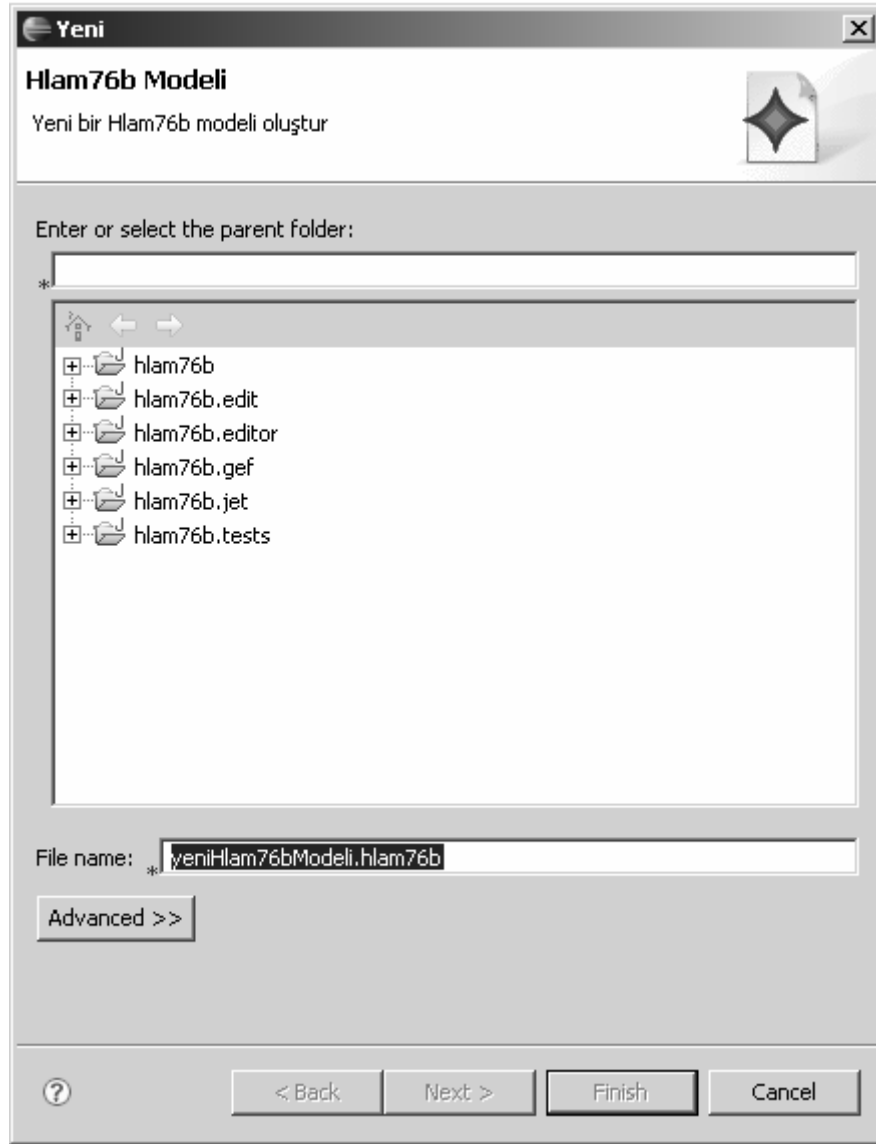
Source compatibility 1.3

4.4.2 HLAM76B Edit Eclipse Eklentisi

Hlam76b Edit Eclipse Eklentisi de Hlam76b EMF modelinden üretilir. Hlam76b Edit Eclipse Eklentisi, oluşturulan Hlam76b Ecore model nesnelere programatik olarak üretmeyi, model nesnelere programatik olarak erişmeyi ve bunların verilerini almayı, ilişkileri bilgisini ve ilişkileri üzerinden ilişkili olduğu nesnelere ulaşmayı sağlar. ItemProvider sınıfları Eclipse PropertySheet'te HLA nesnelere özelliklerini gösterilmek üzere hazırlarlar.

4.4.3 HLAM76B Editör Eclipse Eklentisi

Hlam76b Editör Eclipse Eklentisi, HLAM76B MBGA modelleme aracında yeni bir HLA modeli üretmek için bir model üretme sihirbazını sunar. Bu modelleme sihirbazı 'HLA Modelleme' menüsü üzerindeki  seçilerek açılır. Şekil 4-7'deki gibi bir kullanıcı arayüzü vardır.



Őekil 4-7 H1AM76B HLA Model Sihirbazı

4.4.4 H1AM76B Grafiksel Editör Eclipse Eklentisi

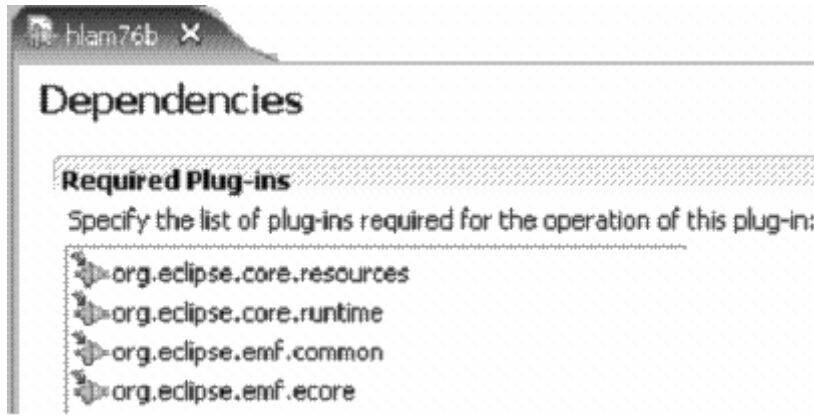
H1am76b GEF Grafiksel Editör Eclipse Eklentisi, kullanıcının ekran üzerinde gorsel olarak nesneleri seřmesini, tařmasını ve model iřine yerleřtirmesini saęlayan grafik Java nesnelelerini iřerir.

Ayrıca bu grafiksel HLA modelleme nesnelерinin özelliklerini H1AM76B Ecore modelinden alarak grafiksel modelde gosterilmesini, grafiksel modelden deęiřtirilmesini saęlayan sınıflar da bu Eclipse eklentisi iřinde yer alır.

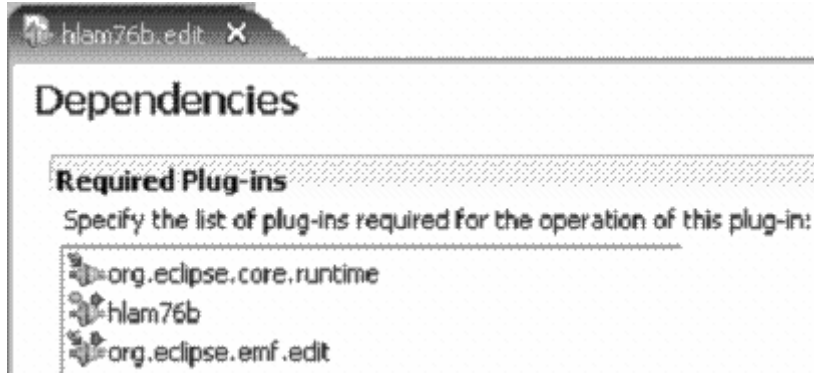
Kullanıcının 'HLA Modelleme ' menüsü üzerinden yaptığı işlemler de bu eklenti içindeki sınıflar tarafından kontrol edilir.

4.4.5 HLAM76B MBGA Alt Bileşenlerinin Etkileşimi

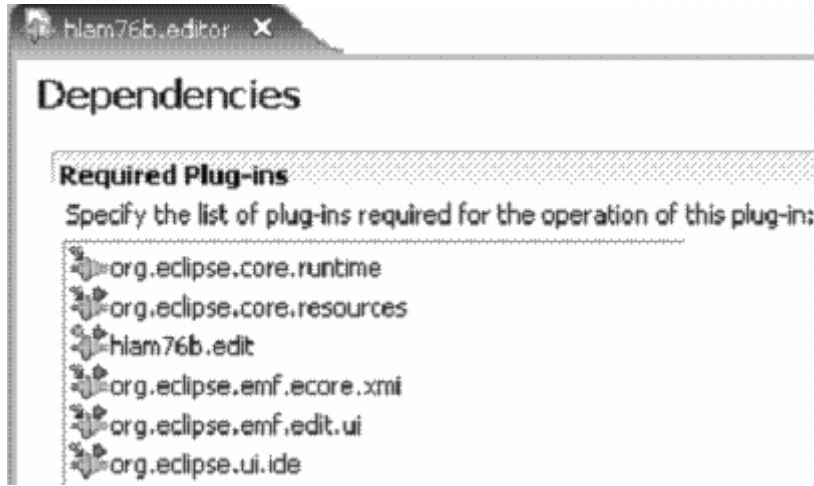
Bu tez çalışması kapsamında geliştirilmiş olan HLA için modelleme ve benzetim geliştirme aracını oluşturan dört adet Eclipse eklentileri birbirlerine ve başka diğer Eclipse eklentilerine ihtiyaç duyarlar. Şekil 4-8, Şekil 4-9, Şekil 4-10 ve Şekil 4-11'de ihtiyaç duyulan tüm Eclipse eklentileri görülmektedir.



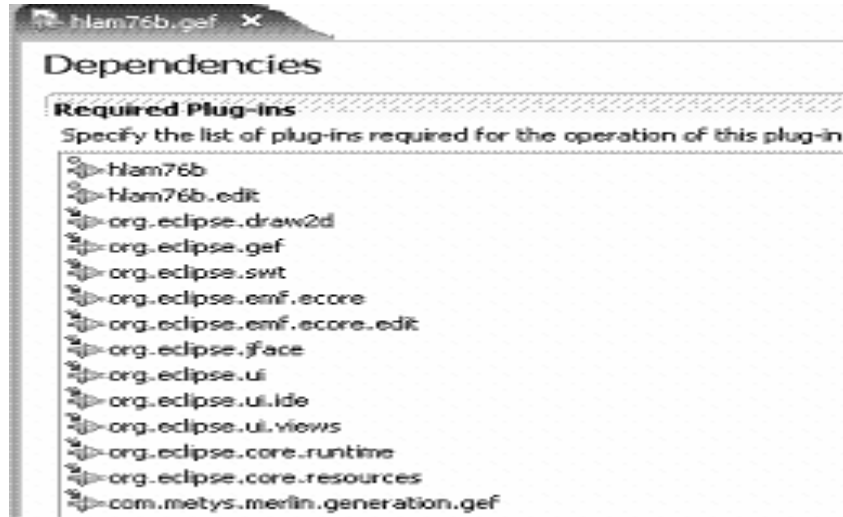
Şekil 4-8 HLAM76B Model - Eklenti Bağımlılıkları



Şekil 4-9 HLAM76B Edit - Eklenti Bağımlılıkları



Şekil 4-10 HLAM76B Editör - Eklenti Bağımlılıkları



Şekil 4-11 HLAM76B GEF - Eklenti Bağımlılıkları

4.5 Araç Setinin Kurulumu

Eclipse platformu ve Eclipse IDE'sinin avantajlarında biri de kolay taşınabilir ve kurulabilir olmasıdır. Modelle aracı Eclipse 3.2 ile ve Java 1.4.2.06 ile geliştirilmiştir.

Geliştirilen Eclipse eklentilerinin JAR dosyaları halinde dışarı aktarımı ve bu JAR halindeki eklentilerin başka bir bilgisayarda kurulu Eclipse ortamının Eclipse/plugins dizinine kopyalanması ve açılması ile aracın kurulumu yapılmış olur.

Aşağıdaki araçların Eclipse eklentileri kurulumundan önce ev sahibi makinada kurulu olması gerekmektedir:

- *Java Software Development Kit (1.4.2.06+)*
- Eclipse 3.2
- EMF eklentileri
- GEF eklentileri
- Merlin eklentileri

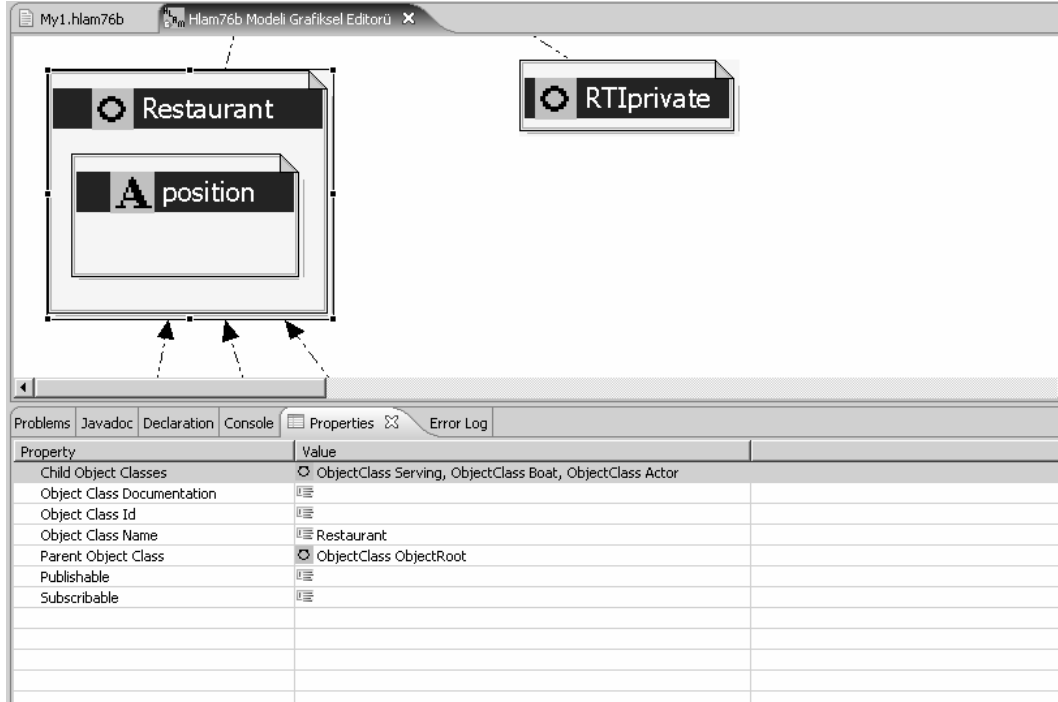
4.6 Araç Setinin Kullanımı

4.6.1 HLA Nesne Modelinin Oluşturulması

HLA nesne modeli, temel olarak, federelerin birbirleriyle uyumlu olarak çalışabilmeleri için kullanacakları nesne ve etkileşim sınıflarıyla modeli tamamlayan yardımcı yapılardan (veri türleri ve boyutlar) oluşur. Bu kesimde sırasıyla nesne sınıflarının, etkileşim sınıflarının, karmaşık veritürlerinin, numaralandırılmış veritürlerinin ve boyutların geliştirilen modelleme aracı kullanarak modellenmesi anlatılacaktır. Model elemanları araç çubuğu üzerindeki ilgili simge yardımıyla modele eklenebilir. Modele eklenen elemanın üzerine sol tıklamayla Eclipse Properties View'inde özelliklerin adı ve değerleri görülebilir. Bura görülen değerler ilk değerleridir ve kullanıcı tarafından düzenlenebilirler.

4.6.2 Nesne Sınıflarının Tanımlanması

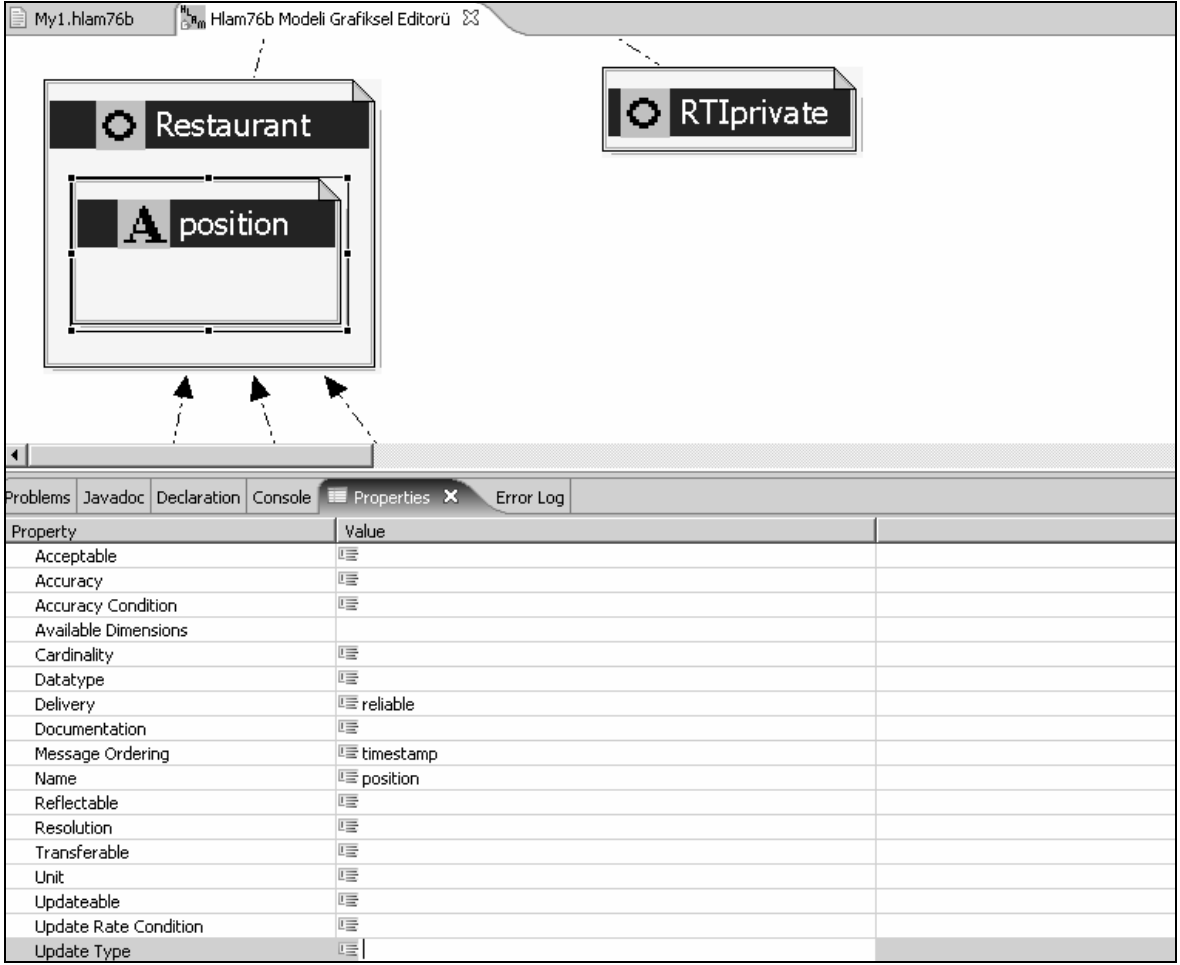
HLA nesne sınıfları, benzetimi yapılacak fiziksel nesnelere tanımlanmakta kullanılır. HLA nesne sınıfları yapı olarak nesneye yönelik sistemlerdeki sınıflara benzerler. Nitelikleri vardır, fakat normal sınıfların aksine yöntemleri olmaz. Tüm sınıfların atası "*HLAObjectRoot*" sınıfıdır. Nesne sınıfları arasında çoklu kalıtıma izin verilmez. Nesne sınıfları, yayımlanabilir ve/veya üye olunabilir olabilirler. Bir sınıfın "yayımlanabilir" olması demek, bir federenin bu sınıfın olgularını yayımlayabileceği manasına gelir. Bir sınıfın "üye olunabilir" olması ise, bir federenin bu sınıfın niteliklerine üye olup değişimleri gözleyip buna göre tepki verebileceği anlamına gelir. Nesne sınıflarının özelliklerini düzenlemek için tasarlanan kullanıcı arayüzü Şekil 4-12'deki gibidir.



Şekil 4-12 HLA Nesne Sınıfı Genel Özellikleri

4.6.3 Nesne Sınıfı Niteliklerinin Tanımlanması

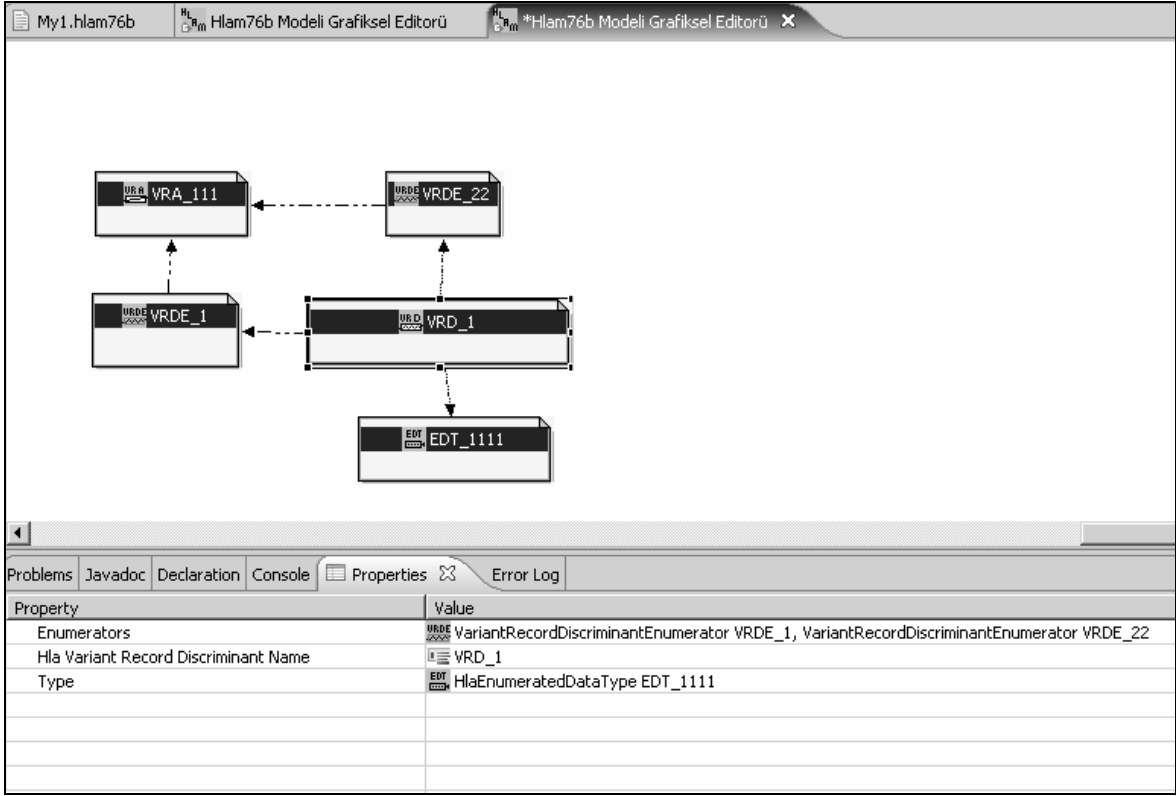
Nitelikler, nesne sınıfı olgularının durumunu belirler. Niteliklerin değerleri zamanla değişebilir. HLA olgu nitelikleri RTI aracılığıyla güncellenir ve federasyondaki diğer ilgilene federelere bildirilir. FOM'da, federasyondaki tüm olgu nitelikleri, SOM'da federenin yayımladığı veya üye olduğu olgu niteliklerinin tanımları bulunur. Niteliklerin çeşitli özellikleri bulunur. Modelleme aracı bu özelliklerin güncellenebilmesi için Şekil 4-13'deki gibi bir arayüz sunar.



Şekil 4-13 HLA Nesne Niteliği Genel Özellikleri

4.6.4 Veri Türü (Datatype)

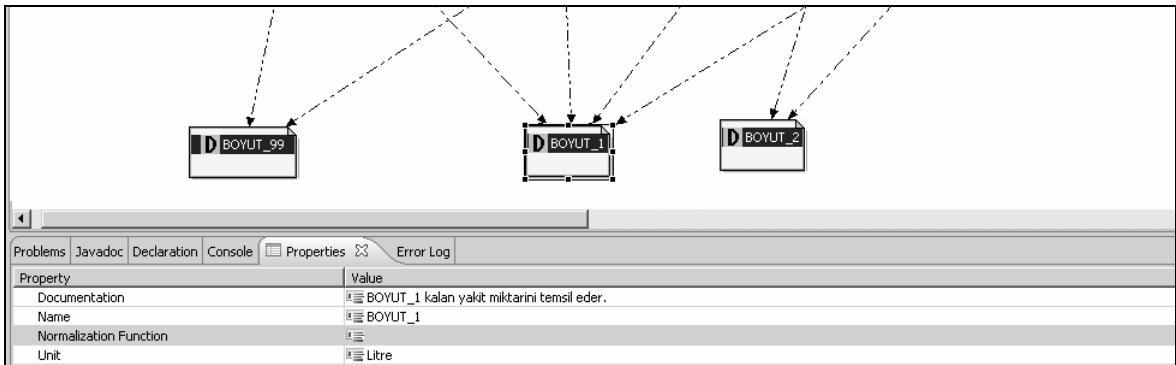
Nitelikler, ilkel (basic) veri türlerinden birinde olabileceği gibi, kullanıcı tanımlı karmaşık veya numaralandırılmış bir veri türünde de olabilirler. Modelleme aracı hem ilkel veri türlerini ve temel (basic) veri türlerini hem de daha karmaşık veri türlerini modelleyebilir. Modelleme aracı veri türlerini düzenleme için Şekil 4-14'deki gibi bir arayüz sunar.



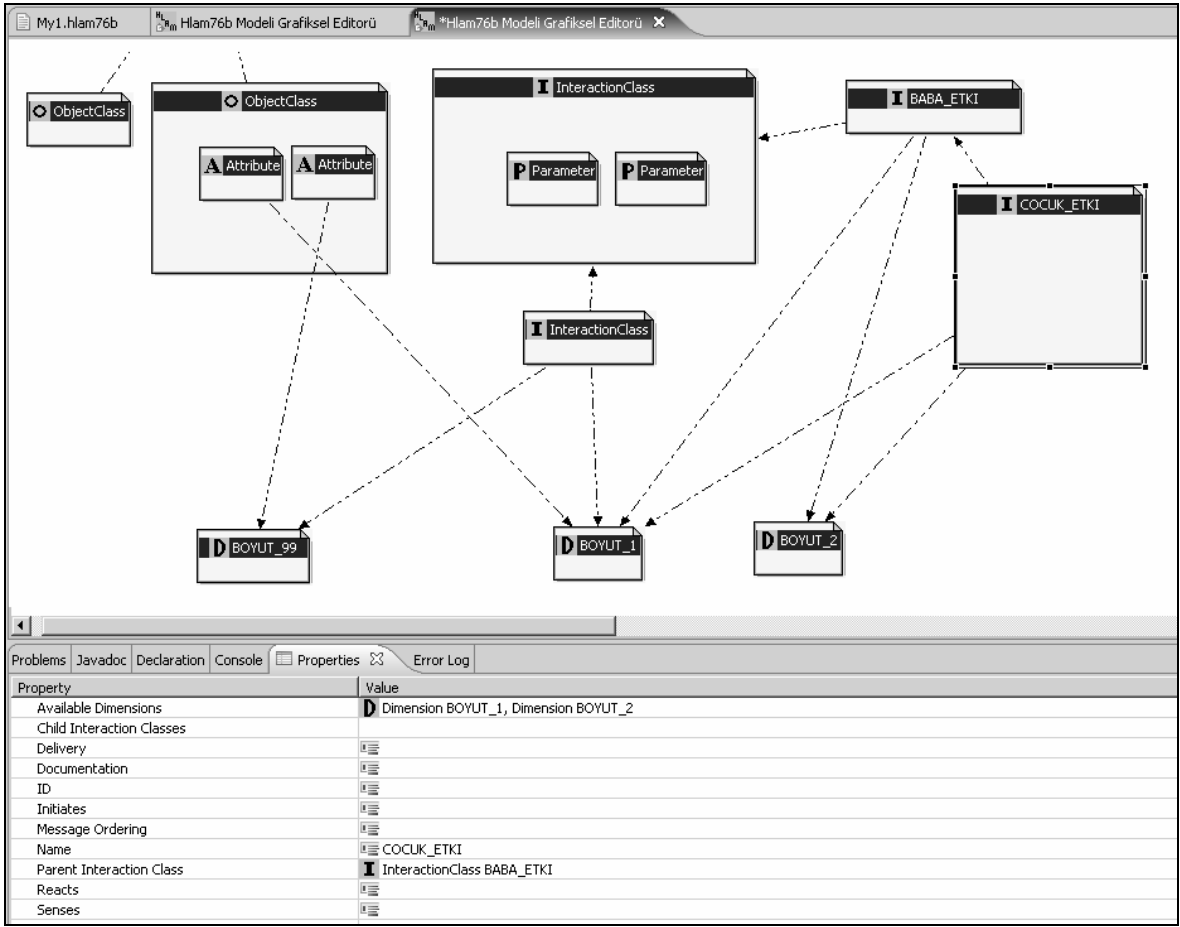
Şekil 4-14 HLA Veri Türleri Özelliklerini Düzenleme

4.6.5 Boyut (*Dimension*):

İlgili federe veya federasyon, Veri Dağıtım Yönetimi (*Data Distribution Management*) servislerini kullanıyorsa, nesne sınıfı nitelikleri için ve etkileşim sınıfı parametreleri için tanımlı boyutlardan biri seçilebilir. Şekil 4-15'te bir boyutun özelliklerini ayarlama ekranı görülmektedir. Şekil 4-16'de ise bu boyutun etkileşim sınıfı tarafından kullanımı görülmektedir.



Şekil 4-15 HLA Boyut Özelliklerini Düzenleme



Şekil 4-16 HLA Boyut Değeri Ayarları

4.6.6 Cardinality

HLA nesne niteliği (Bkz. Şekil 4-13) dizi türünde bir nitelikse *cardinality* olarak dizinin boyu verilmelidir. Sınırsız diziler için 1+ değeri verilmelidir.

4.6.7 Birim (Unit)

HLA nesne niteliği (Bkz. Şekil 4-13) değerlerinin birimi varsa, bu alanda belirtilir.

4.6.8 Duyarlılık (Resolution)

HLA nesne niteliği (Bkz. Şekil 4-13) değerlerini ayırt etmek için gerekli duyarlılık derecesi bu alanda belirtilir.

4.6.9 Doğruluk (Accuracy)

HLA nesne niteliği (Bkz. Şekil 4-13) değerinin beklenen değerden en fazla ne kadar sapabileceği bu alanda belirtilir.

4.6.10 Doğruluk Koşulu (*Accuracy condition*)

HLA nesne niteliği (Bkz. Şekil 4-13) doğruluğunun sağlanması için gerekli koşullar bu alanda belirtilir.

4.6.11 Günleme Türü

- **Static:** Niteliğin değeri statiktir. Federe başlangıçta ve gerektiği zaman değerini günler
- **Periodic:** Federe niteliği düzenli aralıklarla günler
- **Conditional:** Federe niteliği birtakım koşullar oluştuğunda günler
- **NA:** Federe hiçbir zaman niteliği günlemez.

Şekil 4-13'de gösterilen Nitelik Genel Özellikleri Ekranından bu değer girilebilir.

4.6.12 Günlenebilir (*Updateable*) / Yansıtılabilir (*Reflectable*) Olma

Günlenebilir niteliklerin değerleri federasyonun çalışması esnasında değiştirilebilir. Eğer nitelik yansıtılabilir özellikteyse, ilgili federeler niteliğin değerini günler. Şekil 4-13'de gösterilen Nitelik Genel Özellikleri Ekranından bu değer girilebilir.

4.6.13 Devredilebilir (*Transferable*) / Kabuledilebilir(*Acceptable*) Olma

Devredilebilir niteliklerin sahiplikleri federasyon çalışması esnasında devredilebilir. Eğer nitelik kabuledilebilir özellikteyse, başka bir federe niteliğin sahipliğini alabilir. Şekil 4-13'de gösterilen Nitelik Genel Özellikleri Ekranından bu değer girilebilir.

4.6.14 Teslim Türü (*Delivery type*)

Verilerin federeler arasında nasıl taşınacağını belirler. Tüm RTI'lar *HLA reliable* ve *HLA bestEffort* olmak üzere standart iki taşıma şeklini desteklemelidir. Değişik RTI'lar tarafından değişik taşıma yöntemleri tanımlanabilir. Şekil 4-13'de gösterilen Nitelik Genel Özellikleri Ekranından bu değer girilebilir.

4.6.15 İleti Sıralama Türü (*Message ordering type*)

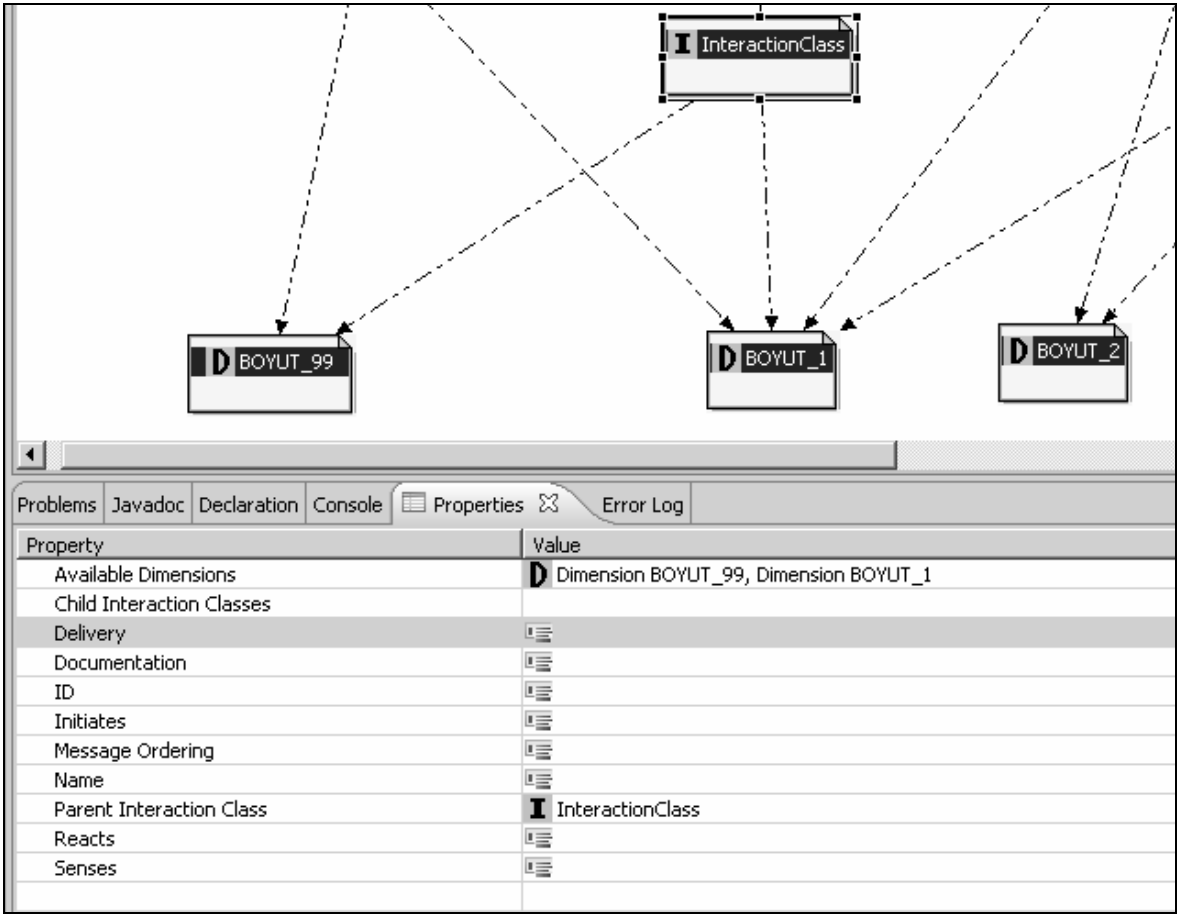
Olguların ilgili federeye ulaştırılma sırasını belirler.

- **Receive:** Sınıf niteliklerinin olguları federeye rastgele bir sırada ulaştırılırlar.
- **TimeStamp:** Sınıf niteliklerinin olguları federeye kendilerine atanan zaman damgalarına göre sıralanmış olarak ulaştırılırlar.

Şekil 4-13'de gösterilen Nitelik Genel Özellikleri Ekranından bu değer girilebilir.

4.6.16 Etkileşim Sınıflarının Tanımlanması

Etkileşim, bir federenin gerçekleştirdiği, başka bir federeyi etkileyebilecek dış eylemdir. Etkileşimler işlev olarak nesneye yönelik sistemlerdeki yöntemlere benzerler. Etkileşimler HLA nesne modellerinde nesne sınıfları gibi sınıf hiyerarşisinde tanımlanmalıdırlar. "*HLAinteractionRoot*" etkileşim sınıfı FOM ve SOM'daki tüm diğer etkileşim sınıflarının atasıdır. Etkileşim sınıflarının belli sayıda parametresi bulunur. Etkileşim sınıflarının özelliklerinin görüntülenmesi Şekil 4-17'da görülebilir [2].



Şekil 4-17 HLA Etkileşim Sınıfı Genel Özellikleri

4.6.16.1 Initiates

Etkileşim sınıfının bir federe tarafından yayımlanabileceğini gösterir.

4.6.16.2 Senses

Etkileşim sınıfının bir federe tarafından üye olunabileceğini gösterir. Federe etkileşimi alsa da tepki vermez, ilgili nesnelere üzerinde değişiklik yapmaz.

4.6.16.3 Reacts

Etkileşim sınıfının bir federe tarafından üye olunabileceğini gösterir. Federe etkileşime tepki verir, ilgili nesnelere üzerinde gerekli değişiklikleri yapar.

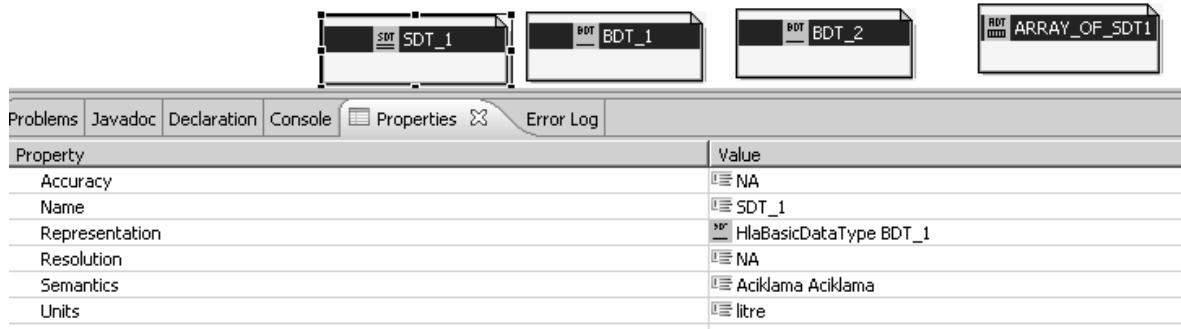
Etkileşim sınıfı, önceden tanımlanmış yönlendirme alanlarıyla ilişkilendirilebilir, etkileşim sınıfının belgelenmesi yapılabilir. Etkileşim sınıfının Delivery Type,

Message Ordering gibi FOM kütüğü oluşturulurken kullanılacak özellikleri ayarlanabilir. Bu özellikler nesne sınıfı niteliklerindekiyle aynı anlamı taşırlar [2].

4.6.17 Karmaşık Veri Türü ve Alanlarının Tanımlanması

Karmaşık veri türlerinin (Complex Data Types) çeşitli sayıda alanları olur. Bunlar ArrayDataType, SimpleDataType, EnumeratedDataType olabilirler (Şekil 4-19). Karmaşık veri türü alanları için nesne sınıfı niteliklerinde olduğu gibi veri türü, birim, duyarlık bilgileri tutulur (Bkz.

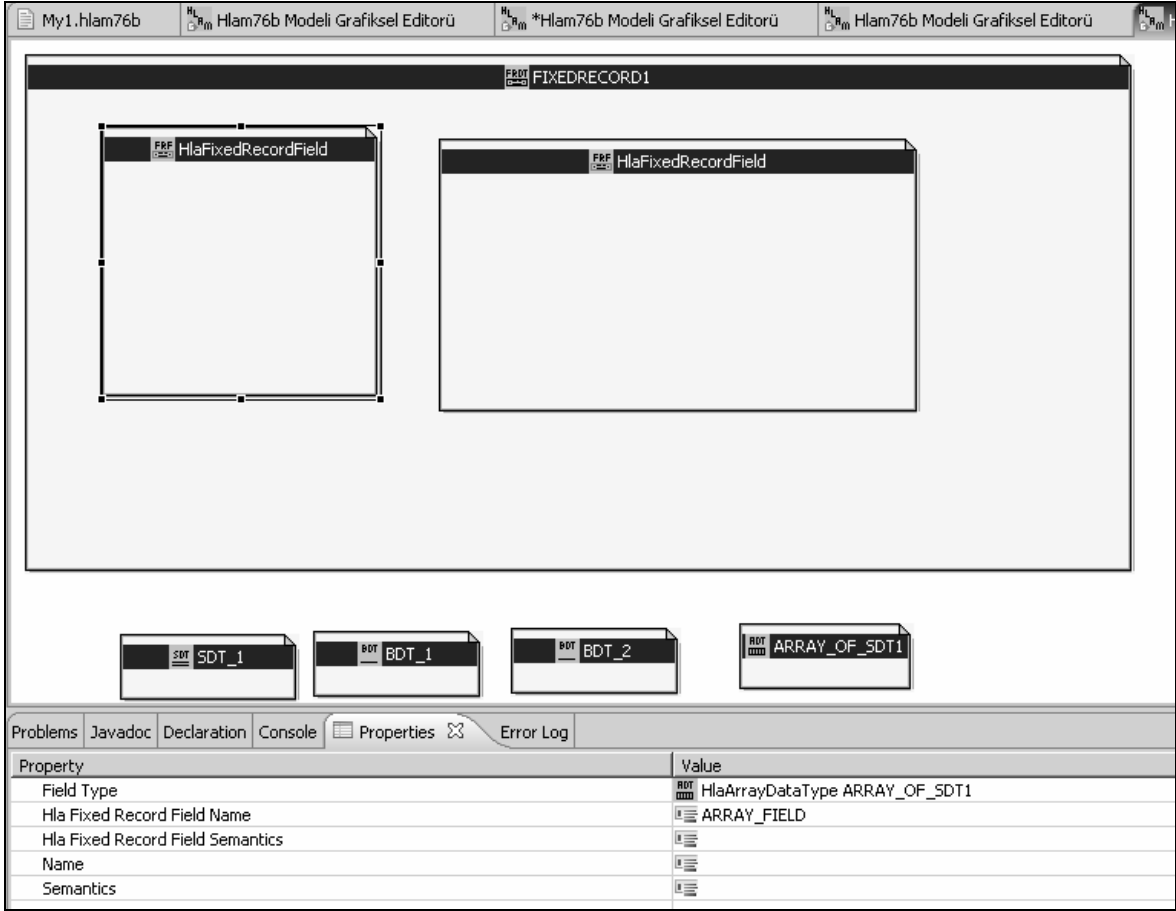
Şekil 4-18).



The screenshot shows a software interface with four data type boxes at the top: SDT_1, BDT_1, BDT_2, and ARRAY_OF_SDT1. Below them is a Properties window for SDT_1. The Properties window has tabs for Problems, Javadoc, Declaration, Console, Properties, and Error Log. The Properties tab is active, showing a table with the following data:

Property	Value
Accuracy	NA
Name	SDT_1
Representation	HlaBasicDataType BDT_1
Resolution	NA
Semantics	Aciklama Aciklama
Units	litre

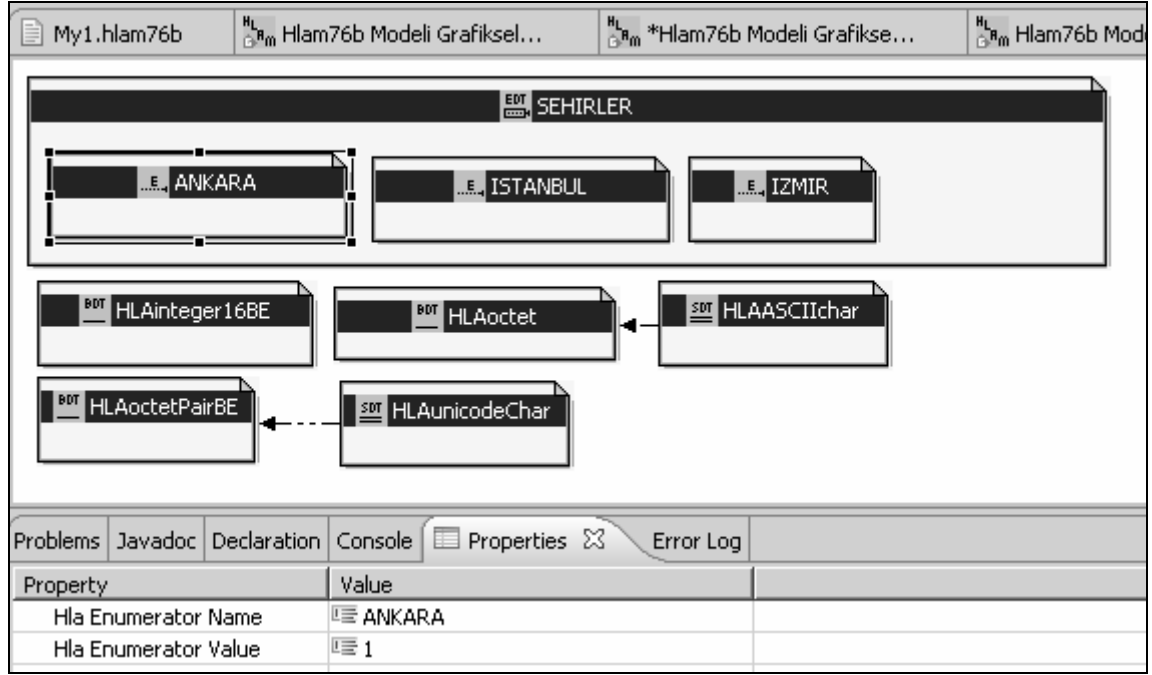
Şekil 4-18 HLA Karmaşık Veri Türü Alanı, Birim, Duyarlık Bilgileri



Şekil 4-19 HLA Karmaşık Veri Türü Genel Özellikleri

4.6.18 Numaralandırılmış Veri Türü ve Numaralandırıcılarının Tanımlanması

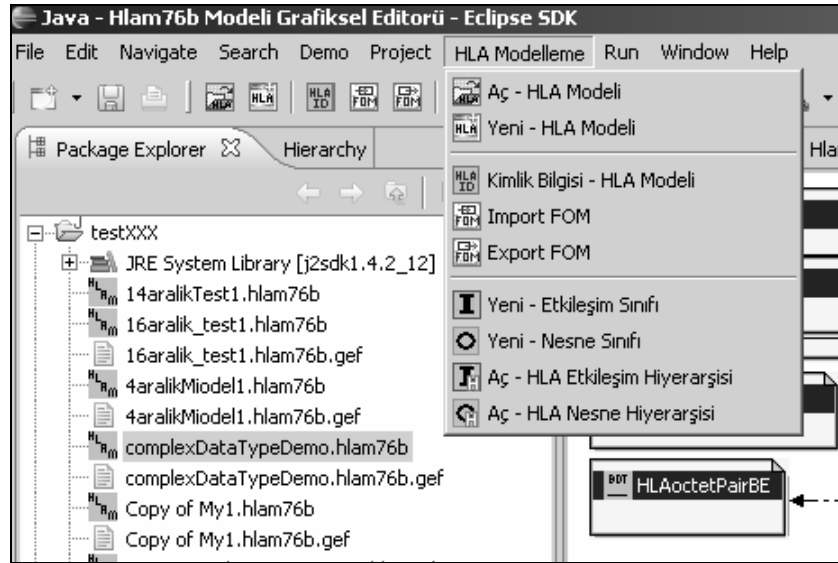
Sınırlı bir kümeden seçilmesi gereken veri türleri için HLA belirtiminde numaralandırılmış veri türü (Enumerated Data Type) tanımlanmıştır. Numaralandırılmış veri türünün özellikleri Şekil 4-20'deki ekrandan düzenlenebilir.



Şekil 4-20 HLA Numaralandırılmış Veri Türü Genel Özellikleri

4.6.19 Model ve Geliştirici Kimlik Bilgilerinin Tanımlanması

Kimlik bilgilerinin tanımlanması için bir arayüz geliştirilmiştir. Bu arayüze HLA Modelleme menüsünden erişilebilir (Bkz. Şekil 4-21 ve Şekil 4-22).


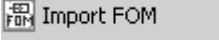


Şekil 4-21 Model Kimlik Bilgilerine Erişim İçin Menü Yapısı

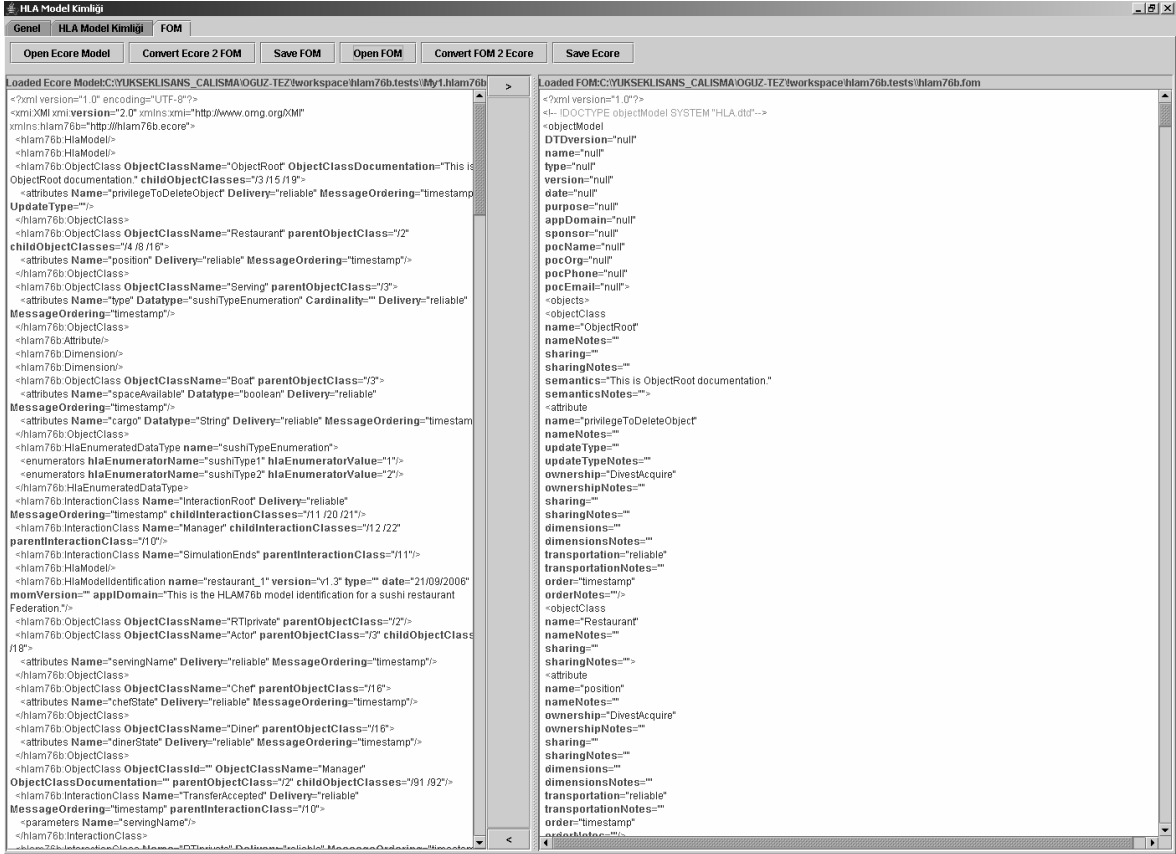
HLA Model Kimliği	
Genel HLA Model Kimliği FOM	
<input type="button" value="Değiştir"/> <input type="button" value="Kaydet"/> <input type="button" value="Vazgeç"/>	
Name	SushiRestaurant
Type	FOM
Version	1516.2
Modification Date	22.12.2006
Purpose	MS. Thesis Demo
Application Domain	Example
Sponsor	Hacettepe Üniversitesi
POC	Oğuz BAKTIR
POC Organization	
POC Telephone	
POC E-mail	abc@abc.de
References	
Other	

Şekil 4-22 Model Kimlik Bilgileri Arayüzü

4.6.20 HLA Modelinden FOM Kütüğünün Oluşturulması

HLA Nesne modeli, geliştirilen araç kullanılarak görsel olarak oluşturulduktan sonra, HLA Modelleme menüsünden  seçilerek FOM kütüğü oluşturulabilir. Yine HLA Modelleme menüsünden  seçilerek daha önce dışarı aktarılmış olan ya da başka bir uygulama tarafından üretilmiş olan bir FOM, içeri aktarılarak uygulamanın dahili kullandığı Hlam76b Ecore modeline çevrilebilir. Böylece içeri aktarılan HLA FOM'da düzenleme yapılarak farklı bir FOM üretilebilir (Bkz. Şekil 4-23).

Uygulama ile, HLA için federasyon geneli nesne modelini içeren FOM kütüğü üretilebilmektedir. Fakat tek bir federasyonun kullandığı nesne ve etkileşim sınıflarını içeren SOM'un üretilmesi, federe ve federe davranışı modelleme ile beraber ileriki bir çalışmaya bırakılmıştır.



Şekil 4-23 FOM Dışarı/İçeri Aktarım (Export/Import) Arayüzü

4.6.21 Federe Modelinin Oluşturulması

Federe modelinde federenin özellikleri ve davranış modeli bilgileri bulunur. Federe modellenmesi bu tez kapsamında incelenmemiştir. Federenin ve davranışının modellenmesini sağlamak bu tez çalışmasının meta modelinin genişletilmesiyle mümkündür.

5 SONUÇLAR VE ÖNERİLER

Bu tez çalışmasıyla ortaya konan modelleme aracı, incelenen diğer araçlardan farklıdır. Bu farkın başlıca nedeni, esnek ve açık bir platform olan Eclipse platformu üzerinde MDA ile geliştirme yapılmış olmasıdır. Dolayısıyla ticari olmayan ve açık bir platform üzerinde geliştirilmiş, genişletmelere ve geliştirmelere hazır bir araç ortaya çıkmıştır. Geliştirilen görsel modelleme aracının diğer bir farkı ise, incelenen araçlarının bazılarının [2] esas aldığı HLA v1.3'e göre değil; daha yeni bir belirtim olan IEEE 1516.2 OMT'ye göre geliştirilmiş olmasıdır.

Eclipse platformuna eklenti olarak geliştirilmiş olan uygulama; başta HLA için federe, federe davranışı, federasyon nesne modeli, benzetim nesne modeli ve federe kodu üretecek şekilde düşünülmüştür. Fakat HLA alan bilgisinin edinilmesi ve HLA için modellenecek nesnelerin bilgisini tutan meta modelin tasarlanmasının fazla zaman alması, gerçekleştirimde kullanılan altyapıların öğrenilmesi takvimi aksatmıştır. Bu yüzden meta model geliştirilirken 76. yinelemeden sonra meta model sabitlenmiş ve gerçekleştirimin diğer safhalarına geçilmiştir.

Ortaya konan ürün HLAM76B MBMA (Modelleme ve Benzetim Geliştirme Aracı) olarak adlandırılmıştır. HLAM76B MBMA, HLA için nesne modeli geliştirmeyi mümkün kılmakta, geliştirilen nesne modellerinin saklanarak daha sonra tekrar düzenlenmesine de imkan sağlamaktadır. Ayrıca istenirse bu araçla geliştirilen nesne modeli IEEE 1516.2 OMT'ye uygun olarak federasyon nesne modeline (FOM) dönüştürülebilmektedir. Bu dönüşümün tersi, yani mevcut bir FOM'un içeri aktarımı da geliştirilen modele göre mümkündür. Bu işlem için FOM'u ayrıştırmak ve gerçekleştirimde kullanılan modele yüklemek yeterlidir. FOM'dan yüklenmiş model, MBGA'nın dahili veri biçimi olan Hlam76b Ecore modelinde tutulur ve XML'in bir türevi olan XMI'ya çevrilerek dosya sistemine muhafaza edilmek üzere aktarılabilir.

Tez kapsamında geliştirilen modelleme aracına temel teşkil eden meta model daha da kapsamlı hale getirilip, tezin gerçekleştirimi esnasında kullanılan diğer altyapılar aynen muhafaza edilerek modelleme aracı daha da geliştirilebilir. Federe ve federe davranışını modelleyecek; ayrıca derlenebilir federe kaynak kodu üretecek şekilde değişiklikler kolayca yapılabilir.

Bu tezin gerekleřtirmesi esnasında Eclipse platformu zerinde kullanılan EMF, GEF, Merlin Generator, JET gibi altyapıların kullanılması, gerekleřtirmesi MDA iine oturtmuř ve yapısal hale getirmiř; fakat gerekleřtirmesi kolaylařtırmamıřtır. Tezin gerekleřtirmesinde kullanılan altyapıların henz taze olup, srekli deęiřim halinde olması yznden, bu altyapıların kullanımı ile ilgili kaynaklar kolay kullanımlı deęildir. Ayrıca kullanılan altyapıların versiyon uyulařmazlıkları ve mevcut bazı yanlıřları nedeniyle altyapıların birbiriyle uyumlu alıřmasında zaman zaman sorunlar yařanmaktadır.

Geliřtirilen modelleme aracı tamamen aık kaynak kodlu Eclipse platformu zerinde eklenti olarak geliřtirilmiřtir. Dolayısıyla Eclipse platformunun getirdięi tm imkanlar kullanılabilir. Eclipse platformunda Eclipse IDE'sine eklenti olarak alıřan ara istenirse Eclipse RCP (Rich Client Platform), SWT (Standart Widget Toolkit) gibi teknolojiler kullanılarak baęımsız alıřan bir uygulama olarak yeniden gerekleřtirilebilir. Dolayısıyla aracın Eclipse platformuna bir eklenti olarak geliřtirilmiř olması herhangi bir dezavantaj oluřturmamaktadır.

Bu tez alıřmasının daha ileri dzeyde geliřtirilmesiyle, HLA iin federe modelleme ve federe iin kod retme yapabilen, federe davranıřlarını da modelleyebilen daha geliřmiř bir modelleme aracının retilmesi mmkndr. Bunun iin meta modelde (Bkz. Őekil 4-3) geliřtirmeler yapmak iřin zdr. Byle bir aracın geliřtirilerek EPL ile yayınlanması dřnlmektedir.

KAYNAKLAR DİZİNİ

- [1] Calytrix Technologies Pty Ltd., 2006, Calytrix SIMplicity Product Overview, <http://www.calytrix.com/siteContent/SIMplicity/intro.php>
- [2] Çelik, T., 2005, Yüksek Düzeyli Mimari İçin Bir Modelleme Aracı Gerçekleştirimi, Yüksek Lisans Tez Çalışması, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü
- [3] Defense Modeling and Simulation Office (DMSO), 1998, High Level Architecture Interface Specification, Version 1.3
- [4] Defense Modeling and Simulation Office (DMSO), 1999, HLA Federation Development and Execution Process (FEDEP) Model v1.5
- [5] Defense Modeling and Simulation Office (DMSO), 1998, HLA Object Model Template, Version 1.3
- [6] Eclipse Foundation, 2005, The Eclipse Modelling Framework Overview, <http://dev.eclipse.org/viewcvs/indextools.cgi/org.eclipse.emf/doc/org.eclipse.emf.doc/references/overview/EMF.html>
- [7] Eclipse Foundation, 2006, The Graphical Editing Framework Project Overview, <http://www.eclipse.org/gef/overview.html>
- [8] Frankel, D. S., 2003, Model Driven Architecture, Applying MDA to Enterprise Computing, Wiley Publishing, 79-94
- [9] Griffin, C., 2005, Transformations in Eclipse, IBM UK Laboratories, 6p.
- [10] Hongwei, W., Heming, Z., 2006, Collaborative Simulation Environment based on HLA and Web Service, 10th International Conference on Computer Supported Cooperative Work in Design, 1-6
- [11] Jia, J., Heming, Z., Bing, G., Keming, W., Chen, D., 2004, HLA-based Collaborative Simulation Platform for Complex Product Design, The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings, vol 2, 462-466
- [12] Klein, U., Schulze, T., Strassburger, S., 1998, Traffic Simulation Based on the High Level Architecture, Simulation Conference Proceedings - Winter, vol 2, 1095-1103
- [13] Kuhl, F., Weatherly, R., Dahmann, J., 1999, Creating Computer Simulation Systems - An Introduction To The High Level Architecture, Prentice Hall
- [14] Object Management Group (OMG), 2006, Model Driven Architecture – How Systems Will Be Built, <http://www.omg.org/mda/>
- [15] Parr, S., Radeski, A., Whitney, R., 2004, The Application Of Tools Support In HLA, Calytrix Technologies Pty Ltd., 5p.

- [16] Pitch Technologies AB. Sweden, 2006, Visual OMT 1516 Software Features Document, <http://www.pitch.se/visualomt/default.asp>
- [17] Scrudder, R., Lutz, R., Dahmann, J., 1998, Automation of the HLA Federation Development and Execution Process, Fall Simulation Interoperability Workshop
- [18] Tolk, A., 2002, Avoiding another Green Elephant – A Proposal for the Next Generation HLA based on the Model Driven Architecture, Simulation Interoperability Workshop, 12p.

EK – 1 Eclipse Eklentileri Tanımlama Kütükleri

HLAM76B Model Eclipse Eklentisi plugin.properties kütüğü:

```
# <copyright>
# OguzBaktir
# </copyright>
#
# $Id$

# =====
# To code developer:
# Do NOT change the properties between this line and the
# "%% END OF TRANSLATED PROPERTIES %" line.
# Make a new property name, append to the end of the file and change
# the code to use the new property.
# =====

# =====
# %% END OF TRANSLATED PROPERTIES %%
# =====

pluginName = Hlam76b Meta Model PI (for IEEE 1516.2 OMT)
providerName = OguzBaktir

# Hlam76b, IEEE 1516.2 OMT tasla\u011F\u0131na g\u00F6re tasarlanm\u0131\u015F ve
uygulanm\u0131\u015F olan,
# benzetim sistemleri i\u00E7in nesne modeli tasarlamakta kullan\u0131labilecek g\u00F6r\u00F6l
bir
# arac\u0131n kullanaca\u011F\u0131 meta modelidir.
# Bahsedilen g\u00F6r\u00F6l nesne modeli tasarlama arac\u0131 da bu meta modele uygun olarak
# geli\u015Ftirilmi\u015Ftir. (Bak\u0131n\u0131z: Hlam76b.Editor ve Hlam76b.Gef)
Bundle-Vendor.0 = OguzBaktir
Bundle-Name.0 = Hlam76b Model PI
```

HLAM76B Model Eclipse Eklentisi *plugin.xml* kütüğü:

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
<!--
  <copyright>
  </copyright>
  $Id$
-->
<plugin>
  <extension point="org.eclipse.emf.ecore.generated_package">
    <package
      uri = "http://hlam76b.ecore"
      class = "hlam76b.Hlam76bPackage"
      genModel = "" />
    </extension>
  </plugin>
```

HLAM76B Edit Eclipse Eklentisi plugin.properties kütüğü:

```
# <copyright>
# OguzBaktir
# </copyright>
#
# $Id$
# =====
# To code developer:
# Do NOT change the properties between this line and the
# "%% END OF TRANSLATED PROPERTIES %" line.
# Make a new property name, append to the end of the file and change
# the code to use the new property.
# =====
# %% END OF TRANSLATED PROPERTIES %%
# =====
```

```

pluginName = Hlam76b Edit PI
providerName = OguzBaktir

# GORSEL OLARAK MODELLENEN NESNELER UZERINDE GOSTERILEN ISIMLER
_UI_HlaModel_type = HlaModel
_UI_ObjectClass_type = ObjectClass
_UI_Attribute_type = Attribute
_UI_Dimension_type = Dimension
_UI_HlaModelIdentification_type = HlaModelIdentification
_UI_InteractionClass_type = InteractionClass
_UI_Parameter_type = Parameter
_UI_HlaBasicDataType_type = HlaBasicDataType
_UI_HlaSimpleDataType_type = HlaSimpleDataType
_UI_HlaUserDefinedDataType_type = HlaUserDefinedDataType
_UI_HlaEnumeratedDataType_type = HlaEnumeratedDataType
_UI_HlaArrayDataType_type = HlaArrayDataType
_UI_HlaFixedRecordDataType_type = HlaFixedRecordDataType
_UI_HlaVariantRecordDataType_type = HlaVariantRecordDataType
_UI_HlaEnumerator_type = HlaEnumerator
_UI_HlaFixedRecordField_type = HlaFixedRecordField
_UI_HlaVariantRecordDiscriminant_type = HlaVariantRecordDiscriminant
_UI_HlaVariantRecordAlternative_type = HlaVariantRecordAlternative
_UI_VariantRecordDiscriminantEnumerator_type = VariantRecordDiscriminantEnumerator
_UI_Unknown_type = Object

_UI_Unknown_datatype= Value

# PROPERTIES VIEW INDE GOSTERILEN FEATURE ISIMLERI
_UI_HlaModel_objectClasses_feature = Object Classes
_UI_HlaModel_attributes_feature = Attributes
_UI_HlaModel_dimensions_feature = Dimensions
_UI_HlaModel_identification_feature = Identification
_UI_HlaModel_interactionClasses_feature = Interaction Classes
_UI_HlaModel_basicDatatypes_feature = Basic Datatypes
_UI_HlaModel_userDefinedDatatypes_feature = User Defined Datatypes
_UI_ObjectClass_ObjectClassId_feature = Object Class Id
_UI_ObjectClass_ObjectClassName_feature = Object Class Name
_UI_ObjectClass_ObjectClassDocumentation_feature = Object Class Documentation
_UI_ObjectClass_Publishable_feature = Publishable
_UI_ObjectClass_Subscribable_feature = Subscribable
_UI_ObjectClass_attributes_feature = Attributes
_UI_ObjectClass_parentObjectClass_feature = Parent Object Class
_UI_ObjectClass_childObjectClasses_feature = Child Object Classes
_UI_Attribute_Name_feature = Name
_UI_Attribute_Datatype_feature = Datatype
_UI_Attribute_Documentation_feature = Documentation
_UI_Attribute_Cardinality_feature = Cardinality
_UI_Attribute_Unit_feature = Unit
_UI_Attribute_Resolution_feature = Resolution
_UI_Attribute_Accuracy_feature = Accuracy
_UI_Attribute_AccuracyCondition_feature = Accuracy Condition
_UI_Attribute_UpdateRateCondition_feature = Update Rate Condition
_UI_Attribute_Updateable_feature = Updateable
_UI_Attribute_Reflectable_feature = Reflectable
_UI_Attribute_Transferable_feature = Transferable
_UI_Attribute_Acceptable_feature = Acceptable
_UI_Attribute_Delivery_feature = Delivery
_UI_Attribute_MessageOrdering_feature = Message Ordering
_UI_Attribute_UpdateType_feature = Update Type
_UI_Attribute_ownerClass_feature = Owner Class
_UI_Attribute_availableDimensions_feature = Available Dimensions
_UI_Dimension_Name_feature = Name
_UI_Dimension_Documentation_feature = Documentation
_UI_Dimension_Datatype_feature = Datatype
_UI_Dimension_Unit_feature = Unit
_UI_Dimension_NormalizationFunction_feature = Normalization Function
_UI_HlaModelIdentification_pocTelephone_feature = Poc Telephone
_UI_HlaModelIdentification_name_feature = Name
_UI_HlaModelIdentification_version_feature = Version
_UI_HlaModelIdentification_type_feature = Type
_UI_HlaModelIdentification_date_feature = Date
_UI_HlaModelIdentification_momVersion_feature = Mom Version
_UI_HlaModelIdentification_applDomain_feature = Appl Domain
_UI_HlaModelIdentification_purpose_feature = Purpose
_UI_HlaModelIdentification_honorificName_feature = Honorific Name
_UI_HlaModelIdentification_firstName_feature = First Name

```

```

_UI_HlaModelIdentification_secondName_feature = Second Name
_UI_HlaModelIdentification_pocName_feature = Poc Name
_UI_HlaModelIdentification_pocOrganization_feature = Poc Organization
_UI_HlaModelIdentification_pocEmail_feature = Poc Email
_UI_HlaModelIdentification_sponsor_feature = Sponsor
_UI_InteractionClass_ID_feature = ID
_UI_InteractionClass_Name_feature = Name
_UI_InteractionClass_initiates_feature = Initiates
_UI_InteractionClass_senses_feature = Senses
_UI_InteractionClass_reacts_feature = Reacts
_UI_InteractionClass_Documentation_feature = Documentation
_UI_InteractionClass_Delivery_feature = Delivery
_UI_InteractionClass_MessageOrdering_feature = Message Ordering
_UI_InteractionClass_childInteractionClasses_feature = Child Interaction Classes
_UI_InteractionClass_parentInteractionClass_feature = Parent Interaction Class
_UI_InteractionClass_parameters_feature = Parameters
_UI_InteractionClass_availableDimensions_feature = Available Dimensions
_UI_Parameter_Name_feature = Name
_UI_Parameter_Datatype_feature = Datatype
_UI_Parameter_Documentation_feature = Documentation
_UI_Parameter_Cardinality_feature = Cardinality
_UI_Parameter_Unit_feature = Unit
_UI_Parameter_Resolution_feature = Resolution
_UI_Parameter_Accuracy_feature = Accuracy
_UI_Parameter_AccuracyCondition_feature = Accuracy Condition
_UI_HlaBasicDataType_hlaBasicDataTypeName_feature = Hla Basic Data Type Name
_UI_HlaBasicDataType_hlaBasicDataTypeSizeInBits_feature = Hla Basic Data Type Size In Bits
_UI_HlaBasicDataType_hlaBasicDataTypeInterpretation_feature = Hla Basic Data Type
Interpretation
_UI_HlaBasicDataType_hlaBasicDataTypeEndian_feature = Hla Basic Data Type Endian
_UI_HlaBasicDataType_hlaBasicDataTypeEncoding_feature = Hla Basic Data Type Encoding
_UI_HlaSimpleDataType_units_feature = Units
_UI_HlaSimpleDataType_resolution_feature = Resolution
_UI_HlaSimpleDataType_accuracy_feature = Accuracy
_UI_HlaSimpleDataType_representation_feature = Representation
_UI_HlaUserDefinedDataType_name_feature = Name
_UI_HlaUserDefinedDataType_semantics_feature = Semantics
_UI_HlaEnumeratedDataType_enumerators_feature = Enumerators
_UI_HlaEnumeratedDataType_representation_feature = Representation
_UI_HlaArrayType_cardinality_feature = Cardinality
_UI_HlaArrayType_encoding_feature = Encoding
_UI_HlaArrayType_elementType_feature = Element Type
_UI_HlaFixedRecordDataType_encoding_feature = Encoding
_UI_HlaFixedRecordDataType_filelds_feature = Filelds
_UI_HlaVariantRecordDataType_encoding_feature = Encoding
_UI_HlaVariantRecordDataType_discriminant_feature = Discriminant
_UI_HlaEnumerator_hlaEnumeratorName_feature = Hla Enumerator Name
_UI_HlaEnumerator_hlaEnumeratorValue_feature = Hla Enumerator Value
_UI_HlaEnumerator_enumDataType_feature = Enum Data Type
_UI_HlaFixedRecordField_hlaFixedRecordFieldName_feature = Hla Fixed Record Field Name
_UI_HlaFixedRecordField_hlaFixedRecordFieldSemantics_feature = Hla Fixed Record Field
Semantics
_UI_HlaFixedRecordField_fieldType_feature = Field Type
_UI_HlaVariantRecordDiscriminant_hlaVariantRecordDiscriminantName_feature = Hla Variant
Record Discriminant Name
_UI_HlaVariantRecordDiscriminant_type_feature = Type
_UI_HlaVariantRecordDiscriminant_enumerators_feature = Enumerators
_UI_HlaVariantRecordAlternative_hlaVariantRecordAlternativeName_feature = Hla Variant
Record Alternative Name
_UI_HlaVariantRecordAlternative_hlaVariantRecordAlternativeSemantics_feature = Hla Variant
Record Alternative Semantics
_UI_HlaVariantRecordAlternative_type_feature = Type
_UI_VariantRecordDiscriminantEnumerator_alternative_feature = Alternative
_UI_Unknown_feature = Unspecified

```

HLAM76B Edit Eclipse Eklentisi plugin.xml kütüğü:

```

<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
<!--
  <copyright>
  </copyright>

  $Id$

```

```
-->
<plugin>
  <extension point="org.eclipse.emf.edit.itemProviderAdapterFactory">
    <factory
      uri = "http://hlam76b.ecore"
      class = "hlam76b.provider.Hlam76bItemProviderAdapterFactory"
      supportedTypes =
        "org.eclipse.emf.edit.provider.IEditingDomainItemProvider
        org.eclipse.emf.edit.provider.IStructuredItemContentProvider
        org.eclipse.emf.edit.provider.ITreeItemContentProvider
        org.eclipse.emf.edit.provider.IItemLabelProvider
        org.eclipse.emf.edit.provider.IItemPropertySource" />
    </extension>
  </plugin>
```

HLAM76B Editör Eclipse Eklentisi plugin.properties kütüğü:

```
# <copyright>
# </copyright>
#
# $Id$

# =====
# To code developer:
# Do NOT change the properties between this line and the
# "%% END OF TRANSLATED PROPERTIES %" line.
# Make a new property name, append to the end of the file and change
# the code to use the new property.
# =====

# =====
# %% END OF TRANSLATED PROPERTIES %%
# =====

pluginName = Hlam76b Editor PI
providerName = OguzBaktir

_UI_Hlam76bEditor_menu = Hlam76b Hiyerar\u015Fik Edit\u00F6r\u00FC
_UI_CreateChild_menu_item = &Yeni Çocuk
_UI_CreateSibling_menu_item = Y&eni Karde\u015F

_UI_ShowPropertiesView_menu_item = Show &Properties View
_UI_RefreshViewer_menu_item = &Tazele

_UI_SelectionPage_label = Selection
_UI_ParentPage_label = Parent
_UI_ListPage_label = List
_UI_TreePage_label = Tree
_UI_TablePage_label = Table
_UI_TreeWithColumnsPage_label = Tree with Columns
_UI_ObjectColumn_label = Object
_UI_SelfColumn_label = Self

_UI_NoObjectSelected = Hi\u00E7bir\u015Fey se\u00E7ilmedi.
_UI_SingleObjectSelected = Se\u00E7ilen nesne: {0}
_UI_MultiObjectSelected = {0} Nesne Se\u00E7ildi

_UI_OpenEditorError_label = Edit\u00F6r A\u00E7

_UI_Wizard_category = IEEE 1516.2 FOM ve EMF Ecore Yaratma Sihirbazlar\u0131

_UI_Hlam76bModelWizard_label = Hlam76b Modeli
_UI_Hlam76bModelWizard_description = Yeni bir Hlam76b modeli olu\u015Ftur

_UI_Hlam76bEditor_label = Hlam76b Hiyerar\u015Fik Edit\u00F6r\u00FC

_UI_Hlam76bEditorFilenameDefaultBase = yeniHlam76bModeli
_UI_Hlam76bEditorFilenameExtension = hlam76b

_UI_Wizard_label = Yeni

_WARN_FilenameExtension = Dosya uzant\u0131s\u0131 ".{0}" olmal\u0131d\u0131r.

_UI_ModelObject = Modellenecek Hlam76b Model Eleman\u0131 (IEEE 1516.2 FOM \u00FCretmek
i\u00E7in HlaModel se\u00E7in)
_UI_XMLEncoding = XML Encoding
```

```

_UI_XMLEncodingChoices = UTF-8 ASCII UTF-16 UTF-16BE UTF-16LE ISO-8859-1
_UI_Wizard_initial_object_description = Yarat\u0131lacak model nesnesini se\u00E7in

_UI_FileConflict_label = Dosya uyu\u015Fmazl\u0131\u011F\u0131
_WARN_FileConflict = Edit\u00F6r a\u00E7\u0131kken, edit\u00F6r d\u0131\u015F\u0131nda
dosyada bir de\u011Fi\u015Fiklik yap\u0131lm\u0131\u015F ve bu da edit\u00F6r\u00FCn
g\u00F6sterdi\u011Fi i\u00E7erikle \u00E7eli\u015Fiyor. Edit\u00F6rdeki
de\u011Fi\u015Fiklikleri ihmal etmek ister misiniz?

```

HLAM76B Editör Eclipse Eklentisi plugin.xml kütüğü:

```

<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>

<!--
<copyright>
</copyright>

$Id$
-->

<plugin>

  <extension
    point = "org.eclipse.ui.newWizards">
    <category
      id = "org.eclipse.emf.ecore.Wizard.category.ID"
      name="%_UI_Wizard_category">
    </category>
    <wizard
      id = "hlam76b.presentation.Hlam76bModelWizardID"
      name = "%_UI_Hlam76bModelWizard_label"
      class = "hlam76b.presentation.Hlam76bModelWizard"
      category = "org.eclipse.emf.ecore.Wizard.category.ID"
      icon = "icons/full/obj16/Hlam76bModelFile.gif">
      <description>%_UI_Hlam76bModelWizard_description</description>
      <selection class = "org.eclipse.core.resources.IResource" />
    </wizard>
  </extension>

  <extension point = "org.eclipse.ui.editors">
    <editor
      id = "hlam76b.presentation.Hlam76bEditorID"
      name = "%_UI_Hlam76bEditor_label"
      icon = "icons/full/obj16/Hlam76bModelFile.gif"
      extensions = "hlam76b"
      class = "hlam76b.presentation.Hlam76bEditor"
      contributorClass="hlam76b.presentation.Hlam76bActionBarContributor" >
    </editor>
  </extension>
</plugin>

```

HLAM76B Grafiksel Editör Eclipse Eklentisi plugin.properties kütüğü:

```

# <copyright>
# </copyright>
#
# $Id$

# =====
# To code developer:
# Do NOT change the properties between this line and the
# "%% END OF TRANSLATED PROPERTIES %" line.
# Make a new property name, append to the end of the file and change
# the code to use the new property.
# =====

# =====
# %% END OF TRANSLATED PROPERTIES %%
# =====

```



```

pluginName = Hlam76b GEF PI
providerName = OguzBaktir

_UI_Hlam76bEditor_menu = &Hlam76b Grafiksel Edit\u00F6r\u00FC
_UI_CreateChild_menu_item = &Yeni Çocuk
_UI_CreateSibling_menu_item = Y&eni Karde\u015F

_UI_ShowPropertiesView_menu_item = Show &Properties View
_UI_RefreshViewer_menu_item = &Tazele

_UI_SelectionPage_label = Selection
_UI_ParentPage_label = Parent
_UI_ListPage_label = List
_UI_TreePage_label = Tree
_UI_TablePage_label = Table
_UI_TreeWithColumnsPage_label = Tree with Columns
_UI_ObjectColumn_label = Object
_UI_SelfColumn_label = Self

_UI_NoObjectSelected = Hi\u00E7bir\u015Fey se\u00E7ilmedi.
_UI_SingleObjectSelected = Se\u00E7ilen nesne: {0}
_UI_MultiObjectSelected = {0} Nesne Se\u00E7ildi

_UI_OpenEditorError_label = Edit\u00F6r\u00FC A\u00E7

_UI_Wizard_category = IEEE 1516.2 FOM ve EMF Ecore Yaratma Sihirbazlar\u0131

_UI_Hlam76bModelWizard_label = Hlam76b Modeli
_UI_Hlam76bModelWizard_description = Yeni bir Hlam76b modeli olu\u015Ftur

_UI_Hlam76bEditor_label = Hlam76b Modeli Grafiksel Editor\u00FC
_UI_Hlam76bEditorFilenameDefaultBase = yeniHlam76bModeli
_UI_Hlam76bEditorFilenameExtension = hlam76b

_UI_Wizard_label = Yeni

_WARN_FilenameExtension = Dosya uzant\u0131s\u0131 ".{0}" olmal\u0131d\u0131r.

_UI_ModelObject = Modellenek Hlam76b Model Eleman\u0131 (IEEE 1516.2 FOM \u00FCretmek
i\u00E7in HlaModel se\u00E7in)
_UI_XMLEncoding = XML Encoding
_UI_XMLEncodingChoices = UTF-8 ASCII UTF-16 UTF-16BE UTF-16LE ISO-8859-1
_UI_Wizard_initial_object_description = Yarat\u0131lacak model nesnesini se\u00E7in

_UI_FileConflict_label = Dosya uyu\u015Fmazl\u0131\u011F\u0131
_WARN_FileConflict = Edit\u00F6r a\u00E7\u0131kken, edit\u00F6r d\u0131\u015F\u0131nda
dosyada bir de\u011Fi\u015Flik yap\u0131lm\u0131\u015F ve bu da edit\u00F6r\u00FCn
g\u00FCsterdi\u011Fi i\u00E7erikle \u00E7eli\u015Fiyor. Edit\u00F6rdeki
de\u011Fi\u015Flikleri ihmal etmek ister misiniz?

```

HLAM76B Grafiksel Editör Eclipse Eklentisi plugin.xml kütüğü:

```

<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>

<plugin
  name = "%pluginName"
  id = "hlam76b.gef"
  version = "2.0.5"
  provider-name = "%providerName"
  class = "hlam76b.gef.Hlam76bGefPlugin">

  <requires>
    <import plugin="hlam76b"/>
    <import plugin="hlam76b.edit"/>
    <import plugin="org.eclipse.draw2d"/>
    <import plugin="org.eclipse.gef"/>
    <import plugin="org.eclipse.swt"/>
    <import plugin="org.eclipse.emf.ecore"/>
    <import plugin="org.eclipse.emf.ecore.edit"/>
    <import plugin="org.eclipse.jface"/>
    <import plugin="org.eclipse.ui"/>

```

```

<import plugin="org.eclipse.ui.ide"/>
<import plugin="org.eclipse.ui.views"/>
<import plugin="org.eclipse.core.runtime"/>
<import plugin="org.eclipse.core.resources"/>
<import plugin="com.metys.merlin.generation.gef"/>
</requires>

<runtime>
  <library name="hlam76bgef.jar">
    <export name="*" />
  </library>
  <library name="lib/xmleditorkit.jar">
    <export name="*" />
  </library>
</runtime>
<extension point="org.eclipse.ui.editors">

  <editor
    id = "hlam76b.gefeditor.Hlam76bEditorID"
    name = "%_UI_Hlam76bEditor_label"
    icon = "icons/full/obj16/Hlam76bEditor.gif"
    extensions = "hlam76b, hlam76b.gef"
    class = "hlam76b.gefeditor.Hlam76bEditor"
    contributorClass="hlam76b.gefeditor.Hlam76bEditorActionBarContributor"/>
</extension>

<extension
  point="org.eclipse.ui.actionSets">
  <actionSet
    id="hlam76b.gef.actionSet"
    label="HLA Model"
    visible="true">
    <menu
      id="HLAModellemeMenu"
      label="HLA Modelleme">
      <separator name="sampleGroup0"/>
      <separator name="sampleGroup1"/>
      <separator name="sampleGroup2"/>
    </menu>
    <action
      class="hlam76b.gef.actions.RunYeniHLAModeli"
      icon="icons/full/obj16/YeniHLAModeliOlustur_ToolBarIcon.gif"
      id="hlam76b.gef.actions.RunYeniHLAModeli"
      label="Yeni - HLA Modeli"
      menubarPath="HLAModellemeMenu/sampleGroup0"
      style="push"
      toolbarPath="sampleGroup0"
      tooltip="Yeni HLA modeli oluşturun." />
    <action
      class="hlam76b.gef.actions.RunAcHLAModeli"
      icon="icons/full/obj16/HLAModeliniAc_ToolBarIcon.gif"
      id="hlam76b.gef.actions.RunAcHLAModeli"
      label="Aç - HLA Modeli"
      menubarPath="HLAModellemeMenu/sampleGroup0"
      style="push"
      toolbarPath="sampleGroup0"
      tooltip="Kaydedilmiş bir HLA modelini aç." />
    </action>
    <!--action
      class="hlam76b.gef.actions.RunSaveHLAModeli"
      icon="icons/full/obj16/HLAModeliniKaydet_ToolBarIcon.gif"
      id="hlam76b.gef.actions.RunSaveHLAModeli"
      label="Kaydet - HLA Modeli"
      menubarPath="HLAModellemeMenu/sampleGroup0"
      style="push"
      toolbarPath="sampleGroup0"
      tooltip="HLA modelini kaydet." />
    <action
      class="hlam76b.gef.actions.RunCloseHLAModeli"
      icon="icons/full/obj16/HLAModeliniKapat_ToolBarIcon.gif"
      id="hlam76b.gef.actions.RunCloseHLAModeli"
      label="Kapat - HLA Modeli"
      menubarPath="HLAModellemeMenu/sampleGroup0"
      style="push"
      toolbarPath="sampleGroup0"
      tooltip="HLA modelini kapat." /-->
    <action

```

```

class="hlam76b.gef.actions.RunExportFOMfromHLAModel"
icon="icons/full/obj16/HLAModelindenFOMExporEt_ToolBarIcon2.gif"
id="hlam76b.gef.actions.RunExportFOMfromHLAModel"
label="Export FOM"
menubarPath="HLAModellemeMenu/sampleGroup1"
style="push"
toolbarPath="sampleGroup1"
tooltip="HLA modelinden FOM dosyası oluştur."/>
<action
class="hlam76b.gef.actions.RunImportFOMtoHLAModel"
icon="icons/full/obj16/HLAModelineFOMImporEt_ToolBarIcon.gif"
id="hlam76b.gef.actions.RunImportFOMtoHLAModel"
label="Import FOM"
menubarPath="HLAModellemeMenu/sampleGroup1"
style="push"
toolbarPath="sampleGroup1"
tooltip="Dışarıdan alınan FOM dosyasını HLA Modeline çevir."/>
<action
class="hlam76b.gef.actions.RunViewEditHLAModelIdentification"
icon="icons/full/obj16/HLAModeliKimlikBilgisi_ToolBarIcon.gif"
id="hlam76b.gef.actions.RunViewEditHLAModelIdentification"
label="Kimlik Bilgisi - HLA Modeli"
menubarPath="HLAModellemeMenu/sampleGroup1"
style="push"
toolbarPath="sampleGroup1"
tooltip="HLA Model kimliğini göster."/>
<action
class="hlam76b.gef.actions.RunViewEditHLAModelObjectHierarchy"
icon="icons/full/obj16/ObjectClassHiyerarşisiniAc_ToolBarIcon.gif"
id="hlam76b.gef.actions.RunViewEditHLAModelObjectHierarchy"
label="Aç - HLA Nesne Hiyerarşisi"
menubarPath="HLAModellemeMenu/sampleGroup2"
style="push"
toolbarPath="sampleGroup2"
tooltip="HLA Nesne Hiyerarşisini göster."/>
<action
class="hlam76b.gef.actions.RunViewEditHLAModelInteractionHierarchy"
icon="icons/full/obj16/InteractionClassHiyerarşisiniAc_ToolBarIcon.gif"
id="hlam76b.gef.actions.RunViewEditHLAModelInteractionHierarchy"
label="Aç - HLA Etkileşim Hiyerarşisi"
menubarPath="HLAModellemeMenu/sampleGroup2"
style="push"
toolbarPath="sampleGroup2"
tooltip="HLA Etkileşim Hiyerarşisini aç."/>
<action
class="hlam76b.gef.actions.RunAddNewObjectClass"
icon="icons/full/obj16/YeniObjectClassOlustur_ToolBarIcon.gif"
id="hlam76b.gef.actions.RunAddNewObjectClass"
label="Yeni - Nesne Sınıfı"
menubarPath="HLAModellemeMenu/sampleGroup2"
style="push"
toolbarPath="sampleGroup2"
tooltip="Yeni Nesne Sınıfı oluştur."/>
<action
class="hlam76b.gef.actions.RunAddNewInteractionClass"
icon="icons/full/obj16/YeniInteractionClassOlustur_ToolBarIcon.gif"
id="hlam76b.gef.actions.RunAddNewInteractionClass"
label="Yeni - Etkileşim Sınıfı"
menubarPath="HLAModellemeMenu/sampleGroup2"
style="push"
toolbarPath="sampleGroup2"
tooltip="Yeni Etkileşim Sınıfı oluştur."/>
</actionSet>
</extension>
</plugin>

```

EK – 2 Hlam76b Ecore Meta Modeli

HLA için modellemede kullanılan olan meta model, hlam76b.ecore kütüğünde saklanır. Aşağıda bu kütüğün içeriği görülmektedir.

```
<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="hlam76b"
  nsURI="http://hlam76b.ecore" nsPrefix="hlam76b">
  <eAnnotations source="http://www.eclipse.org/emf/2002/GenModel">
    <details key="documentation" value="Rational Unified Process uses the &quot;Logical
View in Rose&quot; to organize the Design Model and the Process View and the optional
Business Object Model and Analysis Model."/>
  </eAnnotations>
  <eClassifiers xsi:type="ecore:EClass" name="HlaModel">
    <eOperations name="generateFED">
      <eAnnotations source="http://www.eclipse.org/emf/2002/GenModel">
        <details key="body" value="// generateFED buraya yazilacak."/>
      </eAnnotations>
    </eOperations>
    <eOperations name="generateFOM">
      <eAnnotations source="http://www.eclipse.org/emf/2002/GenModel">
        <details key="body" value="// generateFOM buraya yazilacak."/>
      </eAnnotations>
    </eOperations>
    <eStructuralFeatures xsi:type="ecore:EReference" name="objectClasses" upperBound="-1"
      eType="#//ObjectClass" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="attributes" upperBound="-1"
      eType="#//Attribute" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="dimensions" upperBound="-1"
      eType="#//Dimension" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="identification" lowerBound="1"
      eType="#//HlaModelIdentification" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="interactionClasses"
      upperBound="-1"
      eType="#//InteractionClass" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="basicDatatypes" upperBound="-1"
      eType="#//HlaBasicDataType" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="userDefinedDatatypes"
      upperBound="-1"
      eType="#//HlaUserDefinedDataType" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="ObjectClass">
    <eOperations name="GetFedInfo">
      <eAnnotations source="http://www.eclipse.org/emf/2002/GenModel">
        <details key="body" value="// getFedInfo metodu implement edilecek."/>
      </eAnnotations>
    </eOperations>
    <eOperations name="GetFomInfo">
      <eAnnotations source="http://www.eclipse.org/emf/2002/GenModel">
        <details key="body" value="// getFomInfo metodu implement edilecek."/>
      </eAnnotations>
    </eOperations>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="ObjectClassId"
      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="ObjectClassName"
      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="ObjectClassDocumentation"
      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="Publishable"
      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#/EBooleanObject"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="Subscribable"
      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#/EBooleanObject"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="attributes" lowerBound="1"
      upperBound="-1" eType="#//Attribute" containment="true"
      eOpposite="#//Attribute/ownerClass"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="parentObjectClass"
      eType="#//ObjectClass"
      eOpposite="#//ObjectClass/childObjectClasses"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="childObjectClasses"
      upperBound="-1"
```

```

        eType="#//ObjectClass" eOpposite="#//ObjectClass/parentObjectClass"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="Attribute">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Datatype" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Documentation"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Cardinality"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Unit" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Resolution"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Accuracy" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="AccuracyCondition"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="UpdateRateCondition"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Updateable"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Reflectable"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Transferable"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Acceptable"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Delivery" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="MessageOrdering"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="UpdateType"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="ownerClass" lowerBound="1"
eType="#//ObjectClass" transient="true" eOpposite="#//ObjectClass/attributes"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="availableDimensions"
upperBound="-1"
eType="#//Dimension"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="Dimension">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Documentation"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="Datatype"
eType="#//HlaUserDefinedDataType"
containment="true"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="Unit" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="NormalizationFunction"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaModelIdentification">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="pocTelephone"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="version" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="type" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="date" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="momVersion"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="applDomain"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="purpose" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="honorificName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="firstName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="secondName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>

```

```

    <StructuralFeatures xsi:type="ecore:EAttribute" name="pocName" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="pocOrganization"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="pocEmail" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="sponsor" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
  <Classifiers xsi:type="ecore:EClass" name="InteractionClass">
    <Operations name="GetFedInfo">
      <Annotations source="http://www.eclipse.org/emf/2002/GenModel">
        <details key="body" value="// GetFedInfo metodu implement edilecek."/>
      </Annotations>
    </Operations>
    <Operations name="GetFomInfo">
      <Annotations source="http://www.eclipse.org/emf/2002/GenModel">
        <details key="body" value="// GetFomInfo metodu implement edilecek."/>
      </Annotations>
    </Operations>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="ID" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="initiates"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="senses" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="reacts" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EBooleanObject"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Documentation"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Delivery" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="MessageOrdering"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EReference" name="childInteractionClasses"
upperBound="-1" eType="#//InteractionClass"
eOpposite="#//InteractionClass/parentInteractionClass"/>
    <StructuralFeatures xsi:type="ecore:EReference" name="parentInteractionClass"
eType="#//InteractionClass"
eOpposite="#//InteractionClass/childInteractionClasses"/>
    <StructuralFeatures xsi:type="ecore:EReference" name="parameters" upperBound="-1"
eType="#//Parameter" containment="true"/>
    <StructuralFeatures xsi:type="ecore:EReference" name="availableDimensions"
upperBound="-1"
eType="#//Dimension"/>
  </eClassifiers>
  <Classifiers xsi:type="ecore:EClass" name="Parameter">
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Datatype" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Documentation"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Cardinality"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Unit" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Resolution"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="Accuracy" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="AccuracyCondition"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
  <Classifiers xsi:type="ecore:EClass" name="HlaBasicDataType"
eSuperTypes="#//HlaUserDefinedDataType">
    <StructuralFeatures xsi:type="ecore:EAttribute" name="hlaBasicDataTypeName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="hlaBasicDataTypeSizeInBits"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="hlaBasicDataTypeInterpretation"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="hlaBasicDataTypeEndian"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <StructuralFeatures xsi:type="ecore:EAttribute" name="hlaBasicDataTypeEncoding"

```

```

        eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaSimpleDataType"
eSuperTypes="#//HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="units" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="resolution"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"
        defaultValueLiteral="NA"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="accuracy" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#/EString"
        defaultValueLiteral="NA"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="representation" lowerBound="1"
eType="#//HlaBasicDataType"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="semantics"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaEnumeratedDataType"
eSuperTypes="#//HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EReference" name="enumerators" upperBound="-1"
eType="#//HlaEnumerator" containment="true"
eOpposite="#//HlaEnumerator/enumDataType"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="representation" lowerBound="1"
eType="#//HlaBasicDataType"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaArrayDataType"
eSuperTypes="#//HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="cardinality"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="encoding" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="elementType" lowerBound="1"
eType="#//HlaUserDefinedDataType"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaFixedRecordDataType"
eSuperTypes="#//HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="encoding" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#/EString"
        defaultValueLiteral="HLAfixedRecord"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="filelds" lowerBound="1"
upperBound="-1" eType="#//HlaFixedRecordField" containment="true"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaVariantRecordDataType"
eSuperTypes="#//HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="encoding" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#/EString"
        defaultValueLiteral="HLAvariantRecord"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="discriminant" lowerBound="1"
eType="#//HlaVariantRecordDiscriminant" containment="true"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaEnumerator">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="hlaEnumeratorName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="hlaEnumeratorValue"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EIntegerObject"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="enumDataType" lowerBound="1"
eType="#//HlaEnumeratedDataType" transient="true"
eOpposite="#//HlaEnumeratedDataType/enumerators"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaFixedRecordField"
eSuperTypes="#//HlaUserDefinedDataType">
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="hlaFixedRecordFieldName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EAttribute" name="hlaFixedRecordFieldSemantics"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>
        <eStructuralFeatures xsi:type="ecore:EReference" name="fieldType" lowerBound="1"
eType="#//HlaUserDefinedDataType"/>
    </eClassifiers>
    <eClassifiers xsi:type="ecore:EClass" name="HlaVariantRecordDiscriminant">
        <eStructuralFeatures xsi:type="ecore:EAttribute"
name="hlaVariantRecordDiscriminantName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#/EString"/>

```

```

    <eStructuralFeatures xsi:type="ecore:EReference" name="type" lowerBound="1"
eType="#//HlaEnumeratedDataType"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="enumerators" upperBound="-1"
eType="#//VariantRecordDiscriminantEnumerator"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="HlaVariantRecordAlternative">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="hlaVariantRecordAlternativeName"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute"
name="hlaVariantRecordAlternativeSemantics"
eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="type" lowerBound="1"
eType="#//HlaUserDefinedDataType"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="VariantRecordDiscriminantEnumerator"
eSuperTypes="#//HlaEnumerator">
    <eStructuralFeatures xsi:type="ecore:EReference" name="alternative" lowerBound="1"
eType="#//HlaVariantRecordAlternative"/>
  </eClassifiers>
</ecore:EPackage>

```


EK – 3 Hlam76b Ecore Kütüğü Örneği

```
<?xml version="1.0" encoding="UTF-8"?>
<xml:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:hlam76b="http://hlam76b.ecore">
  <hlam76b:HlaModel/>
  <hlam76b:HlaModelIdentification/>
  <hlam76b:ObjectClass ObjectClassName="A" parentObjectClass="/36" childObjectClasses="/6
/4"/>
  <hlam76b:InteractionClass parentInteractionClass="/5" availableDimensions="/16 /14"/>
  <hlam76b:ObjectClass parentObjectClass="/2">
    <attributes Datatype="" Cardinality="2" AccuracyCondition="" UpdateType="Periodic"
availableDimensions="/14"/>
    <attributes availableDimensions="/16"/>
  </hlam76b:ObjectClass>
  <hlam76b:InteractionClass Name="" childInteractionClasses="/3 /12">
    <parameters/>
    <parameters/>
  </hlam76b:InteractionClass>
  <hlam76b:ObjectClass parentObjectClass="/2"/>
  <hlam76b:HlaBasicDataType/>
  <hlam76b:ObjectClass ObjectClassName="BABA" parentObjectClass="/36"
childObjectClasses="/9"/>
  <hlam76b:ObjectClass ObjectClassName="COCUK" parentObjectClass="/8"
childObjectClasses="/11"/>
  <hlam76b:ObjectClass ObjectClassName="COCUK3" parentObjectClass="/11"/>
  <hlam76b:ObjectClass ObjectClassName="COCUK2" parentObjectClass="/9"
childObjectClasses="/10"/>
  <hlam76b:InteractionClass ID="" Name="BABA_ETKI" childInteractionClasses="/13"
parentInteractionClass="/5" availableDimensions="/14 /15"/>
  <hlam76b:InteractionClass Name="COCUK_ETKI" MessageOrdering=""
parentInteractionClass="/12" availableDimensions="/14 /15"/>
  <hlam76b:Dimension Name="BOYUT_1" Documentation="BOYUT_1 kalan yakit miktarini temsil
eder." Unit="Litre" NormalizationFunction=""/>
  <hlam76b:Dimension Name="BOYUT_2"/>
  <hlam76b:Dimension Name="BOYUT_99"/>
  <hlam76b:HlaSimpleDataType name="SDT_1" representation="/19"/>
  <hlam76b:HlaBasicDataType name="BDT_1"/>
  <hlam76b:HlaBasicDataType name="BDT_2"/>
  <hlam76b:HlaUserDefinedDataType name="UDDT"/>
  <hlam76b:HlaArrayDataType/>
  <hlam76b:HlaEnumeratedDataType representation="/23">
    <enumerators/>
  </hlam76b:HlaEnumeratedDataType>
  <hlam76b:HlaBasicDataType name="BDT_11"/>
  <hlam76b:HlaArrayDataType elementType="/25"/>
  <hlam76b:HlaFixedRecordDataType>
    <filelds fieldType="/20"/>
  </hlam76b:HlaFixedRecordDataType>
  <hlam76b:HlaEnumeratedDataType representation="/19"/>
  <hlam76b:HlaSimpleDataType name="SDT_11" representation="/19"/>
  <hlam76b:HlaVariantRecordDataType name="VRDT_1"/>
  <hlam76b:HlaVariantRecordDiscriminant hlaVariantRecordDiscriminantName="VRD_1"
type="/22"/>
  <hlam76b:HlaVariantRecordAlternative hlaVariantRecordAlternativeName="VR_A1"/>
  <hlam76b:VariantRecordDiscriminantEnumerator hlaEnumeratorName="VRDE_1"
alternative="/35"/>
  <hlam76b:HlaVariantRecordDiscriminant hlaVariantRecordDiscriminantName="VRD_1" type="/33"
enumerators="/31 /34"/>
  <hlam76b:HlaEnumeratedDataType name="EDT_1111"/>
  <hlam76b:VariantRecordDiscriminantEnumerator hlaEnumeratorName="VRDE_22"
alternative="/35"/>
  <hlam76b:HlaVariantRecordAlternative hlaVariantRecordAlternativeName="VRA_111"/>
  <hlam76b:ObjectClass ObjectClassName="ObjectRoot" childObjectClasses="/8 /2"/>
</xmi:XMI>
```

EK – 4 Hlam76b Ecore Modeli GEF Kütüğü Örneği

```
<?xml version="1.0" encoding="ASCII"?>
<gef.model:EDiagram xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gef.model="http://www.metys.com/merlin/generation/gef/model.ecore"
modelResource="platform:/resource/hlam76b.tests/testQ.hlam76b">
  <contents location="45,506" width="183" height="100" diagram="/">
    <eObject href="testQ.hlam76b#/1"/>
  </contents>
  <contents name="" location="347,44" width="211" height="114" diagram="/"
incomingLinks="//@contents.30/@outgoingLinks.1">
    <eObject href="testQ.hlam76b#/2"/>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.4">
      <eReference href="http://hlam76b.ecore#/ObjectClass/childObjectClasses"/>
    </outgoingLinks>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.3">
      <eReference href="http://hlam76b.ecore#/ObjectClass/childObjectClasses"/>
    </outgoingLinks>
  </contents>
  <contents location="700,395" width="106" height="45" diagram="/"
incomingLinks="//@contents.29/@outgoingLinks.0">
    <eObject href="testQ.hlam76b#/3"/>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.13">
      <eReference href="http://hlam76b.ecore#/InteractionClass/availableDimensions"/>
    </outgoingLinks>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.11">
      <eReference href="http://hlam76b.ecore#/InteractionClass/availableDimensions"/>
    </outgoingLinks>
  </contents>
  <contents location="405,214" width="190" height="154" diagram="/"
incomingLinks="//@contents.1/@outgoingLinks.1">
    <eObject href="testQ.hlam76b#/4"/>
    <subNodes location="36,28" width="69" height="45" diagram="/">
      <eObject href="testQ.hlam76b#/4/@attributes.0"/>
      <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.11">
        <eReference href="http://hlam76b.ecore#/Attribute/availableDimensions"/>
      </outgoingLinks>
    </subNodes>
    <subNodes location="109,27" width="69" height="45" diagram="/">
      <eObject href="testQ.hlam76b#/4/@attributes.1"/>
      <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.13">
        <eReference href="http://hlam76b.ecore#/Attribute/availableDimensions"/>
      </outgoingLinks>
    </subNodes>
  </contents>
  <contents location="306,222" width="85" height="45" diagram="/"
incomingLinks="//@contents.1/@outgoingLinks.0">
    <eObject href="testQ.hlam76b#/6"/>
  </contents>
  <contents name="BABA" location="123,78" width="104" height="45" diagram="/"
incomingLinks="//@contents.30/@outgoingLinks.0">
    <eObject href="testQ.hlam76b#/8"/>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.6">
      <eReference href="http://hlam76b.ecore#/ObjectClass/childObjectClasses"/>
    </outgoingLinks>
  </contents>
  <contents name="COCUK" location="142,163" width="107" height="45" diagram="/"
incomingLinks="//@contents.5/@outgoingLinks.0">
    <eObject href="testQ.hlam76b#/9"/>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.8">
      <eReference href="http://hlam76b.ecore#/ObjectClass/childObjectClasses"/>
    </outgoingLinks>
  </contents>
  <contents name="COCUK3" location="153,316" width="85" height="45" diagram="/"
incomingLinks="//@contents.8/@outgoingLinks.0">
    <eObject href="testQ.hlam76b#/10"/>
  </contents>
  <contents name="COCUK2" location="133,238" width="116" height="45" diagram="/"
incomingLinks="//@contents.6/@outgoingLinks.0">
    <eObject href="testQ.hlam76b#/11"/>
    <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.7">
      <eReference href="http://hlam76b.ecore#/ObjectClass/childObjectClasses"/>
    </outgoingLinks>
  </contents>
</gef.model:EDiagram>
```

```

<contents name="BABA_ETKI" location="956,212" width="142" height="45" diagram="/"
incomingLinks="//@contents.29/@outgoingLinks.1">
  <eObject href="testQ.hlam76b#/12"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.10">
    <eReference href="http://hlam76b.ecore#//InteractionClass/childInteractionClasses"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.11">
    <eReference href="http://hlam76b.ecore#//InteractionClass/availableDimensions"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.12">
    <eReference href="http://hlam76b.ecore#//InteractionClass/availableDimensions"/>
  </outgoingLinks>
</contents>
<contents name="COCUK_ETKI" location="1043,296" width="151" height="144" diagram="/"
incomingLinks="//@contents.9/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/13"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.11">
    <eReference href="http://hlam76b.ecore#//InteractionClass/availableDimensions"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.12">
    <eReference href="http://hlam76b.ecore#//InteractionClass/availableDimensions"/>
  </outgoingLinks>
</contents>
<contents name="BOYUT_1" location="724,566" width="73" height="45" diagram="/"
incomingLinks="//@contents.10/@outgoingLinks.0 //@contents.9/@outgoingLinks.1
//@contents.3/@subNodes.0/@outgoingLinks.0 //@contents.2/@outgoingLinks.1">
  <eObject href="testQ.hlam76b#/14"/>
</contents>
<contents name="BOYUT_2" location="883,562" width="73" height="45" diagram="/"
incomingLinks="//@contents.9/@outgoingLinks.2 //@contents.10/@outgoingLinks.1">
  <eObject href="testQ.hlam76b#/15"/>
</contents>
<contents name="BOYUT_99" location="441,568" width="92" height="45" diagram="/"
incomingLinks="//@contents.3/@subNodes.1/@outgoingLinks.0 //@contents.2/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/16"/>
</contents>
<contents name="SDT_1" location="1582,130" width="70" height="30" diagram="/"
  <eObject href="testQ.hlam76b#/17"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.15">
    <eReference href="http://hlam76b.ecore#//HlaSimpleDataType/representation"/>
  </outgoingLinks>
</contents>
<contents name="BDT_2" location="1429,24" width="63" height="37" diagram="/"
incomingLinks="//@contents.14/@outgoingLinks.0 //@contents.21/@outgoingLinks.0
//@contents.20/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/19"/>
</contents>
<contents location="1926,775" width="154" height="102" diagram="/"
incomingLinks="//@contents.23/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/22"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.17">
    <eReference href="http://hlam76b.ecore#//HlaEnumeratedDataType/representation"/>
  </outgoingLinks>
  <subNodes location="29,17" width="99" height="45" diagram="/"
    <eObject href="testQ.hlam76b#/22/@enumerators.0"/>
  </subNodes>
</contents>
<contents name="BDT_11" location="2122,833" width="134" height="45" diagram="/"
incomingLinks="//@contents.16/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/23"/>
</contents>
<contents location="1216,177" width="178" height="80" diagram="/"
  <eObject href="testQ.hlam76b#/24"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.19">
    <eReference href="http://hlam76b.ecore#//HlaArrayType/elementType"/>
  </outgoingLinks>
</contents>
<contents location="1223,-6" width="159" height="141" diagram="/"
incomingLinks="//@contents.18/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/25"/>
  <subNodes location="10,12" width="131" height="45" diagram="/"
    <eObject href="testQ.hlam76b#/25/@filelds.0"/>
  </subNodes>
</contents>
<contents location="1581,-25" width="154" height="45" diagram="/"
  <eObject href="testQ.hlam76b#/26"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.15">

```

```

    <eReference href="http://hlam76b.ecore#//HlaEnumeratedDataType/representation"/>
  </outgoingLinks>
</contents>
<contents name="SDT_11" location="1580,44" width="150" height="45" diagram=""/>
  <eObject href="testQ.hlam76b#/27"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.15">
    <eReference href="http://hlam76b.ecore#//HlaSimpleDataType/representation"/>
  </outgoingLinks>
</contents>
<contents name="VRDT_1" location="1269,510" width="179" height="45" diagram=""/>
  <eObject href="testQ.hlam76b#/28"/>
</contents>
<contents name="VRD_1" location="1929,676" width="153" height="73" diagram=""/>
  <eObject href="testQ.hlam76b#/29"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.16">
    <eReference href="http://hlam76b.ecore#//HlaVariantRecordDiscriminant/type"/>
  </outgoingLinks>
</contents>
<contents name="VRDE_1" location="2650,319" width="99" height="51" diagram="/"
incomingLinks="//@contents.25/@outgoingLinks.1">
  <eObject href="testQ.hlam76b#/31"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.28">
    <eReference
href="http://hlam76b.ecore#//VariantRecordDiscriminantEnumerator/alternative"/>
  </outgoingLinks>
</contents>
<contents name="VRD_1" location="2792,323" width="174" height="45" diagram=""/>
  <eObject href="testQ.hlam76b#/32"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.26">
    <eReference href="http://hlam76b.ecore#//HlaVariantRecordDiscriminant/type"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.24">
    <eReference href="http://hlam76b.ecore#//HlaVariantRecordDiscriminant/enumerators"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.27">
    <eReference href="http://hlam76b.ecore#//HlaVariantRecordDiscriminant/enumerators"/>
  </outgoingLinks>
</contents>
<contents name="EDT_1111" location="2825,400" width="129" height="45" diagram="/"
incomingLinks="//@contents.25/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/33"/>
</contents>
<contents name="VRDE_22" location="2843,239" width="78" height="45" diagram="/"
incomingLinks="//@contents.25/@outgoingLinks.2">
  <eObject href="testQ.hlam76b#/34"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.28">
    <eReference
href="http://hlam76b.ecore#//VariantRecordDiscriminantEnumerator/alternative"/>
  </outgoingLinks>
</contents>
<contents name="VRA_111" location="2652,239" width="103" height="45" diagram="/"
incomingLinks="//@contents.24/@outgoingLinks.0 //@contents.27/@outgoingLinks.0">
  <eObject href="testQ.hlam76b#/35"/>
</contents>
<contents name="" location="628,203" width="259" height="157" diagram=""/>
  <eObject href="testQ.hlam76b#/5"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.2">
    <eReference href="http://hlam76b.ecore#//InteractionClass/childInteractionClasses"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.9">
    <eReference href="http://hlam76b.ecore#//InteractionClass/childInteractionClasses"/>
  </outgoingLinks>
  <subNodes location="57,43" width="75" height="45" diagram=""/>
    <eObject href="testQ.hlam76b#/5/@parameters.0"/>
  </subNodes>
  <subNodes location="143,43" width="75" height="45" diagram=""/>
    <eObject href="testQ.hlam76b#/5/@parameters.1"/>
  </subNodes>
</contents>
<contents name="ObjectRoot" location="290,-16" width="92" height="45" diagram=""/>
  <eObject href="testQ.hlam76b#/36"/>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.5">
    <eReference href="http://hlam76b.ecore#//ObjectClass/childObjectClasses"/>
  </outgoingLinks>
  <outgoingLinks xsi:type="gef.model:EReferenceLink" target="//@contents.1">
    <eReference href="http://hlam76b.ecore#//ObjectClass/childObjectClasses"/>
  </outgoingLinks>

```

```
</contents>  
</gef.model:EDiagram>
```

EK – 5 IEEE 1516.2 HLA.DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- HLA.dtd This is version 1.0 of a DTD file to fully define the OMT in XML terms -->
<!ELEMENT objectModel (
objects?,
interactions?,
dimensions?,
time?,
tags?,
synchronizations?,
transportations?,
switches?,
dataTypes?,
notes?)>
<!ATTLIST objectModel
DTDversion CDATA #FIXED "1516.2"
name CDATA #REQUIRED
nameNotes NMTOKENS #IMPLIED
type (FOM|SOM) #REQUIRED
typeNotes NMTOKENS #IMPLIED
version CDATA #IMPLIED
versionNotes NMTOKENS #IMPLIED
date CDATA #IMPLIED
dateNotes NMTOKENS #IMPLIED
purpose CDATA #IMPLIED
purposeNotes NMTOKENS #IMPLIED
appDomain CDATA #IMPLIED
appDomainNotes NMTOKENS #IMPLIED
sponsor CDATA #IMPLIED
sponsorNotes NMTOKENS #IMPLIED
pocName CDATA #IMPLIED
pocNameNotes NMTOKENS #IMPLIED
pocOrg CDATA #IMPLIED
pocOrgNotes NMTOKENS #IMPLIED
pocPhone CDATA #IMPLIED
pocPhoneNotes NMTOKENS #IMPLIED
pocEmail CDATA #IMPLIED
pocEmailNotes NMTOKENS #IMPLIED
references CDATA #IMPLIED
referencesNotes NMTOKENS #IMPLIED
other CDATA #IMPLIED
otherNotes NMTOKENS #IMPLIED>
<!ELEMENT objects (objectClass+)>
<!ELEMENT objectClass (attribute*, objectClass*)>
<!ATTLIST objectClass
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
sharing (Publish|Subscribe|PublishSubscribe|Neither) #IMPLIED
sharingNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT attribute EMPTY>
<!ATTLIST attribute
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
updateType (Static|Periodic|Conditional|NA) #IMPLIED
updateTypeNotes NMTOKENS #IMPLIED
updateCondition CDATA #IMPLIED
updateConditionNotes NMTOKENS #IMPLIED
ownership (Divest|Acquire|DivestAcquire|NoTransfer)
#IMPLIED
ownershipNotes NMTOKENS #IMPLIED
sharing (Publish|Subscribe|PublishSubscribe|Neither)
#IMPLIED
sharingNotes NMTOKENS #IMPLIED
dimensions NMTOKENS #IMPLIED
dimensionsNotes NMTOKENS #IMPLIED
transportation NMTOKEN #IMPLIED
transportationNotes NMTOKENS #IMPLIED
order (Receive|TimeStamp) #IMPLIED
orderNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED>
```

```

<!ELEMENT interactions (interactionClass+)>
<!ELEMENT interactionClass (parameter*, interactionClass*)>
<!ATTLIST interactionClass
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
sharing (Publish|Subscribe|PublishSubscribe|Neither)
#IMPLIED
sharingNotes NMTOKENS #IMPLIED
dimensions NMTOKENS #IMPLIED
dimensionsNotes NMTOKENS #IMPLIED
transportation NMTOKEN #IMPLIED
transportationNotes NMTOKENS #IMPLIED
order (Receive|TimeStamp) #IMPLIED
orderNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT parameter EMPTY>
<!ATTLIST parameter
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT dimensions (dimension*)>
<!ELEMENT dimension EMPTY>
<!ATTLIST dimension
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
upperBound CDATA #IMPLIED
upperBoundNotes NMTOKENS #IMPLIED
normalization CDATA #IMPLIED
normalizationNotes NMTOKENS #IMPLIED
value CDATA #IMPLIED
valueNotes NMTOKENS #IMPLIED>
<!ELEMENT time (timeStamp?, lookahead?)>
<!ELEMENT timeStamp EMPTY>
<!ATTLIST timeStamp
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT lookahead EMPTY>
<!ATTLIST lookahead
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT tags (updateReflectTag?,
sendReceiveTag?,
deleteRemoveTag?,
divestitureRequestTag?,
divestitureCompletionTag?,
acquisitionRequestTag?,
requestUpdateTag?)>
<!ELEMENT updateReflectTag EMPTY>
<!ATTLIST updateReflectTag
dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT sendReceiveTag EMPTY>
<!ATTLIST sendReceiveTag
dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT deleteRemoveTag EMPTY>
<!ATTLIST deleteRemoveTag
dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT divestitureRequestTag EMPTY>
<!ATTLIST divestitureRequestTag

```

```

dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT divestitureCompletionTag EMPTY>
<!ATTLIST divestitureCompletionTag
dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT acquisitionRequestTag EMPTY>
<!ATTLIST acquisitionRequestTag
dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT requestUpdateTag EMPTY>
<!ATTLIST requestUpdateTag
dataType NMTOKEN #REQUIRED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT synchronizations (synchronization+)>
<!ELEMENT synchronization EMPTY>
<!ATTLIST synchronization
label NMTOKEN #REQUIRED
labelNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
capability (Register|Achieve|RegisterAchieve|NoSynch) #IMPLIED
capabilityNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT transportations (transportation+)>
<!ELEMENT transportation EMPTY>
<!ATTLIST transportation
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
description CDATA #IMPLIED
descriptionNotes NMTOKENS #IMPLIED>
<!ELEMENT switches EMPTY>
<!ATTLIST switches
autoProvide (Enabled|Disabled) #IMPLIED
autoProvideNotes NMTOKENS #IMPLIED
conveyRegionDesignatorSets (Enabled|Disabled) #IMPLIED
conveyRegionDesignatorSetsNotes NMTOKENS #IMPLIED
attributeScopeAdvisory (Enabled|Disabled) #IMPLIED
attributeScopeAdvisoryNotes NMTOKENS #IMPLIED
attributeRelevanceAdvisory (Enabled|Disabled) #IMPLIED
attributeRelevanceAdvisoryNotes NMTOKENS #IMPLIED
objectClassRelevanceAdvisory (Enabled|Disabled) #IMPLIED
objectClassRelevanceAdvisoryNotes NMTOKENS #IMPLIED
interactionRelevanceAdvisory (Enabled|Disabled) #IMPLIED
interactionRelevanceAdvisoryNotes NMTOKENS #IMPLIED
serviceReporting (Enabled|Disabled) #IMPLIED
serviceReportingNotes NMTOKENS #IMPLIED>
<!ELEMENT dataTypes (basicDataRepresentations,
simpleDataTypes?,
enumeratedDataTypes?,
arrayDataTypes?,
fixedRecordDataTypes?,
variantRecordDataTypes?)>
<!ELEMENT basicDataRepresentations (basicData+)>
<!ELEMENT basicData EMPTY>
<!ATTLIST basicData
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
size CDATA #IMPLIED
sizeNotes NMTOKENS #IMPLIED
interpretation CDATA #IMPLIED
interpretationNotes NMTOKENS #IMPLIED
endian (Big|Little) #IMPLIED
endianNotes NMTOKENS #IMPLIED
encoding CDATA #IMPLIED
encodingNotes NMTOKENS #IMPLIED>
<!ELEMENT simpleDataTypes (simpleData+)>
<!ELEMENT simpleData EMPTY>

```



```

<!ATTLIST simpleData
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
representation NMTOKEN #IMPLIED
representationNotes NMTOKENS #IMPLIED
units CDATA #IMPLIED
unitsNotes NMTOKENS #IMPLIED
resolution CDATA #IMPLIED
resolutionNotes NMTOKENS #IMPLIED
accuracy CDATA #IMPLIED
accuracyNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED>
<!ELEMENT enumeratedDataTypes (enumeratedData+)>
<!ELEMENT enumeratedData (enumerator+)>
<!ATTLIST enumeratedData
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
representation NMTOKEN #IMPLIED
representationNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT enumerator EMPTY>
<!ATTLIST enumerator
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
values NMTOKENS #IMPLIED
valuesNotes NMTOKENS #IMPLIED>
<!ELEMENT arrayDataTypes (arrayData+)>
<!ELEMENT arrayData EMPTY>
<!ATTLIST arrayData
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
cardinality CDATA #IMPLIED
cardinalityNotes NMTOKENS #IMPLIED
encoding CDATA #IMPLIED
encodingNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED>
<!ELEMENT fixedRecordDataTypes (fixedRecordData+)>
<!ELEMENT fixedRecordData (field+)>
<!ATTLIST fixedRecordData
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
encoding CDATA #IMPLIED
encodingNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
<!ELEMENT field EMPTY>
<!ATTLIST field
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED>
<!ELEMENT variantRecordDataTypes (variantRecordData+)>
<!ELEMENT variantRecordData (alternative+)>
<!ATTLIST variantRecordData
name NMTOKEN #REQUIRED
nameNotes NMTOKENS #IMPLIED
discriminant CDATA #IMPLIED
discriminantNotes NMTOKENS #IMPLIED
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
encoding CDATA #IMPLIED
encodingNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED>
<!ELEMENT alternative EMPTY>
<!ATTLIST alternative
enumerator CDATA #REQUIRED
enumeratorNotes NMTOKENS #IMPLIED
name NMTOKEN #IMPLIED
nameNotes NMTOKENS #IMPLIED

```

```
dataType NMTOKEN #IMPLIED
dataTypeNotes NMTOKENS #IMPLIED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED>
<!ELEMENT notes (note+)>
<!ELEMENT note EMPTY>
<!ATTLIST note
name NMTOKEN #REQUIRED
semantics CDATA #IMPLIED
semanticsNotes NMTOKENS #IMPLIED >
```

ÖZGEÇMİŞ

Adı Soyadı : Oğuz Baktır

Doğum Yeri : Kayseri

Doğum Yılı : 1977

Medeni Hali : Evli

Eğitim ve Akademik Durumu:

Lise : 1992-1995 Ankara Fen Lisesi, ANKARA

Lisans : 1995-1999 Bilkent Üniversitesi Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü, ANKARA

Yabancı Dil : İngilizce, Almanca

İş Tecrübesi:

1999 – 2001 Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü,
Araştırma Görevlisi

2001 – 2004 Andersen Consulting, Accenture Ltd., Danışman

2004 – ... Havelsan A.Ş., Yazılım Mühendisi