

TÜRK HAVA KURUMU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAVALİMANI TERMİNAL MODELLEMESİ İÇİN BİR
METAMODEL

YÜKSEK LİSANS TEZİ
İsmet CAMCI

Elektrik ve Bilgisayar Anabilim Dalı
Elektrik ve Bilgisayar Mühendisliği Programı

MAYIS 2015

TÜRK HAVA KURUMU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAVALİMANI TERMİNAL MODELLEMESİ İÇİN BİR
METAMODEL

YÜKSEK LİSANS TEZİ

İsmet CAMCI

1203617002

Elektrik ve Bilgisayar Anabilim Dalı

Elektrik ve Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Deniz ÇETİNKAYA

MAYIS 2015

Türk Hava Kurumu Üniversitesi Fen Bilimleri Enstitüsü'nün 1203617002 numaralı Yüksek Lisans öğrencisi, İsmet CAMCI ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı HAVALİMANI TERMİNAL MODELLEMESİ İÇİN BİR METAMODEL başlıklı tezini, aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : Yrd. Doç. Dr. Deniz ÇETİNKAYA
Türk Hava Kurumu Üniversitesi

Jüri Üyeleri : Yrd. Doç. Dr. Engin DEMİR
Türk Hava Kurumu Üniversitesi

Doç. Dr. Sevil ŞEN
Hacettepe Üniversitesi

Tez Savunma Tarihi : 25.05.2015

TÜRK HAVA KURUMU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ'NE

Yüksek Lisans Tezi olarak sunduğum, HAVALİMANI TERMİNAL MODELLEMESİ İÇİN BİR METAMODEL adlı çalışmamın, tarafımdan akademik etik ve kurallara aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım kaynakların kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

25.05.2015

İsmet CAMCI

ÖNSÖZ

Bu tezde, havalimanı terminal binalarının tasarımının ya da modernizasyonun daha verimli bir şekilde yapılabilmesi için bu alanların modellenmesinde kullanılacak bir metamodel geliştirilmiştir. Geliştirilen bu metamodel ile terminal yerleşim planının tasarımının yapılabilmesi amaçlanmıştır. Ayrıca, havalimanı terminalleri için simülasyon modellerinin geliştirilmesi sırasında bir altyapı sağlaması hedeflenmiştir.

Çalışmamın yazılım mühendisliği ve modelleme ve simülasyon alanında faydalı olmasını dileyerek; tez çalışmamda yardımlarını hiç esirgemeyen, bilgi ve tecrübesiyle beni yönlendiren danışmanım Sayın Yrd. Doç. Dr. Deniz Çetinkaya'ya teşekkürü bir borç bilirim.

Ayrıca bu süre zarfında verdikleri destekten dolayı aileme sevgi ve teşekkürlerimi sunarım.

25.05.2015

İsmet CAMCI

İÇİNDEKİLER

ÖNSÖZ	iv
İÇİNDEKİLER	v
TABLolar	vii
ŞEKİLLER	viii
KISALTMALAR	x
ÖZET	xi
ABSTRACT	xiii
BİRİNCİ BÖLÜM	1
GİRİŞ	1
İKİNCİ BÖLÜM	4
LİTERATÜR ÖZETİ VE YAPILAN ÇALIŞMANIN ÖNEMİ	4
ÜÇÜNCÜ BÖLÜM	8
UYGULAMA ALANI İLE İLGİLİ BİLGİLER	8
3.1. Havalimanlarının Yapısı.....	8
3.2. Havalimanı Terminal Binaları	9
DÖRDÜNCÜ BÖLÜM	12
KULLANILAN YÖNTEMLER	12
4.1. Modelleme ve Model Güdümlü Yazılım Geliştirme.....	12
4.2. Modelleme ve Simülasyon	15
4.3. Alana Özgü Modelleme Dilleri ve Metamodelleme	16
4.4. Model Dönüşümü ve Model Dönüşüm Dilleri.....	17
BEŞİNCİ BÖLÜM	19
TERMİNAL MODELLEMESİ İÇİN ÖNERİLEN METAMODEL	19
5.1. Metamodelleme Ortamı.....	19
5.2. Metamodel Hakkında Genel Bilgiler	20
5.3. Havalimanı Terminal Modellemesi için Önerilen Metamodel.....	23
5.3.1. Terminal.....	24

5.3.2. Yolcu Akış	34
ALTINCI BÖLÜM	41
MODELLEME EDİTÖRÜ VE KULLANICI ARAYÜZLERİ	41
6.1. Terminal Yerleşim Modeli	41
6.1.1. Kat Modeli	42
6.1.2. İç Hat ve Dış Hat Modelleri.....	44
6.2. Yolcu Akış Modeli	52
6.2.1. İç Hatlar Giden Yolcu Senaryosu	54
6.2.2. Dış Hatlar Gelen Yolcu Senaryosu.....	55
6.3. Modellerin XMI formatında saklanması	58
6.4. Simülasyon Modellerine Dönüşüm	59
YEDİNCİ BÖLÜM	63
SONUÇLAR VE ÖNERİLER	63
KAYNAKLAR	67
EKLER	72
ÖZGEÇMİŞ	95

TABLÖLAR

Tablo 5.1: Kara tarafı servisleri.....	31
Tablo 5.2: Yolcu akış şeması model elemanları ve deęişkenleri.	40
Tablo 6.1: Örnek model dönüşümü için eşleştirmeler.	62

ŞEKİLLER

Şekil 1.1: Türkiye geneli havalimanları yolcu trafiği.....	1
Şekil 1.2: Türkiye geneli havalimanları uçak trafiği	2
Şekil 2.1: Bu tezde kullanılan yöntem ve ilgili disiplinler	7
Şekil 3.1: Havalimanı genel yerleşimi.....	8
Şekil 4.1: Yazılım geliştirmede model kullanımı	13
Şekil 4.2: Model, modelleme dili ve sistem arasındaki ilişki.....	15
Şekil 4.3: Metamodelleme	17
Şekil 4.4: Model dönüşümleri	18
Şekil 5.1: Terminal metamodeli tasarım şeması.....	22
Şekil 5.2: Havalimanı terminal metamodeli ana şeması.....	23
Şekil 5.3: Terminal metamodeli şeması.....	24
Şekil 5.4: Terminal metamodeli sınıf diyagramı.	25
Şekil 5.5: TerminalPlan modeli.	26
Şekil 5.6: TerminalPlan modeli tasarım ekranı değişken penceresi.....	26
Şekil 5.7: Kat modeli	27
Şekil 5.8: Kat modeli tasarım ekranı.	27
Şekil 5.9: Kat modelinde tanımlı kısıtlamalar	27
Şekil 5.10: Merdiven kontrolü OCL kodu.....	28
Şekil 5.11: Pasaport kontrolü OCL kodu.....	28
Şekil 5.12: Dış hatlar terminali bulunduğu halde pasaport kontrol alanının bulunmadığını gösteren tasarım ekranı uyarı penceresi.....	29
Şekil 5.13: Gerekli alan kontrolü OCL kodu.....	29
Şekil 5.14: Kara tarafı servisleri atomu	30
Şekil 5.15: Kara ve hava tarafı servisleri atomu.....	31
Şekil 5.16: Hava tarafı atomu metamodeli	33
Şekil 5.17: Yolcu akış metamodeli sınıf diagramı.	35
Şekil 5.18: Yolcu akış metamodeli ana şeması.	36
Şekil 5.19: Başla ve bitir elemanlarının görünümü	37
Şekil 5.20: Karar atomunun bağlantısı.	37
Şekil 5.21: Karar kaynak kuralı.....	38
Şekil 5.22: Karar hedef kuralı.....	38
Şekil 5.23: Karar yapısı kullanıcı ekranı	38
Şekil 5.24: Dağıt ve topla yolcu akış şeması elemanlarında tanımlı kurallar.....	39
Şekil 6.1: Modelleme ekranı TerminalPlan modeli özellikler ekranı.....	41

Şekil 6.2: Modelleme ekranı TerminalPlan ekranı altındaki kat modeli özellikler ekranı.....	42
Şekil 6.3: Kat modelinde bulunan havaalanı birimleri model elemanları.....	42
Şekil 6.4: Terminal sınır çizgileri dikey ve yatay çizgi uygulaması.....	44
Şekil 6.5: Terminal sınır çizgileri nokta uygulaması.....	44
Şekil 6.6: Hava tarafında bulunan farklı birimler.....	45
Şekil 6.7: Sabiha Gökçen Havalimanı kara tarafı giden yolcu bölümü.....	47
Şekil 6.8: Sabiha Gökçen Havalimanı kara tarafı gelen yolcu bölümü.....	48
Şekil 6.9: Sabiha Gökçen Havalimanı hava tarafı gelen yolcu bölümü.....	49
Şekil 6.10: Sabiha Gökçen Havalimanı hava tarafı giden yolcu bölümü.....	50
Şekil 6.11: Sabiha Gökçen Havalimanı hava tarafı giden yolcu -1. kat.....	51
Şekil 6.12: Sabiha Gökçen havalimanı hava tarafı giden yolcu asma katı.....	51
Şekil 6.13: Akış şeması model elemanları.....	52
Şekil 6.14: Akış şeması hava tarafı giden yolcu servisleri özellikler penceresi.....	53
Şekil 6.15: Yolcu akış şemasında bağlantı yapılamayacak yerler için örnekler.....	54
Şekil 6.16: Dağıt ve topla kural dışı kullanım için uyarı pencereleri.....	54
Şekil 6.17: İç hatlar giden yolcu akışı için hazırlanan yolcu akış modeli.....	56
Şekil 6.18: Dış hatlar gelen yolcu akışı için hazırlanan yolcu akış modeli.....	57
Şekil 6.19: Küçük bir havalimanının terminal yerleşim planı.....	60
Şekil 6.20: Küçük bir havalimanı için yolcu akış şeması.....	60
Şekil 6.21: Yolcu akış şemasına göre oluşturulan simülasyon modeli örneği.....	61

KISALTMALAR

IATA	: Uluslararası Hava Taşımacılığı Birliği (International Air Transport Association)
ICAO	: Uluslararası Sivil Havacılık Organizasyonu (International Civil Aviation Organization)
FAA	: Federal Havacılık Kurumu (Federal Aviation Administration)
OCL	: Nesne Kısıtlama Dili (Object Constraint Language)
MDA	: Model Güdümlü Mimari (Model Driven Architecture)
OMG	: Nesne Yönetim Grubu (Object Management Group)
MOF	: Üst Nesne Aracı (Meta Object Facility)
UML	: Birleşik Modelleme Dili (Unified Modeling Language)
CIM	: Hesaplama Bağımsız Model (Computation Independent Model)
PIM	: Platform Bağımsız Model (Platform Independent Model)
PSM	: Platform Bağımlı Model (Platform Specific Model)
XML	: Genişletilebilir İşaretleme Dili (Extensible Markup Language)
DHMİ	: Devlet Hava Meydanları İşletmesi
SHGM	: Sivil Havacılık Genel Müdürlüğü
GME	: Genel Modelleme Ortamı (Generic Modeling Environment)
MIC	: Modelle Entegre Hesaplama (Model Integrated Computing)
ATL	: Atlas Dönüşüm Dili (Atlas Transformation Language)
ISIS	: Yazılım Entegre Sistemler Enstitüsü (Institute for Software Integrated Systems)

ÖZET

HAVALİMANI TERMİNAL MODELLEMESİ İÇİN BİR METAMODEL

CAMCI, İsmet

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Deniz ÇETİNKAYA

Mayıs 2015, 95 sayfa

Kolay ve hızlı ulaşım olanakları sağlayan havacılık sektörü, her geçen gün daha da gelişerek büyümektedir. Havalimanlarının en önemli birimlerinden olan havalimanı terminallerinin modellenerek alt sistemlerin analiz edilmesi, bu alanların daha planlı, daha kaliteli ve ihtiyaçlara daha iyi cevap verecek alanlar olmasını sağlayacaktır.

Havalimanı analizi ve tasarımı için modelleme ve simülasyon (benzetim) tekniklerinden faydalanılması son yıllarda önem kazanmıştır. Ancak, bir modelleme ve simülasyon çalışması genelde her proje için yeniden yapılmakta ve bu nedenle oldukça uzun sürebilmektedir.

Bu çalışmada, havalimanı terminal modellemesi için bir metamodel geliştirilmiştir. Geliştirilen metamodel ile modelleme elemanları, özellikleri ve sınırları biçimsel olarak tanımlanmış bir modelleme dili önerilmiştir. Önerilen bu modelleme dili ile farklı boyutlardaki havalimanı terminallerinin yerleşim planlarının modellenmesi ve yolcu akış şemalarının çizilebilmesi hedeflenmiştir. Ayrıca, elde edilecek modellerin ileride otomatik olarak simülasyon modeline dönüştürülebilmesi amaçlanmıştır. Bu modeller aynı zamanda terminal yerleşim planının görsel olarak sunulmasında da kullanılabilir.

Metamodel kullanımı sayesinde, havalimanı terminallerinin analizinde, tasarımında ya da modernizasyonunda kullanılacak modeller alana özgü bir modelleme dili ile geliştirilecek ve model dönüşümleri için hazır olacaktır. Böylece,

bu alandaki modelleme ve simülasyon çalışmalarında önemli bir maliyet ve zaman tasarrufu sağlanacaktır.

Anahtar Kelimeler: metamodel, havalimanı modelleme, model güdümlü geliştirme, modelleme ve simülasyon, terminal tasarımı, alana özgü modelleme

ABSTRACT

A METAMODEL FOR AIRPORT TERMINAL MODELLING

CAMCI, İsmet

Master of Science, Electrical and Computer Engineering

Thesis Advisor: Asst. Prof. Dr. Deniz Çetinkaya

May 2015, 95 pages

The aviation sector which provides easy and fast transportation facilities is growing every day. Modeling and analysing the subsystems in an airport terminal, which is one of the most import units of the airports, will provide more structured areas with higher quality and will ensure that these areas better satisfy the needs.

The use of modelling and simulation techniques for analysis and design of the airports has gained importance in recent years. However, a modelling and simulation study is usually done from scratch for each project and therefore the projects can take quite a long time.

In this study, a metamodel for airport terminal modeling is developed. This metamodel formally defines a modeling language with its modeling elements, their properties and constraints. The proposed modelling language is aimed to be used for layout modeling of the airport terminals and defining passenger flow diagrams. Besides, the models are expected to be automatically transformed to simulation models in future research studies. At the same time, these model can be used for visual representation of the terminal layout.

By the use of the metamodeling method, models that can be used in the analysis, design or modernization of the airport terminals will be developed with a domain specific modelling language and will be ready for model transformations. As a result,

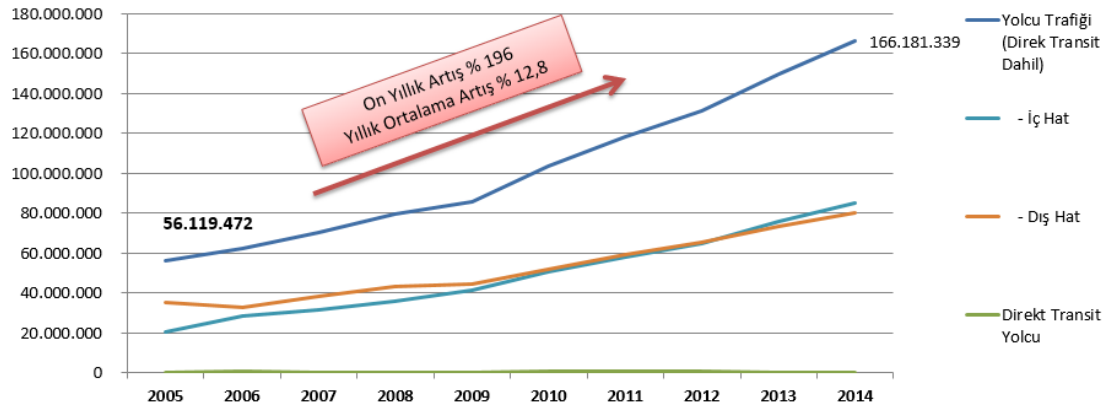
this research will help to reduce the costs in the airport terminal related modelling and simulation studies.

Keywords: metamodel, airport modeling, model driven development, modeling and simulation, terminal design, domain specific modeling.

BİRİNCİ BÖLÜM

GİRİŞ

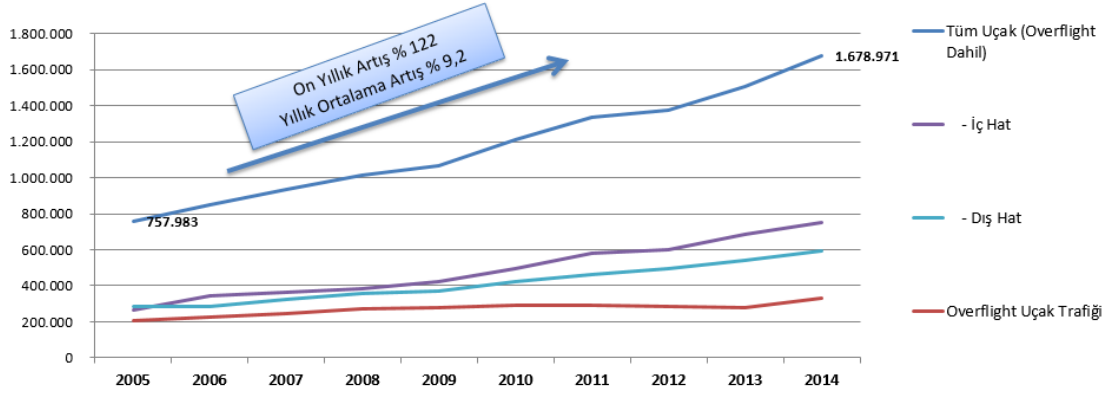
Küreselleşen dünyada ulaşımın çabuk ve kolay olması, gelişen teknoloji ile hızlanan insan hayatında, oldukça önem arz etmektedir. Bununla beraber, hızlı bir ulaşım imkânı sunan hava ulaşımına olan talep her geçen gün artmaktadır. Hava taşımacılığına olan talebin artması ile birlikte havacılık sektörü de sürekli olarak gelişmektedir. Birçok ülke ile birlikte Türkiye’de de havalimanı sayısı son yıllarda artmıştır. Ülkemizde 2003 yılında 26 adet aktif olarak kullanılan havalimanı bulunurken, 2014 sonu itibari ile 53 adet havalimanı sayısına ulaşılması bu sektördeki büyümeyi göstermektedir [1]. Şekil 1.1 ve Şekil 1.2’de son 10 yıla ait Türkiye geneli yolcu ve uçak trafiğindeki artış gösterilmektedir.



Şekil 1.1: Türkiye geneli havalimanları yolcu trafiği [1].

Havalimanlarındaki uçak ve yolcu trafiğinin her geçen gün artması, havalimanlarının artan ihtiyaçları karşılayacak şekilde ve yeni nesil teknolojiler kullanılarak yapılmasının önemini artırmaktadır. Tasarlanacak olan havalimanlarındaki yolcu memnuniyetinin ve sunulan hizmetlerin kalitesinin

artırılması için, ihtiyaçlara karşılık verebilme noktasında alt sistemlerin analiz edilmesi oldukça önemlidir.



Şekil 1.2: Türkiye geneli havalimanları uçak trafiği [1].

Havalimanlarının kullanılabilirliğinin veya ihtiyaca uygunluğunun havalimanı inşa edildikten sonra tespiti uzun zaman almasının yanında düzeltilmesi de oldukça maliyetli problemlerdir. Yolcuların seyahat sonrası terminalden ayrılırken ya da aktarmalar sırasında, birimler arası geçişlerde uzun mesafeler yürümeleri, yürüyen merdiven gibi geçişleri kolaylaştıracak araçları kullanamamaları, kapıları bulmakta zorluk yaşamaları, tekrarlanan güvenlik kontrollerinden geçmeleri, vb. gibi eksik ya da yanlış planlamadan kaynaklanan problemlerden birkaçıdır.

Sistemlerin modelleme ve simülasyon yöntemleri ile analiz edilip planlanması tasarım aşamasında kullanılan etkili bir yöntemdir [2]. Havalimanı terminallerinin de önceden modellenmesi ve analiz edilmesi bu alanların daha kaliteli ve daha verimli ortamlar haline gelmesini sağlayacaktır. Havalimanı modellemesi ve simülasyonu, dünyada yaygın bir şekilde uygulanmaktadır [3, 4, 5]. Yeni yapılacak havalimanları için modelleme ortamlarının kullanılması, ileriye dönük ihtiyaçların belirlenmesi ya da mevcut şartlarda sistemin analizi ve çözülmesi için oldukça önemlidir. Ayrıca, bu ortamlar mevcut havalimanlarının modernizasyonu sırasında da değişiklikler yapılmadan önce, olası etkilerin izlenebilmesinde kullanılabilir. Ayrıca, bu ortamlar mevcut havalimanlarının modernizasyonu sırasında da değişiklikler yapılmadan önce, olası etkilerin izlenebilmesinde kullanılabilir.

Modelleme ve simülasyon çalışmalarında simülasyon modellerinin her proje için yeniden geliştirilmesi zaman almakta ve maliyetleri yükseltmektedir. Bu nedenle, son yıllarda simülasyon alanında model güdümlü geliştirme yaklaşımları öne çıkmıştır [6, 7, 8]. Model güdümlü geliştirme, modelleri temel geliştirme aracı olarak ele alan ve

metamodelleme ve model dönüşümleri gibi yöntemleri kullanarak modellerden kod üretmeyi sağlayan bir yöntemdir [9, 10, 11]. Modeller, soyutlama (abstraction) yoluyla karmaşık problemlerin çözümünde ya da kompleks yapıların inşasından önce sistemin gerçeğe yakın bir benzerini görmemizi sağlar [9]. Metamodelleme, bir modelleme dilinin ya da yönteminin model olarak ifade edilmesidir. Geliştirilen bu üst modele ise metamodel denir.

Bu çalışmada, havalimanları içerisinde yer alan terminallerin yapımında maliyet ve zaman tasarrufu sağlanması için, havalimanı terminal modellenmesinde kullanılacak bir metamodel geliştirilmiştir. Geliştirilen metamodel iki bölümden oluşmaktadır. Birinci bölüm ile inşa edilecek farklı büyüklükteki havalimanı terminallerinin modellenmesine olanak sağlanmakta ve tasarlanan modeller ile terminal yerleşim planı da görsel olarak elde edilebilmektedir. İkinci bölümde, geliştirilen modelleme dili ile kullanıcılar, sistem analizinde ve simülasyon dönüşümünde önemli bir veri olan yolcu akış şeması modellerini oluşturabileceklerdir. Ayrıca model güdümlü geliştirme yaklaşımı sayesinde, bu metamodel ile tanımlanan modeller için, model dönüşümleri de yazılabilecektir. Böylece, geliştirilen modelleme dili kullanılarak herhangi bir havalimanı terminali modellemesi yapılabilecek, ve bu modeller kullanılarak üretilecek simülasyon modelleri sayesinde havalimanı terminallerinin farklı açılardan simülasyonu yapılabilecektir.

Geliştirilen metamodel, standartlar kapsamındaki kural ve kısıtlamaları içerdiği için olası eksiklik ve hatalar çok daha önce fark edilebilmektedir. Ayrıca, modelleme sırasında değişikliklerin kolay yapılabilmesi sayesinde birçok olasılık göz önüne alınarak farklı modeller hazırlanabilmektedir. Bu sayede, en optimum çözüme ulaşılması ve en verimli tasarımın oluşması daha ekonomik, daha az zamanda ve daha kolay bir şekilde yapılabilmektedir.

İKİNCİ BÖLÜM

LİTERATÜR ÖZETİ VE YAPILAN ÇALIŞMANIN ÖNEMİ

Hava ulaşımı 1900’lü yılların başlarında askeri amaçlı kullanılırken, sivil havacılığın başlaması ile havalimanı terminalleri oluşturulmuş ve gelişerek günümüzdeki halini almıştır. Genel olarak havalimanı terminali, yer ve hava taşımacılığı arasındaki bağlantıyı sağlayan, yük ve yolcu taşımacılığına hizmet eden, yerel ulaşım sistemlerine bağlı karmaşık yapılardır. Günümüzde ise hava ulaşımına talebin her geçen gün artması bu yapılardaki yolcu yoğunluğunu da artırmaktadır. Ancak, havalimanı terminal tasarımlarının önceden test ve analiz edilmeden havalimanlarının inşa edilmesi, terminalin belli alanlarında yoğunlukların yaşanması, terminal birimlerinin eksik olması ya da yanlış bölgelere yerleştirilmesi ve ileriye dönük değişimlere kapalı olması gibi sorunlara neden olmaktadır. Bu sorunlar çerçevesinde akademik olarak bilişim, mimari ve sivil havacılık alanlarında bazı çalışmalar yapılmıştır.

Literatürdeki çalışmalardan bahsetmeden önce, kavram karmaşasını önlemek için havalimanı ve havaalanı terimlerinin açıklaması verilecektir. Çoğu zaman birbirinin yerine de kullanılan bu terimler, Devlet Hava Meydanları İşletmesi Genel Müdürlüğü (DHMI) terimler sözlüğünde ayrı olarak tanımlanmıştır. Buna göre, havaalanı “Bütünü ya da bir bölümü içinde hava araçların; iniş, kalkış ve yer hareketlerini gerçekleştirebilmeleri için karada veya suda oluşturulmuş, (bina, tesis ve teçhizatla donatılmış) tanımlanmış saha” şeklinde geniş kapsamlı olarak tanımlanmıştır. Havalimanı ise, “Uluslararası hava trafiği, gelişi ve gidişlerine hizmet vermek amacıyla tesis edilmiş olup, gümrük, göçmenlik, halk sağlığı, hayvan ve bitki karantina işlemleri ve benzeri işlemlerin bünyesinde vakit kaybedilmeksizin yürütüldüğü havaalanıdır” şeklinde tanımlanmıştır [12].

Havacılık ile ilgili konularda devletin resmi belgelerinde ya da arařtırmalarda, İngilizce'deki airport terimine karřılık olarak havaalanı ve havalimanı terimlerinin ikisi de kullanılabildiđi için, akademik çalıřmalarda her iki terim de kullanılmıřtır. Bu bölümde, çalıřmalardaki orijinal ifadeler aynen kullanılacaktır. Ancak, tezin genelinde havalimanı kelimesi tercih edilmiřtir.

Arusođlu [13], "Havaalanı yolcu hareket simülasyon model önerisi" isimli yüksek lisans tez çalıřmasında, havaalanı terminalindeki mekanların ve bu mekanlara ait parametrelerin yolcu tip ve hareketleriyle birlikte ele alınarak daha sonra geliştirilebilecek simülasyon modelleri için kullanılmasını önermiřtir. Mutlu'nun [14] "Havalimanı ve havaalanı terminal yapılarında yolcu beğenisinin arařtırılması ve ölçülmesi" isimli yüksek lisans tez çalıřmasında 7 adet havaalanı terminali ile ilgili, yolculara memnuniyet anketi yapılmıřtır. Bu anket sonucunda mekân ve hizmet analizleri yapılarak, terminallerin iyileřtirilmesine yönelik ve yeni terminal tasarımları için yol gösterici olacak sonuçlar sunulmuřtur. Acımert'in [15], "Çađdař havaalanı, terminal planlamasında akıř olgusu, plan tipleri, tip seçimi ve kriterleri" isimli yüksek lisans tez çalıřmasında havalimanı terminallerinin niteliđi ve içeriđinin deđiřime uđradıđı belirtilerek kullanıcı ve havalimanı iliřkisi incelenmiřtir. Terminallerin tip sınıflandırmaları ve birbirleri ile kıyaslamaları yapılmıřtır.

Yurt dıřında, havalimanlarının performansına ve verimliliđine yönelik çalıřmalar oldukça fazladır. Bu alanda, modelleme ve simülasyon tekniklerinin kullanılması da giderek artmaktadır. Guizzi [3] havalimanı terminallerindeki yolcu akıřının modellenmesi için ayrıklı olay simülasyonu (discrete event simulation) yöntemi ile bir model önermiřtir. Önerdikleri modeli güney İtalya'daki bir havalimanı için test etmiřlerdir. Takakuwa ve Oyama [4] benzer bir çalıřmayı Japonya'daki bir havalimanı için yapmıřlardır. Freivalde ve Lace'nin [16] çalıřmalarında da, Litvanya'daki bir havalimanında yolcu akıřının iyileřtirilmesine yönelik arařtırma yapılmıřtır. Lazzaroni [17] ise Vancouver havalimanı tasarımı temel alınarak yolcu ve bagaj verilerine göre yolcu ve bagaj akıřı için bir model önerisinde bulunmuřtur.

Türkiye'deki Sivil Havacılık Genel Müdürlüğü'ne (SHGM) karřılık gelebilen, açılımı Federal Havacılık Kurumu (Federal Aviation Administration) olan FAA, mevcut havaalanları ve tasarlanacak havaalanları için, hava sahası üzerindeki uçakların hareketini izleyen Havalimanı ve Havaalanı Simülasyon Modeli (SIMMOD) isimli bir simülasyon programını kullanmaktadır [18]. SIMMOD girdi parametreleri

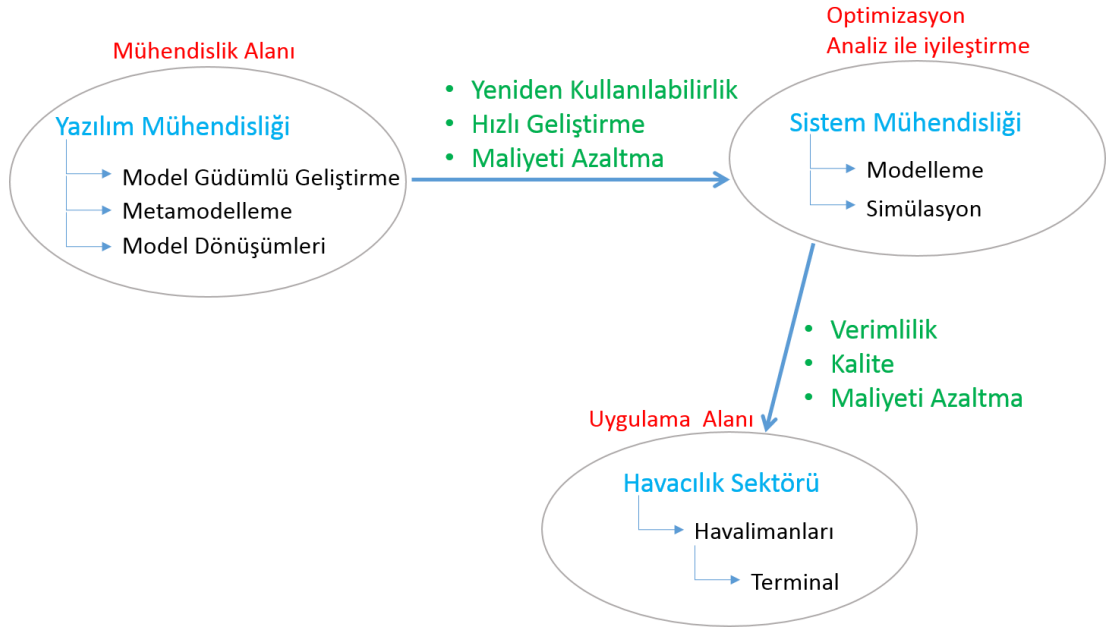
olarak uçak uçuş programlarını, hava trafik kontrol kuralları ve prosedürlerini, hava sahası rota yapılarını, havalimanı düzeni ve yer kontrol prosedürlerini içererek uçuşlarda, hava ile yer arasındaki durum ve yoğunluğun simülasyon ile analiz edilmesini sağlamaktadır. Bir coğrafi bilgi sistemi programından faydalanılarak da havalimanı coğrafi konumu mevcut simülasyon programına entegre edilerek analizlerde gerçeğe daha yakın sonuçlar elde edilmesi sağlanmıştır. Bu çalışmalarda, havalimanı iç tasarımına yönelik değil, uçuşa yönelik analiz amaçlı modelleme ve simülasyon çalışmaları yapılmıştır [19].

Literatürdeki uygulama çalışmaları incelendiğinde, havalimanı terminalleri ile ilgili araştırmaların genelde yolcu hareketleri ve ihtiyaçlarına yönelik olduğu ve simülasyon modellerinin belirli bir havalimanı için geliştirildiği görülmektedir. Genel amaçlı kullanılabilir, havalimanı terminal modellemesine yönelik çalışmalar genelde bileşen tabanlı geliştirme yaklaşımı ile beraber verilmiştir. Verbraeck ve Valentin'in [5] havalimanı terminal modellemesi için simülasyon modelleme bileşenleri sunan çalışması buna örnek verilebilir.

Bu çalışmada, havalimanları içerisinde yer alan terminallerin modellenmesinde kullanılabilir bir metamodel geliştirilmiştir. Geliştirilen metamodel, havalimanları terminal modellemesi için alana özgü bir modelleme dili tanımlamaktadır. Bu modelleme dili ile yukarıda bahsedilen akademik çalışmalardaki model önerilerinin akış şeması çizimleri ve terminal tasarımları yapılabilir. Metamodel geliştirilirken ileride bu çalışmanın model dönüşümleri ile simülasyon modeline dönüştürülebilmesi de hesaba katılmıştır. Simülasyon sayesinde farklı model önerileri deneyerek daha detaylı analizler yapılabilir.

Sonuç olarak, bildiğimiz kadarıyla, farklı havalimanları için kullanılabilir ve model güdümlü geliştirme yaklaşımı ile oluşturulmuş, ileride simülasyon amaçlı kullanılabilir, alana özgü bir modelleme yaklaşımı sunan bir çalışma bulunmamaktadır. Bu bağlamda, havalimanı terminal modellemesi için geliştirilen bu metamodel havacılık sektöründe önemli açılımlara zemin hazırlayabilecektir. Bu çalışma, metamodelleme yöntemi kullanılarak alana özgü bir havalimanı terminal modelleme yöntemi sunmakta ve model güdümlü geliştirme yaklaşımı ile geliştirilen modellerin simülasyon modellerine dönüştürülebilmesi için altyapı sağlamaktadır. Bu sayede yazılım mühendisliği, sistem mühendisliği ve havacılık sektörünü bir araya getirmesi ile diğerlerinden ayrılmaktadır.

Bu çalışmada kullanılan yöntemler ve uygulama alanı ile ilgili ana başlıklar Şekil 2.1’de ilgili oldukları alanlara göre gruplandırılarak gösterilmiştir. Bu tezde, model güdümlü geliştirme ve metamodelleme konuları öne çıkmaktadır.



Şekil 2.1: Bu tezde kullanılan yöntem ve ilgili disiplinler.

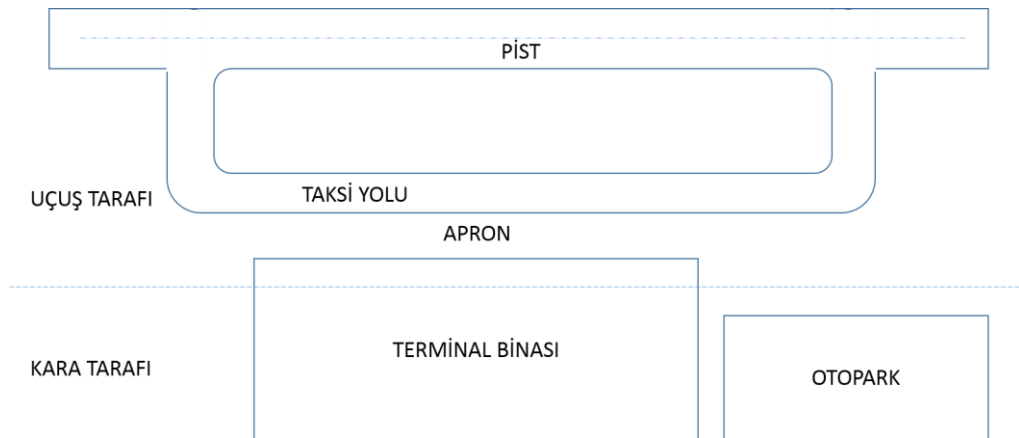
ÜÇÜNCÜ BÖLÜM

UYGULAMA ALANI İLE İLGİLİ BİLGİLER

İlk yolcu ve kargo taşımacılığı uçuşları, 1908 yılında ABD'de başlamıştır. Gelişen teknolojiye paralel olarak uçakların sayısının, kapasitesinin ve özelliklerinin artması ile havayolu taşımacılığına olan talepte sürekli artmış ve ulaşımda önemli bir paya sahip olmuştur. Havaalanlarındaki pist sayısı, pistlerin uzunluğu, terminallerin yapısı, uçuş kapılarının sayısı gibi faktörler havalimanlarının kapasitesinin belirlenmesinde ve havayolu ulaşımının gelişiminde önemli bir yer almıştır [20, 21].

3.1. Havalimanlarının Yapısı

Havalimanlarının her geçen gün artan uçak ve yolcu trafiğini sorunsuz bir şekilde, aksamalar olmadan sağlayabilmesi için, havalimanlarındaki uçuş tarafının (airpart) ve kara tarafının (landpart) planlı, ihtiyaca uygun ve gelişime açık bir şekilde tasarlanması gerekmektedir. Şekil 3.1'de havalimanlarındaki temel birimler ve birimler arası ilişkiler gösterilmiştir [21]. Bu birimler yolcu terminal binası, otopark alanı, pist, taksi yolu, apron, vb. gibi olarak tanımlanmıştır.



Şekil 3.1: Havalimanı genel yerleşimi.

Havalimanları tasarlanırken ve havayolu trafiği işletilirken uluslararası standartlar dikkate alınmaktadır. Bu standartları tanımlayan kuruluşlardan en önemli bir tanesi Uluslararası Hava Taşımacılığı Birliği'dir (IATA, International Air Transport Association). IATA 1945'te kurulmuştur, ancak kökeni 1919'da kurulan Uluslararası Hava Trafik Birliği'ne (International Air Traffic Association) dayanmaktadır. Sadece havayolu firmalarının üye olabildiği IATA, havayolu ulaşımının daha kaliteli, güvenli ve ekonomik olması için 118 ülkeden 250'nin üzerinde üyeye sahip olan uluslararası bir kuruluştur [22]

Uluslararası anlaşmalarla hava ulaşımı ile ilgili standartları belirleyen bir başka kuruluş da Uluslararası Sivil Havacılık Organizasyonu'dur (ICAO, International Civil Aviation Organization). ICAO 1944 yılında Birleşmiş Milletler bünyesinde kurulan bir organizasyondur ve 191 ülke bu yapıya üyedir. Bu kuruluşlar, sivil havacılığın düzenli ve güvenli bir şekilde yapılmasını sağlamak için çeşitli alanlarda standartlar belirlemekte ve ayrıca yeni projeler geliştirilmesine ön ayak olmaktadır [23]. IATA ve ICAO havaalanları için belli kodlar kullanmaktadır. ICAO kodları genellikle uçuş seyri, planlanması ve trafiği üzerine işlemlerde kullanılırken, IATA havalimanındaki uçak iniş-kalkış zamanlamaları, bilet kontrol ve bagaj işlemleri gibi yer hizmetleri ile ilgilidir.

3.2. Havalimanı Terminal Binaları

Yolcuların uçuşla ilgili gidiş-geliş işlemlerini gerçekleştirdiği ve bunlarla ilgili hizmetleri aldığı havalimanlarındaki binalara terminal denir [12]. Gidiş işlemleri bilet alış, bilet kontrol, pasaport kontrol, uçağa biniş, vb. gibi işlemlerdir. Geliş işlemleri ise uçaktan iniş, (bagaj alım, gümrük kontrol, vb. gibi) işlemlerdir. Terminal binaları kapasite ve büyüklük gibi bazı kriterlere göre küçük, orta, ve büyük olarak sınıflandırılmıştır. Örneğin, küçük bir havalimanı azami yılda 1 milyon yolcuya hizmet vermektedir. Terminal binalarındaki alanlardan bazıları ve kullanılan havacılık terimleri aşağıda açıklanmıştır. Bu alanlardan özellikle pasaport, gümrük ve bilet kontrol noktaları, bagaj alım ile check-in alanları yolcuların uçağa binene kadar terminalde geçirdikleri süreyi etkileyen önemli birimlerdir.

Geliş-Gidiş: Geliş (arrival), uçakla gelen yolcunun terminale gelişini ve terminaldeki işlemlerini; gidiş (departure) ise uçakla gidecek yolcunun terminalden çıkışını ve gitmek için terminalde yaptığı işlemleri temsil eder. Gelen yolcunun

kullandığı kat ayrı bir kat ise geliş katı olarak adlandırılır. Aynı şekilde, giden yolcunun kullandığı kat ayrı bir kat ise gidiş katı olarak adlandırılır.

İç-Dış Hatlar: Ülke sınırları içindeki havaalanları arasında yapılan seferler iç hatlar (domestic) olarak adlandırılır. Ülke sınırları dışında bir veya daha fazla ülkeye yapılan seferler ise dış hatlar (international) olarak adlandırılır [12].

Kara-Hava Tarafı: Havalimanı terminal binalarında, biletli veya biletsiz yolcuların ve yolcu yakınlarının beraber kullanabildiği alanları kapsayan bölüm kara tarafı olarak adlandırılırken; sadece biletli yolcuların geçebileceği ve o alandaki hizmetleri kullanabileceği terminal bölümü ise hava tarafı veya uçuş tarafı olarak adlandırılır [24] .

Terminal Giriş-Çıkış: Giden biletli-biletsiz yolcuların veya yolcu yakınlarının terminal binasına girmek ve binadan çıkmak için kullandığı bölümdür.

Pasaport ve Bilet Kontrol: Terminal binası kara tarafından hava tarafına geçişte uçuş bileti veya uluslararası uçacak yolcular için pasaport kontrolü yapılan noktalardır.

Gümrük Kontrol: Havalimanlarının gelen yolcu alanlarında yapılan gümrük kontrol işlemleridir.

Check-in: Terminallerde, yolcuların bilet ve bagaj kontrollerinin ve uçuş için gerekli işlemlerinin yapıldığı bölümlerdir. Havayolu firmaları tarafından veya havalimanı yer hizmetleri kuruluşları tarafından görevlendirilmiş kişiler tarafından yapılır.

Bagaj Alım Alanları: Gelen yolcuların bagajlarını teslim aldıkları noktalardır.

Yolcu Bekleme Odaları (Giden Yolcu Salonları): Yolcuların uçağa binmeden önce bekledikleri salonlardır.

Transit Yolcu: Havalimanına geliş yaptıktan sonra, çıkış yapmadan aynı biletle seyahatlerine devam eden yolculardır. Teknik problemlerden ötürü uçak değiştiren fakat aynı uçuş sayılı uçakla devam eden yolcular direkt transit yolcular olarak sayılır [12].

Transfer Yolcu: Farklı biletle ve farklı uçakla seyahatlerine devam eden ama 24 saat içinde havalimanına varıp ayrılan yolculardır. Bu yolcular varışta ve kalkışta olmak üzere iki kez sayılırlar [12].

Terminal Servisleri: Havalimanı terminal işletmesinin yolculara sunduđu ücretli veya ücretsiz hizmetlerin tamamıdır. Örneđin, danışma, güvenlik kontrol işlemleri, tuvaletler, vb. gibi bu servislerden sayılabilir.

Özel servisler: Özel şirketlerin yolculara sunduđu ücretli veya ücretsiz hizmetlerin tamamıdır. Örneđin, restoranlar, havayollarının bilet satış ofisleri, alışveriş mağazaları, vb. gibi bu servislerden sayılabilir.

DÖRDÜNCÜ BÖLÜM

KULLANILAN YÖNTEMLER

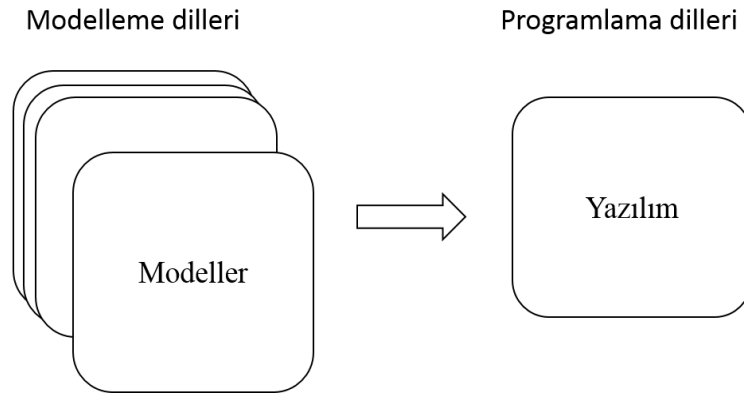
4.1. Modelleme ve Model Güdümlü Yazılım Geliştirme

Günümüzde, teknolojik gelişmelerin sonucu olarak hemen hemen her alanda daha fonksiyonel ve daha karmaşık sistemler ortaya çıkmıştır. Bu sistemlerin hızlı ve kaliteli bir şekilde oluşturulup, verimli ve sorunsuz olarak çalışması ise önemli bir ihtiyaç haline gelmiştir. Aynı zamanda, bu karmaşık sistemlerin oluşturulmasında ya da işletilmesinde kullanılan yazılım sistemleri de daha karmaşık hale gelmiştir [25].

Modelleme yaklaşımı ve model tabanlı üretim yaklaşımı birçok farklı mühendislik alanında kullanılmaktadır. Karmaşık ve yapımından sonra tekrar düzeltilmesi maliyetli bir sistemin veya böyle bir sistemde yer alacak bir parçanın daha kaliteli bir şekilde oluşturulmasını sağlamak için, geliştirme aşamasından önce modelleme yoluyla analiz ve tasarım gibi yöntemler uygulanmaktadır.

Modelleme, en genel anlamıyla bir sistemin bir amaç için temsili olarak ifade edilmesidir. Sistemi temsil eden ve bir modelleme dili ile oluşturulan, modelleme işinin çıktısına model denir. Yazılım projeleri sonucunda ürün olarak ortaya çıkan yazılımlar da aslında birer sistemdir ve modelleme yaklaşımı yazılımlar için de yıllardır kullanılmaktadır. Geliştirilecek olan yazılım, daha kodlamaya başlamadan önce modellenir ve tasarlanır. Analiz veya tasarım aşamasında hazırlanan modeller, ileride geliştirilecek olan yazılımı farklı açılardan temsil eder. Büyük bir yapının inşasında veya fonksiyonel bir makinanın yapımında model kullanımı ne kadar önemli ise gelişmiş bir yazılımın tasarımında da, soyutlama yaparak modelleme yaklaşımının kullanılması önemli avantajlar sağlamaktadır. Örneğin, mantıksal hatalar daha kodlama aşamasına geçmeden düzeltilebilmekte, yazılım ekibi içerisinde daha iyi ve etkili bir iletişim sağlanmakta, tekrar kullanılabilir kod oranı artmakta, tasarım aşamasında sistem alt bileşenlerine ayrılmışsa proje yönetimi kolaylaşmaktadır. Şekil

4.1’de farklı modelleme dilleri ile geliştirilmiş modeller kullanılarak yazılımın kodlanması ifade edilmiştir. Modellerden koda geçiş, klasik yaklaşımlarda programı yazacak kişilerin sorumluluğunda iken, güncel yaklaşımlarda otomatik kod üretimi sayesinde daha sistematik yapılabilmektedir. Yaygın olarak kullanılan modelleme dilleri, Birleşik Modelleme Dili (UML, Unified Modeling Language), İş Süreci Modeli ve Gösterimi (BPMN, Business Process Model and Notation), Bütünleşik Tanımlama Dilleri (IDEF, Integrated Definition Language), vb. gibi dillerdir.



Şekil 4.1: Yazılım geliştirmede model kullanımı.

Son yıllarda popülerliği artan yazılım geliştirme yaklaşımlarından biri Model Güdümlü Geliştirme (MDD, Model Driven Development) yaklaşımıdır. Yazılım geliştirme yaşam döngüsü içerisinde farklı modeller geliştirilerek yazılımdaki karmaşıklığın azaltılmasını ve kodun büyük bir kısmının otomatik olarak üretilmesini amaçlayan bu yaklaşım, modellerin daha etkin ve verimli kullanımını hedeflemektedir [9].

Yazılım alanında, 1 ve 0’lardan oluşan makine dilinden insanların daha rahat anlayabileceği programlama dillerine geçiş ile başlayan yazılımda soyutlama yaklaşımı model güdümlü geliştirmenin en önemli yapı taşlarından biridir. Kolay anlaşılabilirliğin yanı sıra daha az kod ile daha fazla işlem yapılabilmesi sağlanarak tasarımcılar için önemli kolaylıklar getirilmiştir [26].

Model güdümlü yaklaşımlar sayesinde, tasarımcılar herhangi bir programlama dili kullanmadan yazılım geliştirebilmektedir. Böylece alan uzmanları ve yazılım ekibi birlikte tasarımı oluşturmakta ve koda geçiş hızlı bir şekilde yapılabilmektedir [27]. En çok kullanılan model güdümlü yaklaşımlar model güdümlü mimari (MDA, Model

Driven Architecture) ve modelle bütünleşmiş hesaplama (MIC, Model Integrated Computing) yaklaşımlarıdır. Model güdümlü mimari 2001 yılında Nesne Yönetim Grubu (OMG, Object Management Group) tarafından tanımlanan bir yazılım geliştirme yaklaşımıdır. Model güdümlü mimari dört seviyeli bir modelleme yapısı önermektedir. 1. seviyede gerçek dünya ya da sistemde kullanılacak nesnelere karşılık gelen ifadeler bulunur. 2. seviyede bu nesnelere ve sistemi temsil eden modeller yer alır. 3. seviyede model tanımlayabilmek için gereken modelleme elemanları ve modelleme dili metamodellerle tanımlanır. 4. seviyede ise modelleme dilini tanımlayabilmek için gereken modelleme elemanları tanımlanır [28].

Model güdümlü mimari, soyutlama seviyesini artırarak model oluşturmayı ve bu modeller arasında dönüşümler sağlayarak hedeflenen sistemi elde etmeyi amaçlamaktadır. Temelde üç farklı soyutlama düzeyi tanımlanmıştır. Bunlar aşağıda listelenmiştir:

Hesaplama Bağımsız Model (CIM, Computation Independent Model):

Sistemin çalışmasını herkes tarafından kolaylıkla anlaşılacak şekilde açıklayan ve uygulamaya dönük bir bilgi olmadan nesnelere, nesnelere arasındaki ilişkileri ve ihtiyaçları belirleyen modeldir. CIM’de, geliştirilecek alan ile ilgili tanımlamalar ve gereksinim modelleri tanımlanır.

Platform Bağımsız Model (PIM, Platform Independent Model):

PIM, sistemin yapısını ve davranışlarını tanımlayan, herhangi bir platforma bağlı olmayan, problem odaklı çözüm üretmeyi amaçlayan bir modeldir. Sistemlerin UML diyagramlarıyla tasarımları bu sürece bir örnektir.

Platform Bağımlı Model (PSM, Platform Specific Model):

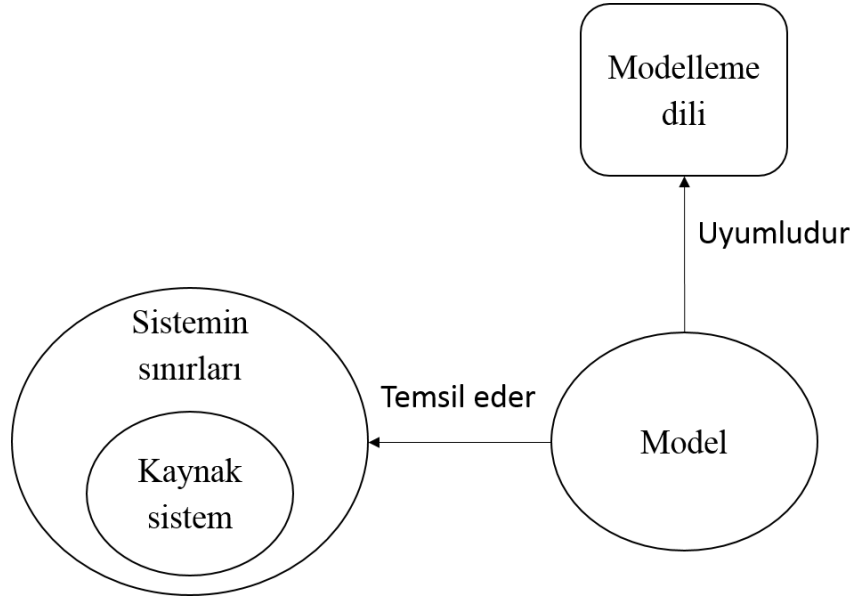
CIM ve PIM seviyesindeki modellerin kullanılması ile elde edilen ve belirli bir platform için tanımlanan geliştirme modelidir. Kullanılacak kavram, kısıt ve tanımlamalar önceki seviyelerde belirlendiği için model dönüşümleri ile elde edilebilir [29].

Modelle bütünleşmiş hesaplama yaklaşımı ise yine 2001 yılında Avrupa’da Vanderbilt Üniversitesi Yazılım Entegre Sistemler Enstitüsü (ISIS, Institute for Software Integrated Systems) tarafından geliştirilmiş ve uygulamaya dönük araç desteği de olan bir yaklaşımdır. Temelde model güdümlü mimari ile benzer yapıları bulundurmaktadır. Bu tez çalışmasında kullanılan metamodelleme aracı modelle

bütünleşmiş hesaplama yaklaşımına uygundur. Detaylar 5. ve 6. bölümde açıklanmıştır.

4.2. Modelleme ve Simülasyon

Modelleme ve simülasyon, sistemlerin analiz edilmesi ve tasarlanması sırasında kullanılan etkili bir yöntemdir [2]. Modelleme, bir önceki bölümde açıklandığı gibi bir sistemin temsili olarak ifade edilmesidir. Modelleme bir amaç için yapılmaktadır ve bu amaca uygun olarak sistemin bir bölümü ya da sistemdeki belirli kısımlar, olaylar ya da süreçler ele alınır. Temsil edilen sistem kaynak sistem olarak adlandırılır ve belirli sınırlarla çevrelenmiştir. Şekil 4.2’de model, modelleme dili ve sistem arasındaki ilişki gösterilmiştir [9, 30].



Şekil 4.2: Model, modelleme dili ve sistem arasındaki ilişki [30].

“Modelin zamana bağlı gerçekleştirilme metodu” olarak tanımlanan simülasyon ile modeldeki parametrelerin zamana göre değişiminin izlenebilmesi amaçlanmıştır [31]. Simülasyon, fiziksel, mantıksal, matematiksel ilişkiler kümesi şeklinde olabilecek modellerin farklı girdiler ile zamana bağlı değişimlerini bilgisayar ortamında izleyip, analizlerini yapabileceğimiz bir araçtır. Bir modelin gerçek uygulamasının yapılarak denenmesinin maliyetli olması ve her olay için tekrarlanması oldukça zaman alması nedeniyle simülasyon yöntemi kullanılır. Bu sayede risk, zaman ve maliyet bakımından tasarruf sağlanmış olur [32].

Son yıllarda modelleme ve simülasyon yöntemlerinin, hemen hemen her alanda getirdiği kolaylıklar ve karmaşık sistemlerin analizinde ve denenmesindeki başarıları ile tüm dünyada önemi artmıştır. Birçok ülkenin hükümetlerince de desteklenen modelleme ve simülasyon teknikleri askeri, ulaşım, moleküler biyoloji, genetik, uluslararası ilişkiler, sağlık hizmetleri gibi bir çok alanda yaygın olarak kullanılmaktadır [31].

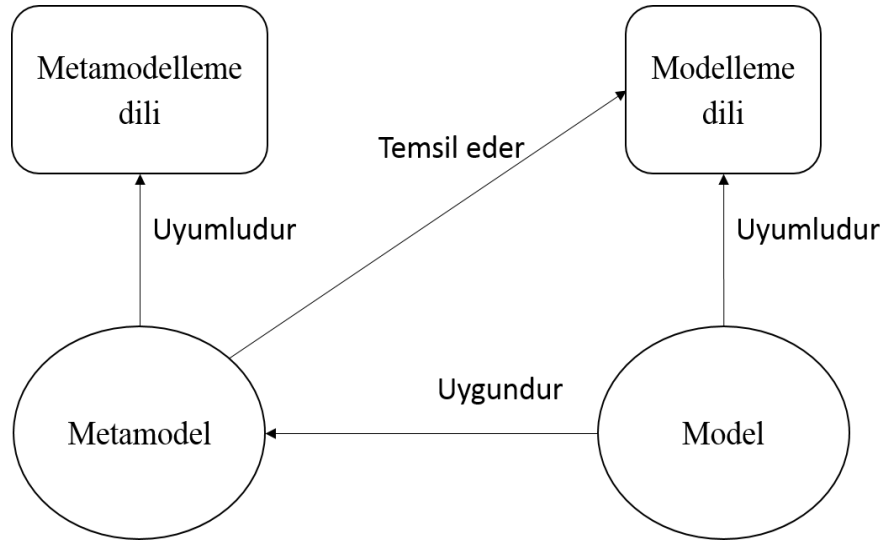
4.3. Alana Özgü Modelleme Dilleri ve Metamodelleme

Alana özgü modelleme yaklaşımı, daha detaylı ve alana özgü geliştirilmiş modelleme elemanları ile model geliştirmeyi amaçlayan bir yöntemdir. Alana özgü diller, sınırlı bir alanda, belirli bir problem türünü çözmek için geliştirilen dillerdir. Geliştirilen alana özgü dillerin görsel öğeleri kullanması alan uzmanlarının modelleri daha kolay anlamasına ve model geliştirme aşamalarında da bizzat yer almalarına olanak sağlamıştır.

Alana özgü modelleme dili geliştirmede UML eklenti mekanizması (profile) veya metamodelleme gibi yöntemler kullanılır [33, 34]. Bu tezde geliştirilen alana özgü dil için de metamodelleme yöntemi kullanılmıştır.

Metamodelleme, bir modelleme dilinin ya da yönteminin model olarak ifade edilmesidir. Geliştirilen bu üst modele metamodel denir. Bilgisayar mühendisliğinde, metamodel terimi bir modelleme dili ile geliştirilebilecek modellerin sınırlarının belirlendiği, modellerde kullanılan kavram tanımlarının yapıldığı, kavramların özelliklerinin belirlendiği ve kapsamının tanımlandığı modele karşılık gelir.

Bir metamodel aynı modellerde olduğu gibi, bir modelleme dili ile geliştirilir. Bu modelleme diline, metamodelleme dili denir. Metamodelleme dili de aslında bir metamodel olarak ifade edilebilir, bu durumda buna da meta-metamodel denir [9, 10, 35]. Şekil 4.3'te model ve metamodel arasındaki ilişki gösterilmiştir.

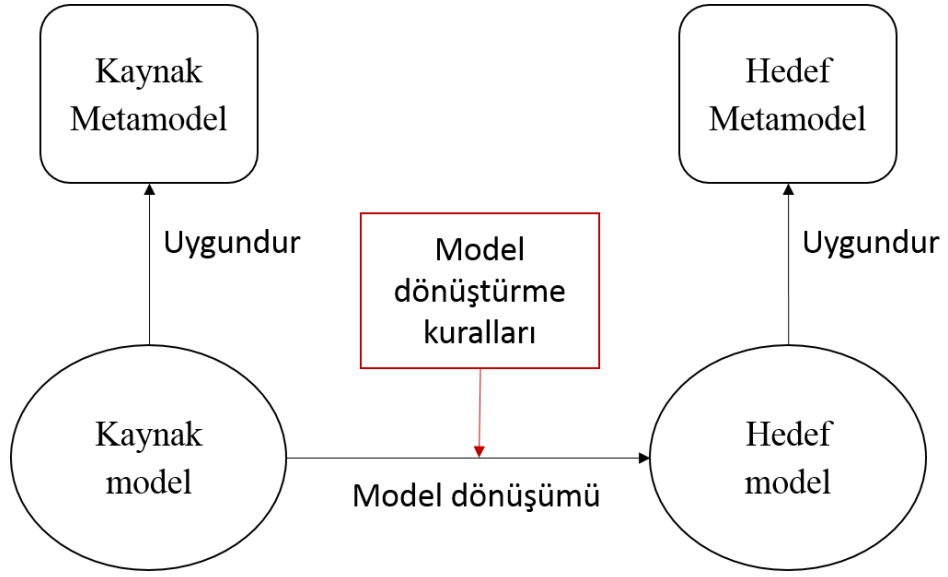


Şekil 4.3: Metamodelleme [36].

Metamodelleme yönteminin kullanımında araç desteği çok önemlidir. Metamodellerin çizilebildiği ve modelleme amacı ile kullanılabilirdiği birçok yazılım aracı vardır. En popüler olanları, Eclipse Modelleme Altyapısı(EMF, Eclipse Modeling Framework), MetaEdit+, Genel Modelleme Ortamı (GME, Generic Modeling Environment), Genel Eclipse Modelleme Sistemi (GEMS, Generic Eclipse Modeling System), Melanie, Kermeta, gibi araçlardır [37]. Bu araçlarda kullanılan metamodelleme dilleri, Meta Nesne Vasıtası (MOF, Meta Object Facility), Eclipse Ecore, Kermeta KM3, MetaGME, Essential MOF (EMOF), vb. gibi dillerdir. En çok kullanılan MOF, OMG tarafından geliştirilmiş bir nesne tanımlama standardıdır.

4.4. Model Dönüşümü ve Model Dönüşüm Dilleri

Model dönüşümü, en genel anlamı ile bir modelden başka bir modele bazı dönüşüm kuralları uygulayarak geçmek anlamına gelir. Model güdümlü geliştirme ve yazılım mühendisliği açısından ise, metamodelleri kullanarak yapılan daha biçimsel dönüşümler model dönüşümlerini ifade eder. Model dönüşümlerinde kaynak ve hedef metamodeller ile kaynak ve hedef modellerden söz edilir. [38, 39]. Şekil 4.4'te gösterildiği şekilde, bir model dönüşüm dili ile yazılmış model dönüşüm kuralları uygulanarak, kaynak modelden hedef model otomatik olarak üretilir.



Şekil 4.4: Model dönüşümleri [30].

En çok bilinen model dönüşüm dilleri modelle bütünleşmiş hesaplama yapısını kullanan Grafik Yeniden Yazma ve Dönüştürme Dili (GREAT, Graph Rewriting and Transformation Language), MOF Sorgula/Görüntüle/Dönüştür dili (QVT, Query/View/Transformation), Atlas Dönüşüm Dili (ATL, Atlas Transformation Language), Xtend gibi dillerdir.

BEŞİNCİ BÖLÜM

TERMİNAL MODELLEMESİ İÇİN ÖNERİLEN METAMODEL

Bu bölümde havalimanı terminal modellemesi için önerilen metamodel açıklanacaktır. Önerilen metamodel ile havalimanı terminal binalarının yerleşim planları çizilebilecek ve yolcuların terminaldeki akış diyagramları tanımlanabilecektir. Bu metamodel kullanılarak model güdümlü geliştirme araçları sayesinde otomatik bir modelleme editörü de geliştirilmiştir. Metamodelin ve modelleme editörünün potansiyel kullanıcıları mimarlar, havalimanı mühendisleri, endüstri mühendisleri ve yazılım mühendisleri gibi havalimanı tasarımının çeşitli aşamalarında çalışan kişiler olabilir.

Bu tezde sunulan çalışma iki aşamadan oluşmaktadır. Bunlar metamodelleme ve modelleme kısımlarıdır. Birinci aşama olan metamodelleme iki bölümden oluşmaktadır. Birinci bölümde havalimanı terminal binası yerleşim planı için bir metamodel bulunmaktadır. İkinci bölümde terminal yolcu akış şeması modelinin oluşturulabileceği bir metamodel bulunmaktadır. Modelleme aşamasında ise, metamodelle uyumlu olarak bir model editörü kullanılarak modeller geliştirilmesi sağlanmıştır. Bu bölümde önce metamodelleme ortamı, sonra da metamodel açıklanmıştır. Modelleme editörü altıncı bölümde açıklanmıştır.

5.1. Metamodelleme Ortamı

Önerilen metamodelin geliştirilmesinde GME programı kullanılmıştır. GME Vanderbilt Üniversitesi tarafından akademik ortamda geliştirilmiş, teknik desteği olan, kullanımı kolay ve ücretsiz bir programdır. On yıldan uzun süredir kullanılması, düzenli olarak güncellenmesi, performans açısından çok sayıda sınıf için sorunsuz bir şekilde çalışması, görsel öğelerin istenilen bir şekilde kullanılabilmesi gibi nedenlerle en uygun metamodelleme ortamı olarak GME programı uygun bulunmuştur.

GME programı seçilmeden önce alternatif araçlardan bazıları denenmiştir, ancak MetaEdit gibi ücretli uygulamalar değerlendirmeye alınmamıştır. Denenen araçlardan Melanie’de sınıf sayıları arttığında program performansında aksamalar olmuştur. GEMS aracının kullanımı ise GME’ye göre daha zor bulunmuştur.

GME programı ile modelleme elemanlarını ve kurallarını tasarımcının belirlediği metamodeller oluşturulabilir. Metamodeller, MetaGME meta-metamodeline uygun olarak GME’nin sunduğu metamodelleme ortamı sayesinde kolaylıkla tanımlanabilir.

GME menü kullanımları kolay ve yapı olarak karmaşık olmayan bir programdır. Bunun yanında C, C++, Java gibi yüksek seviyeli dillerle ve UML gibi genel amaçlı modelleme dilleriyle de uyumlu çalışmaktadır. Bu uyumluluk sayesinde, GME ortamında geliştirilen metamodellere C++ ve Java gibi programlama dilleri ile yazılmış bileşenler sistem aracı olarak eklenebilmektedir. Harici olarak eklenebilen bu araçlar ile GME, sade yapısına rağmen daha geniş fonksiyonlara sahip olabilmektedir. Ayrıca, farklı dilleri destekliyor olması sayesinde daha çok kullanıcıya hitap edilmesini sağlamıştır.

Metamodel oluşturulurken MetaGME metamodelinin UML class diyagramlarına benzer MOF tabanlı modelleme elemanları kullanılmıştır. Model elemanlarının oluşturulması veya birbirleri arasında bağlantı kurulması belli kurallar çerçevesinde yapılabilmektedir. Bu kurallar GME programının kendi özellikleri ile oluşturulabileceği gibi Nesne Kısıtlama Dili (OCL, Object Constraint Language) kullanılarak da kapsamlı bir şekilde tanımlanabilmektedir [40]. OCL metinsel bir dildir [41]. Metamodelimizde de OCL kullanılarak bazı model elemanları için kısıtlamalar (constraints) tanımlanmıştır. Bunlardan bazıları örnek olması açısından ilerleyen bölümlerde açıklanmıştır.

5.2. Metamodel Hakkında Genel Bilgiler

Geliştirilen metamodelde havalimanı terminal binaları için IATA, ICAO gibi uluslararası kuruluşlar tarafından belirlenen ve kabul görmüş standartlar dikkate alınarak bir modelleme dili tasarlanmıştır. Mimarlar ya da havalimanı mühendisleri metamodelle belirlenen kurallar, kısıtlamalar ve tanımlamalar çerçevesinde havalimanı terminal binası modellemesi yapabileceklerdir. Geliştirilen modelde istenmeyen kısıtlamalar varsa ya da farklı birim ve modellerin fonksiyonel olarak

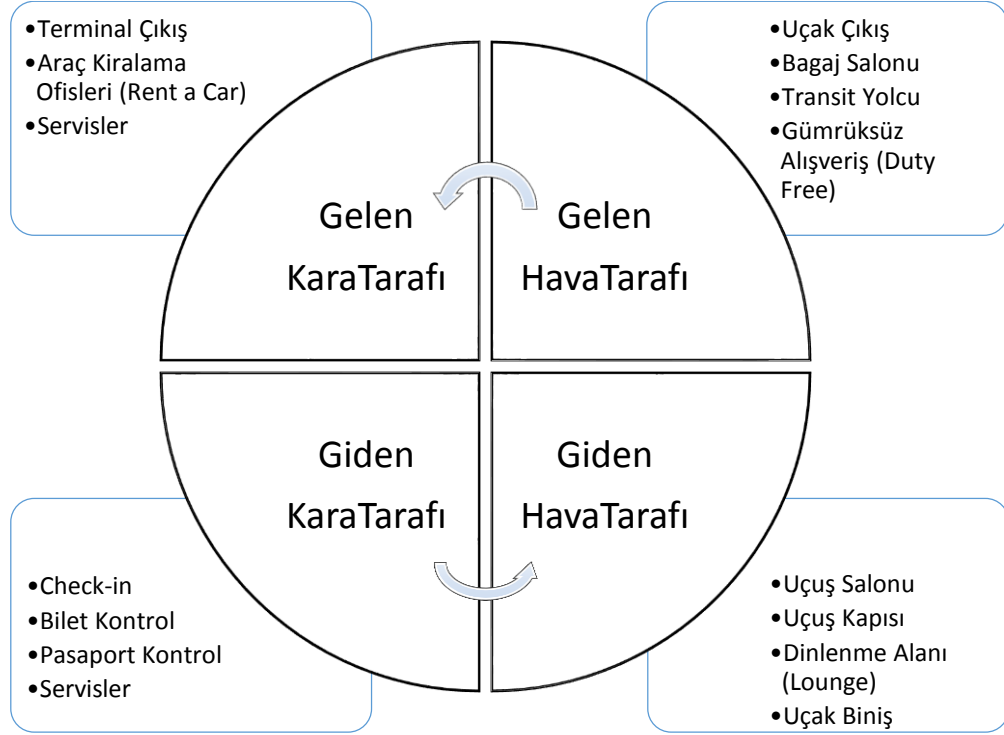
eklenmesi gerekirse, yazılım mühendisleri tarafından metamodelle müdahale edilerek gerekli güncellemeler yapılabilir.

Her boyuttaki havalimanının tasarımında ve mevcut havalimanlarının modernizasyonunda kullanılabilecek olan bu metamodelde, uluslararası standartları modele doğru bir şekilde uygulanabilmesi ve dünyada artık daha çok dikkat edilen engelsiz havalimanı projeleri gibi diğer hassasiyetlerin tasarıma doğru bir şekilde yansıtılabilmesi sağlanmıştır. Bu bağlamda standartlarda belirtilen minimum birimler metamodelle eklenmesinin yanında, kat farkı olması durumunda engelliler için platform ya da asansörlerin eklenmesi zorunluluğu gibi bazı kurallar ve kısıtlamalar ile modeli geliştiren kullanıcıların daha verimli ve doğru tasarımlar geliştirmesinin sağlanması amaçlanmıştır.

Daha doğru tasarımlar oluşturabilmek için modelleme çalışmasında ölçeklendirme işlemi yapılmıştır. GME programı ile geliştirilen metamodel kullanılarak örnek çizimi yapılan Sabiha Gökçen Havalimanı terminal binası 200.000 metrekare alana kuruludur. Modelleme çalışmasında bu bina yaklaşık olarak 1/2.000.000 oranında küçültülerek çizilebilmiştir. GME programında alan büyüklüklerinin piksel olarak hesaplanmasından dolayı birimler piksel cinsinden boyutlandırılmıştır. GME programının kullanımı sırasında karşılaşılan en önemli zorluk, modelleme elemanlarının konum bilgisine doğrudan erişilememesi olmuştur. Bu nedenle terminaldeki birimlerin boyutları ve birimler arası uzaklıkların hassas ve otomatik bir şekilde kontrolü sağlanamamıştır. Ancak, piksel olarak alan büyüklüğü bilgisine erişilebildiği için bu kontroller özel olarak yapılabilmektedir. Bu nedenle, terminaldeki konuma ve uzaklığa bağlı kısıtlamalar bu tezin kapsamı dışında tutulmuştur. İşeriye yapılacak çalışmalarda, istenirse konum bilgisine C++ veya Java gibi bir dille eklenti yazılarak erişmek ve buna bağlı olarak model üzerinde kısıtlamalar tanımlamak mümkündür.

Metamodelde havalimanı terminali ile ilgili isimler kullanılırken sivil havacılık genel müdürlüğünün uluslararası standartlar dikkate alınarak hazırlanan Havaalanı Planlama Kılavuzundaki terimler esas alınmıştır. Bu bağlamda metamodelde terminal binası Şekil 5.1’de gösterildiği gibi iki ana bölüm olarak ele alınmıştır. Pasaport veya bilet kontrol noktalarından geçtikten sonra sadece biletli yolcuların uçağa binmeden önce kullandığı alan havalimanı terminalinin *Hava Tarafı* olarak isimlendirilmiştir. Terminal içinde bileti olmayan kişilerin de kullanabildiği, terminal girişi ile pasaport

veya bilet kontrol noktaları arasında kalan alan da havalimanı terminalinin *Kara Tarafı* olarak isimlendirilmiştir. Hava ve kara tarafının her iki bölümü de gelen ve giden yolcu bölümleri olmak üzere ikiye ayrılmıştır.



Şekil 5.1: Terminal metamodeli tasarım şeması.

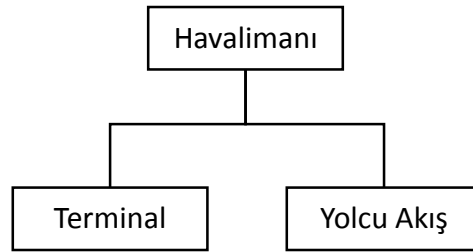
Aynı zamanda giden ve gelen bölümleri de, iç ve dış hatlar olarak bir alt gruba ayrılabilir. Metamodelin bu şekilde tasarımı ile, kara tarafında *gümrüksüz alışveriş (dutyfree)*, *dinlenme alanı (lounge)*, *uçuş bekleme salonları gibi alanların (flighthall)* olması engellenmiştir. Aynı şekilde, hava tarafında da *check-in* bankoları ve *araç kiralama (rentacar)* ofisleri gibi alanların olması engellenmiştir. Böylece terminal planında hatalı yerleşimler kontrol edilmiştir. Metamodel içine yerleştirilen kurallar ile her terminalde olması gereken check-in bankosu, terminal giriş-çıkışları, güvenlik birimleri gibi ve dış hatlar var ise pasaport kontrol noktasının olması gibi kontrollerin de yapılması sağlanmıştır. Akış şeması metamodeli oluşturulurken de bağlantılar yapılırken uçakla gelen bir yolcu doğrudan check-in bankosuna geçemeyeceği için bu tür bağlantılar sınırlandırılmıştır.

Metamodelde birimler, uluslararası kullanımları her havalimanında İngilizceleri ile beraber verildiği için İngilizce olarak isimlendirilmiştir. GME aracının kullanımı sırasında karşılaşılan bir başka zorluk da Türkçe karakter kullanımı olmuştur. Türkçe karakterlerin sınıf isimlerinde kullanılamaması nedeniyle, sınıflar genelde İngilizce olarak adlandırılmıştır.

Metamodeldeki *Terminal* ve *Yolcu Akış (PassengerFlow)* modelinin İngilizce isimlendirilen birimlerinin metamodel ile modelleme kullanıcı ara yüzündeki isimleri ve karşılıkları Ek-A Tablo A.1’de verilmiştir. Model düzenleyicisinde görünmeyen ancak metamodelde tanımlanan İngilizce kısıt, değişken, atom veya model isimleri ise Ek-A Tablo A.2’de ayrı olarak verilmiştir.

5.3. Havalimanı Terminal Modellemesi için Önerilen Metamodel

Önerilen metamodel Şekil 5.2’de gösterildiği gibi *Havalimanı* olarak adlandırılan bir en üst sınıf (buna kök sınıf da denir) ve bunu oluşturan *Terminal* ve *Yolcu Akış* ana sınıflarından oluşmaktadır. Terminal kısmı terminal metamodeli, yolcu akış kısmı yolcu akış metamodeli olarak adlandırılacaktır. Aslında, bu ikisi birlikte Havalimanı metamodelini oluşturmaktadır.

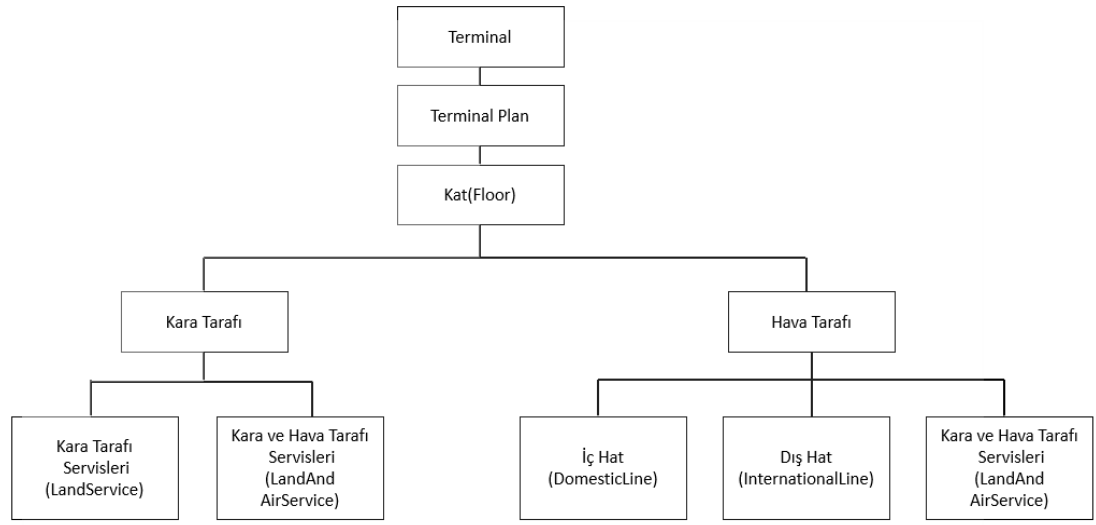


Şekil 5.2: Havalimanı terminal metamodeli ana şeması.

Terminal metamodeli küçük, orta, büyük ölçekli havalimanı terminal binalarında bulunabilecek birimleri içinde barındırmaktadır. *Terminal* metamodeli ile kullanıcılara, metamodelin kısıt ve kuralları çerçevesinde, terminal binası yerleşim planını oluşturabilecekleri bir yöntem sunulmaktadır. *Yolcu akış* metamodeli ile de kullanıcılara, metamodel’in kısıt ve kuralları çerçevesinde, terminal binası giriş-çıkış kapısıyla uçağa biniş-iniş kapısı arasında yolcunun terminal içinde uğrayabileceği birimlerin, sıralamasını ve akışını tasarlayacağı bir yöntem sunulmaktadır.

5.3.1. Terminal

Terminal metamodeli ile kullanıcı, havalimanı terminal binasının modellemesini yapabilecektir. Terminal binasının tanımı ve kapsamı uluslararası standartlarda, “yolcunun uçağa biniş noktasından başlayıp, yer ulaşımı ile terminal binasının bulunduğu yer arasında kalan, uçuşlar arası geçişlerde dahil olmak üzere tüm aktivite ve birimlerdir” şeklinde belirtilmektedir [42]. Şekil 5.3’te *Terminal* metamodelinin ana hatları şema ile gösterilmiştir. Şekil 5.4’te ise *Terminal* metamodelinin sınıf diyagramı gösterilmiştir. Aşağıda alt sınıflar detaylı olarak açıklanmıştır.

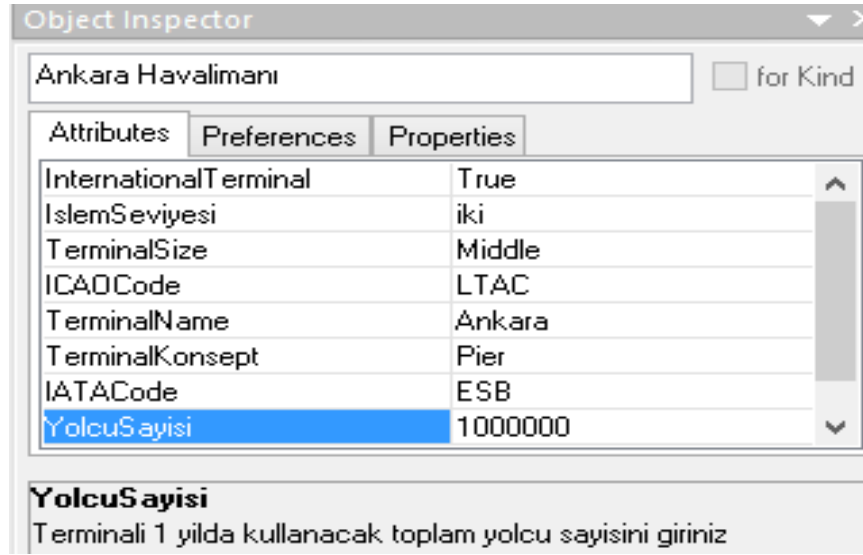


Şekil 5.3: Terminal metamodeli şeması.

Terminal Plan: *TerminalPlan* modeli havalimanı terminali için eklenmesi gereken ilk modeldir. Terminal binasında bulunan katlar ve tüm birimler bu model içine yerleştirilecektir. *TerminalPlan* modeli eklendikten sonra Şekil 5.5’te değişkenleri (*attributes*) gösterildiği gibi terminalin ismi, uluslararası havalimanı olup olmadığı, işlem seviyesi (terminal binasında geliş ve gidişlerin aynı kat seviyesinde yapılması halinde tek, dikey olarak farklı kat seviyelerinde yapılması halinde çift, hava tarafı ya da kara tarafı cephesinden biri dikey ise $1/2$ (tek-çift) seviye olarak seçilebilir), terminal boyutu (küçük, orta, büyük), ICAO havalimanı terminal kodu, terminal konsepti (pier, uydu, açık apron, linear, kompakt modül, diğer), IATA havalimanı terminal kodu, yıllık yolcu sayısı gibi bilgilerin girilmesi gerekmektedir [24]. Böylece terminalin özellikleri modelin ilk oluşturulma aşamasında belirlenmektedir. Şekil 5.6’da ise metamodel kullanılarak elde edilen model editöründeki *TerminalPlan* modeli değişkenlerine ait özellikler penceresi gösterilmiştir.

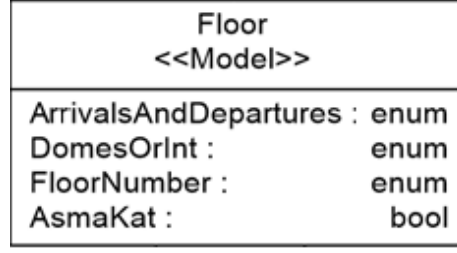
TerminalPlan <<Model>>	
InternationalTerminal :	bool
IslemSeviyesi :	enum
TerminalSize :	enum
ICAOCode :	field
TerminalName :	field
TerminalKonsept :	enum
IATACode :	field
YolcuSayisi :	field

Şekil 5.5: TerminalPlan modeli.

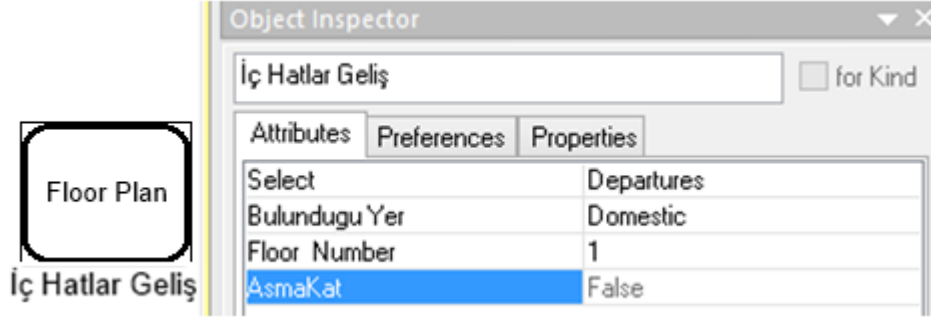


Şekil 5.6: TerminalPlan modeli tasarım ekranı değişken penceresi.

Kat (Floor): *Kat* modeli *TerminalPlan* modelinin içinde bulunan tek modeldir. Yani terminal modeli oluşturulduktan sonra ilk olarak terminalin kaç katlı olacağı belirlenmelidir. Kat sayısı kadar *kat* modelinden oluşturulması gerekir. Bu modelde Şekil 5.7 de gösterilen geliş-gidiş seçimi (geliş, gidiş ya da ortak kullanım olacak ise geliş-gidiş şeklinde), iç hat ya da dış hat seçimi, kat numarası ve asma kat olup olmadığı seçimi için değişkenler bulunmaktadır. Şekil 5.8’de model editöründeki kat modeline ait özellikler penceresi gösterilmiştir.

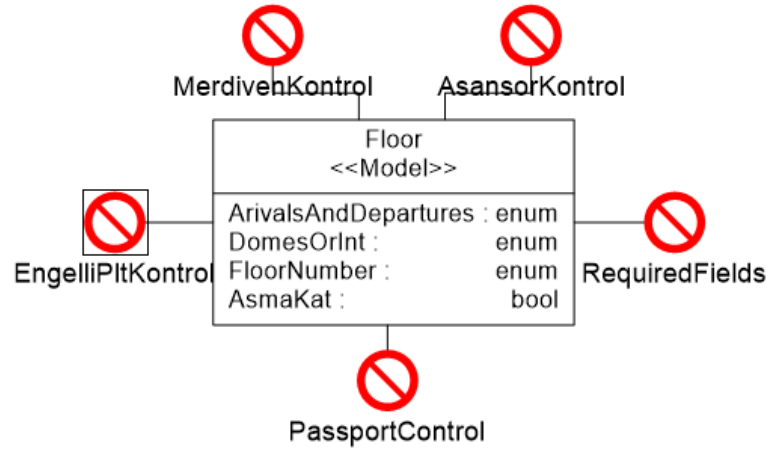


Şekil 5.7: Kat modeli.



Şekil 5.8: Kat modeli tasarım ekranı.

Kat modelinde Şekil 5.9’da gösterildiği gibi, *merdiven kontrol*, *asansör kontrol*, *engelli platform kontrol*, *pasaport kontrol* (*passportcontrol*), *gerekli alan* (*required fields*) kontrol olmak üzere beş adet kısıtlama tanımlanmıştır.



Şekil 5.9: Kat modelinde tanımlı kısıtlamalar.

Merdiven Kontrol: Şekil 5.10’da kodları verilen *merdiven kontrol* ile birden fazla katın varlığı kontrol edilerek, kat sayısının birden fazla olması halinde model içinde bir yürüyen merdiven yerleştirilmesini zorunlu kılmaktadır.

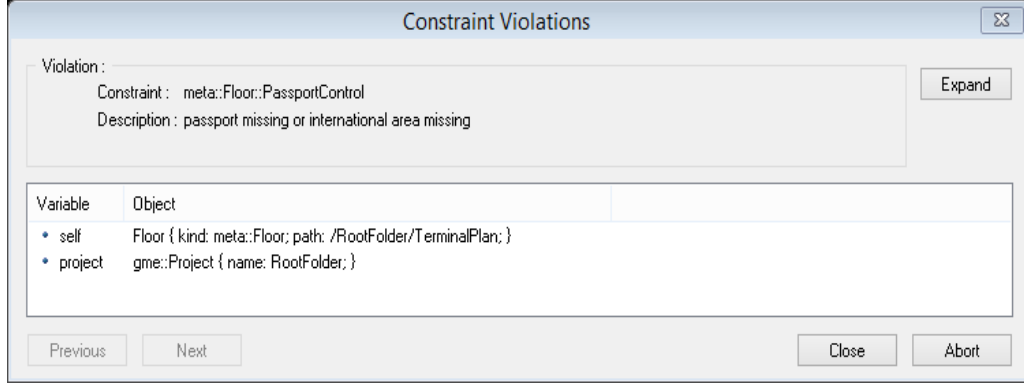
```
if project.allInstancesOf( Floor )->size>1 then
  if project.allInstancesOf( Elevator )->size>0 then
    true
  else false
  endif
else true
endif
```

Şekil 5.10: Merdiven kontrolü OCL kodu.

Pasaport kontrol: Şekil 5.11’de verilen OCL kodları ile *dış hat (InternationalLine)* modelinin eklenip eklenmediği kontrol edilerek, eklenmesi halinde *pasaport kontrol* atomunun modelde bulunmasını zorunlu kılar. Eğer *pasaport kontrol* alanı eklenmiş ise Şekil 5.12’de olduğu gibi dış hatlar modelinin de *kat* modelinde bulunması gerektiğinin uyarısını verir.

```
if self.parts( InternationalLine ) ->size > 0 then
  if self.parts( Passport ) ->size > 0 then true
  else false
  endif
else
  if self.parts( InternationalLine) ->size < 1 then
    if self.parts( Passport ) ->size > 0 then false
    else true
    endif
  else true
  endif
endif
```

Şekil 5.11: Pasaport kontrolü OCL kodu.



Şekil 5.12: Dış hatlar terminali bulunduğu halde pasaport kontrol alanının bulunmadığını gösteren tasarım ekranı uyarı penceresi.

Gerekli Alan: Havalimanının büyüklüğüne bakılmaksızın her havalimanında bulunması gereken *check-in* bankosu, *güvenlik (security)* gibi alanların Şekil 5.13'te OCL kodları ile kontrolünün yapılmasını sağlar.

```
self.parts(CheckBanko)->size > 0
and
self.parts(Sequity)->size > 0 and
self.parts(Wc)->size > 0 and
self.parts(XRay)->size > 0 and
self.parts(Information)->size > 0
```

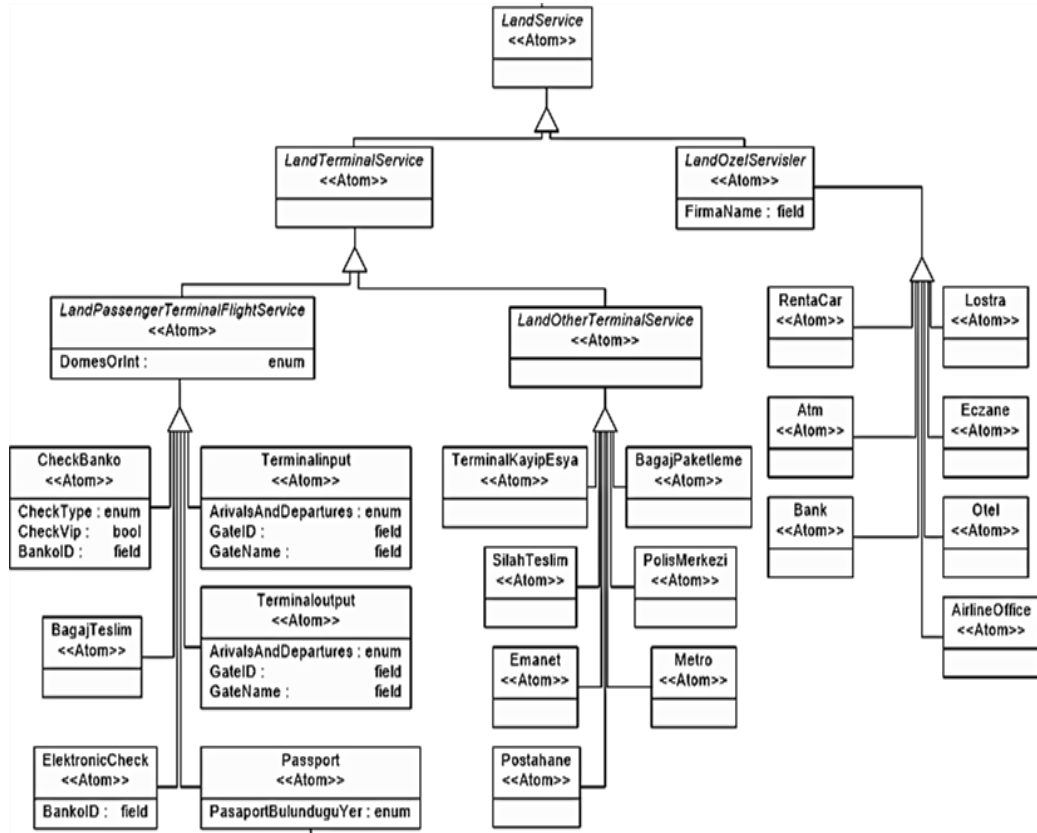
Şekil 5.13: Gerekli alan kontrolü OCL kodu.

Merdiven Kontrol ve *EngelliPltKontrol* kısıtlamalarıyla da kat farkı olması halinde yürüyen merdiven ve engelli platform kontrolünün yapılması sağlanmaktadır.

Kat modeli iki kısımdan oluşmaktadır: biletli yolcuların, yolcu yakınlarının ve tüm halkın kullanımına açık olan bölümler *Kara Tarafı*, sadece biletli yolcuların kullanabileceği bölümler ise *Hava Tarafı* olarak gruplandırılmıştır.

Kara Tarafı: *Kara Tarafı* atomu, sadece kara tarafında bulunan *Kara Tarafı Servisleri (LandService)* ve hem kara hem de hava tarafında bulunan *Kara ve Hava*

Tarafı Servisleri (LandAirService) olarak gruplandırılmıştır. Şekil 5.14'te *Kara Tarafı servisleri* atomunun sınıf diyagramı gösterilmiştir.



Şekil 5.14: Kara tarafı servisleri atomu.

Kara Tarafı Servisleri atomu *terminal servisleri* ve *özel servisler* olmak üzere ikiye ayrılmıştır.

Kara Tarafı Özel Servisler (LandOzelServisler) sınıfı altında, terminallerde bulunan firmalara ait *Araç Kiralama*, *Lostra*, *Atm*, *Eczane*, *Banka*, *Otel*, *Uçuş Firması Ofisleri* bulunmaktadır. Tasarımda yerleştirilen her özel servis için firma adının girilmesini sağlayacak *Firma İsmi* değişkeni tanımlanmıştır.

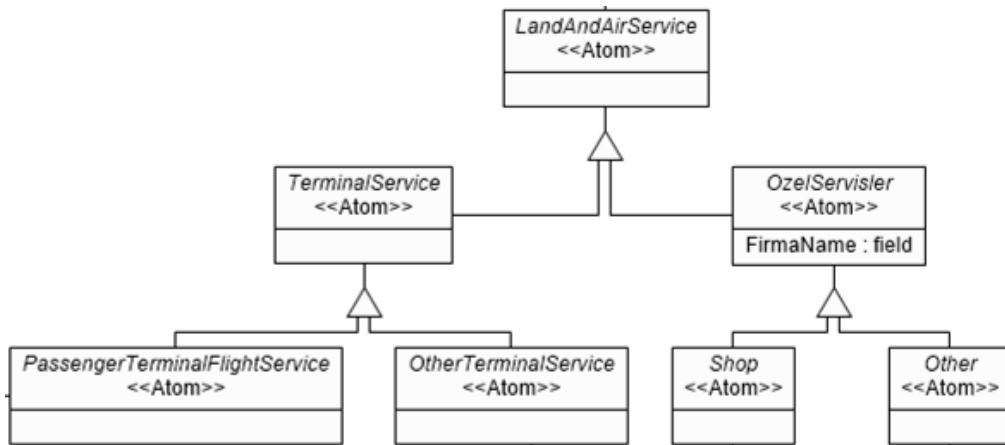
Kara Tarafı Terminal Servisleri (LandTerminalService) sınıfı altında; *TerminalKayipEsvya*, *BagajPaketleme*, *PolisMerkezi*, *Metro*, *SilahTeslim*, *Emanet*, *Postahane* alanları tanımlanmıştır. Bu atom altında tanımlı diğer alanlar ve değişkenler Tablo 5.1'de verilmiştir.

Tablo 5.1: Kara tarafı servisleri.

Birim Adı	Tanımlanan Değişkenler
Terminal Giriş	İç-Dış Hat, Geliş-Gidiş, Gate Id (kapı numarası), Gate Name(kapı ismi) bilgileri
Terminal Çıkış	
Check-in	İç-Dış Hat, Check Type(İç-Dış hat), VIP, Banko numarası bilgileri
Pasaport	İç-Dış Hat
ElektronikCheck	İç-Dış Hat, Banko numarası

Check-in bankosuna verilecek ismin iki karakteri geçmemesi için (C1, C2 vb.) “*self.name.size()<3*” şeklinde bir kısıtlama eklenmiştir.

Kara ve Hava Tarafı Servisleri (LandAndAirService) atomu, terminalin hem hava hem de kara tarafında bulunabilecek birimler, *Terminal Servisleri* ile *Özel Servisler* olmak üzere ikiye ayrılmıştır. Bu atom Şekil 5.15’te gösterilmiştir.



Şekil 5.15: Kara ve hava tarafı servisleri atomu.

Şekil 5.15’te görüldüğü gibi birimler özelliklerine göre gruplandırılarak tasarlanmıştır. Bu gruplandırma işlemleri modelleme esnasında kullanıcının

havalimanının özelliğine göre birim eklemek istemesi halinde metamodele müdahale ederken güncelleme işlemlerinin daha rahat ve doğru bir şekilde yapılabilmesini sağlayacaktır.

Terminal Servisleri (TerminalService) atomu altında terminal işletmesinin yolcularına sunduğu hizmet birimleri bulunmaktadır.

Bu doğrultuda *Yolcu Terminal Uçuş Servisleri (PassengerTerminalFlightService)* alt sınıfında, yolcuların uçuşa yönelik kullanabilecekleri hem hava hem de kara tarafında bulunabilecek *Genel Kapı (Gate)*, *XRy*, *Bilet Kontrol* terminal hizmet birimleri bulunmaktadır. *Genel kapı* ile hava tarafı ve kara tarafı arası geçişlerde bulunan kapılar temsil edilmektedir.

Diğer Terminal Servisleri (OtherTerminalService) atomu altında ise hem hava hem de kara tarafında bulunabilecek uçuşa yönelik terminal hizmetleri dışındaki birimler bulunmaktadır. Bu birimler aşağıda listelenmiştir:

Terminal Hizmetleri atomu altında; *Çocuk Oyun Alanı (ChildrenGameArea)*, *Bagaj Taşıma Arabası (BaggageTrolleys)*, *Tekerlekli Sandalye Servisi (WheelChairServis)*, *Çocuk Bakım Odası (WcAndChildcare)*, *Danışma (Information)*, *Wc Engelli, Güvenlik, Wc;*

Alan ve Salonlar (AreaAndHalls) atomu altında; *Vip Salonu (VipHall)*, *PilotDinlenmeSalonu*, *Serbest Alan (FreeSpace)*, *Konferans Salonu (ConferenceHall)*, *İbadet Yerleri (Worship)*, tasarımda farklı alanlar oluşturmak için *YatayCizgi, DikeyCizgi;*

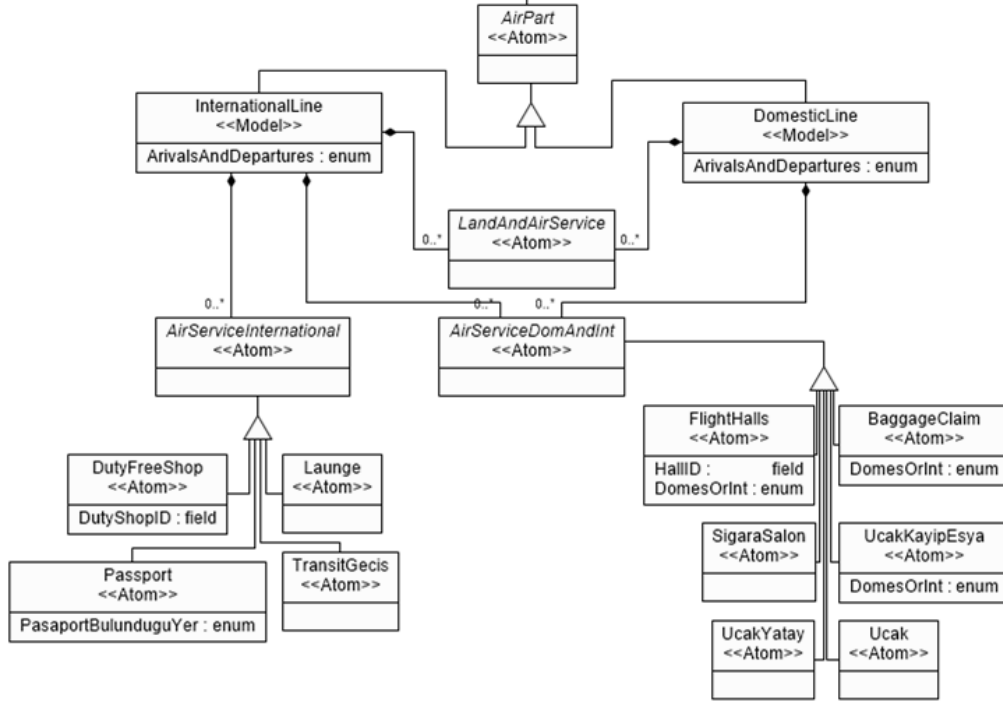
Ücretli Hizmetler atomu altında; *Döviz Bürosu (ChangeOffice)*, *Vergi İndirimi (TaxFree)*, *Telefon;*

Konum Değiştirme atomu altında; *Yürüyen Merdiven (Escalator)*, *Asansör (Elevator)*, *Merdiven (Stairs)*, *Engelli Platformu (EngelliPlt)*, *Yürüyen Yol (WalkWay);*

gibi birimler gruplandırılmıştır.

Özel Servisler atomu altında terminal işletmesi dışındaki, hem hava hem de kara tarafında bulunabilecek özel firmaların havaalanlarında vermiş olduğu hizmetler gruplandırılmıştır. *Kitapevi, Hediyelik Mağaza, Cafe-Restorant, Diğer Mağaza (OtherShop)*, ve *Gazete alanları Mağaza (Shop)* atomu altında; *Duş salonları (ShowerBath)*, *Kuru Temizleme, Kuaför, Spa*, alanları ise *Diğer (Other)* atomu altında kalıtım yoluyla yerleştirilmiştir.

Hava Tarafı (AirPart): *Hava Tarafı* atomunda Şekil 5.16’da gösterildiği gibi, kullanıcıların hava tarafı tasarımlarını gerçekleştirebilecekleri *İç Hat* modeli ile *Dış Hat* modeli olmak üzere iki adet birim bulunmaktadır. Tasarım ekranında hava tarafı bölümünün iç tasarımı, düzen açısından ve kısıtlamaların daha doğru uygulanabilmesi için *Kat* modeli içinde bir model olarak tasarlanmasının daha uygun olacağı düşünülmüştür.



Şekil 5.16: Hava tarafı atomu metamodeli.

İç Hat modelinde *Kara ve Hava Tarafı Servisleri* atom grubu ile *Hava Tarafı Servisleri* atomu altında tanımlı terminalin hava tarafında bulunabilecek *Uçuş Salonu* (*FlightHalls*), *Bagaj Alım*, *Sigara Salonu*, *Ucak Kayıp Eşya*, *Ucak* ve *UcakYatay* (uçak resmi) alanları eklenebilmektedir.

Dış Hat modelinde ise *Kara ve Hava Tarafı Servisleri* atom grubu ile sadece terminalin dış hat hava tarafında bulunabilecek *Hava Tarafı Servisleri Dış Hat* (*AirServiceInternational*) atomunda tanımlı *Gümrüksüz Alışveriş* (*DutyFreeShop*), *Dinlenme Alanı*, *Pasaport*, *Transit Geçiş* alanları tanımlanmıştır.

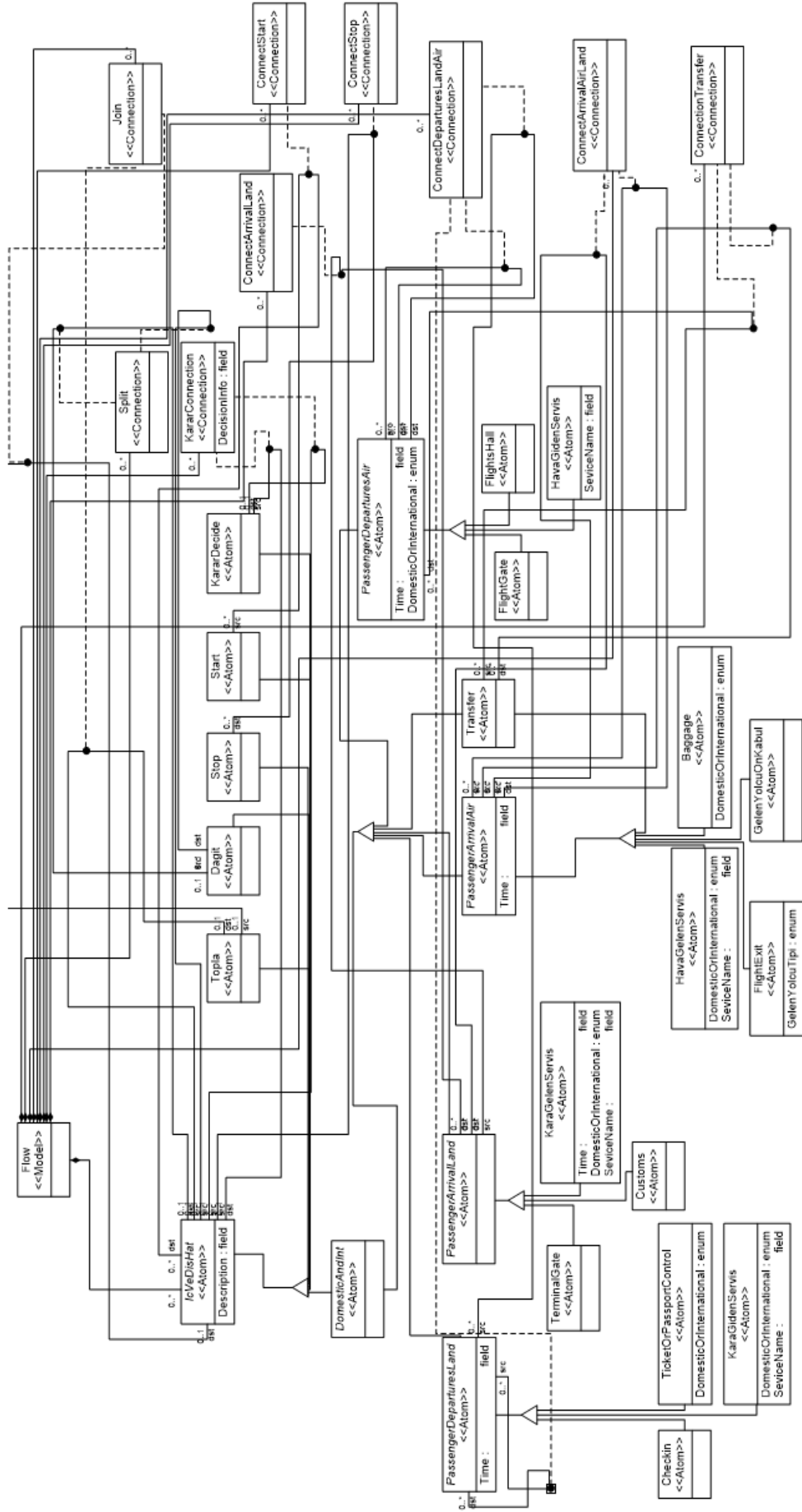
5.3.2. Yolcu Akış

Sınıf diyagramı Şekil 5.17’de verilen bu metamodel ile kullanıcılar, tasarımını yaptıkları terminal binasının yolcu akış şemalarını oluşturabileceklerdir. Bu metamodelde, havalimanında yolcuların kullanabilecekleri birimler belli başlıklar altında toplanarak tasarımı yapılmıştır. Gruplandırılan atomlar altında tanımlanan değişken seçenekleri kullanılarak genel olarak ifade edilen birimlerin neler oldukları tanımlanabilmektedir.

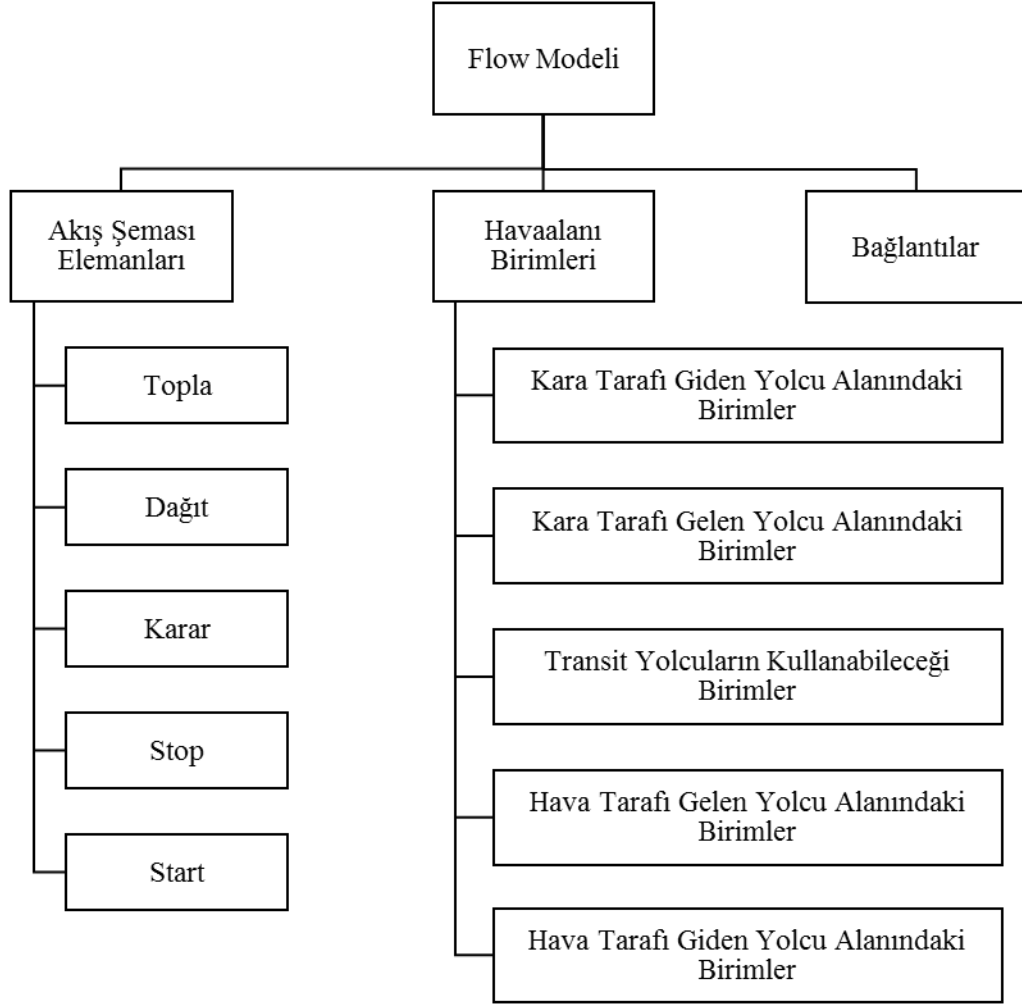
Bunun yanı sıra simülasyon modeli tasarlanırken önemli verilerden bir tanesi olan, birimlerde geçen zaman değişkeni olarak akış şemasında kullanılan havalimanı elemanlarına eklenmiştir. Yolcu akış şeması oluşturulurken birimlerde geçen zaman sabit ya da değişken bir sayı ifadesi olarak *Zaman (Time)* değişkeni alanına girilebilmektedir. Zaman değişkeni simülasyonda sık kullanılan dağılım fonksiyonları parametreleri ile birlikte girilebilir. Örneğin, sürekli dağılım (uniform distribution) minimum ve maksimum değerleri ya da normal dağılım (normal distribution) ortalama ve varyans değerleri ile tanımlanabilir.

Şekil 5.18’de ana şeması verilen *Yolcu Akış* metamodelinde kullanılabilecek birimlerin tamamı *Akış* modeli içinde tanımlanmıştır. Model içindeki birimler işlevsel olarak Akış Şeması Elemanları, Havalimanı Birimleri, Bağlantılar (Connection) olmak üzere üç farklı gruba ayrılabilir. Oluşturulan bağlantılar için kısıtlama tanımlamaları da gereken yerlerde yapılmıştır.

Bu metamodel ile geliştirilen modeller, ileride yazılabilecek simülasyon model dönüşümlerinde kullanılabilecektir. Model güdümlü geliştirme yaklaşımları ile daha kolay ve daha sistemli olarak geliştirilebilen simülasyon modelleri ile havalimanı terminali yolcu akış analizlerinin yapılmasıyla daha kullanışlı terminal binaları tasarlanabilecektir.

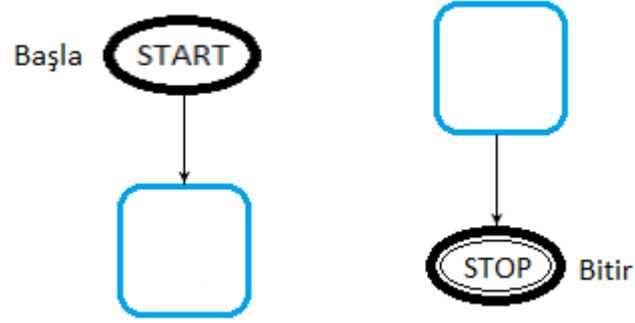


Şekil 5.17 : Yolcu akış metamodeli sınıf diagramı.



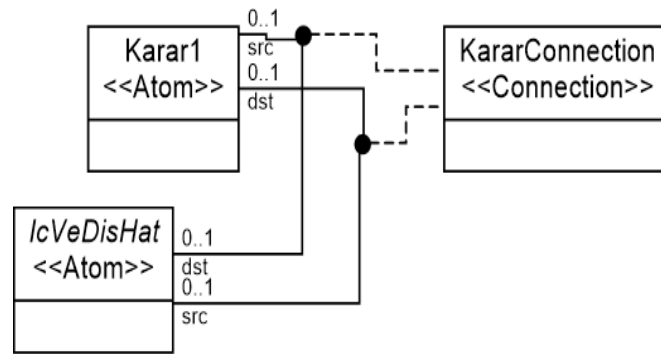
Şekil 5.18: Yolcu akış metamodeli ana şeması.

Akış Şeması Elemanları: Yaygın olarak kullanılan UML aktivite diyagramlarında yer alan akış şeması elemanlarından beş tanesi bu metamodelde kullanılmıştır. Bunlar, Topla, Dağıt, Karar, Başla (*Start*) ve Bitir (*Stop*) elemanlarıdır. Model görünümü Şekil 5.19’da verilen *Başla* ve *Bitir* akış şemasının başlangıç ve bitişinin tanımlandığı atomlardır. *Başla* elemanı *Başla Bağlantısı* (*ConnectStart*) kullanarak hiçbir atom tarafından *hedef* (*destination*) bağlantı olarak gösterilemezken modeldeki tüm elemanlara *kaynak* (*source*) pozisyonunda bağlantı sağlayabilir. *Bitir* elemanı ise *Bitir Bağlantısı* (*ConnectStop*) kullanılarak hiçbir atoma kaynak pozisyonunda bağlanamazken, modeldeki tüm elemanlarla hedef pozisyonunda bağlantı sağlayabilir.



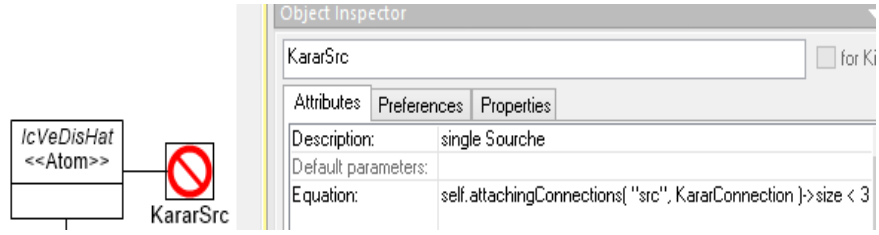
Şekil 5.19: Başla ve bitir elemanlarının görünümü.

Karar atomu akış şemasında kontrol ifadesini temsil eden model elemanıdır. *Karar* atomu *Karar Bağlantı* (*KararConnection*) atomu aracılığıyla, diğer model elemanlarıyla olan kaynak ve hedef bağlantısı Şekil 5.20’de gösterildiği gibi tanımlanmıştır.

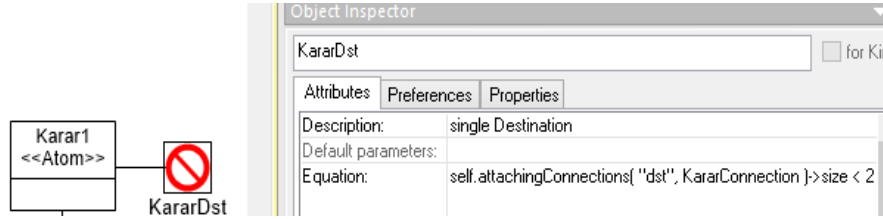


Şekil 5.20: Karar atomunun bağlantısı.

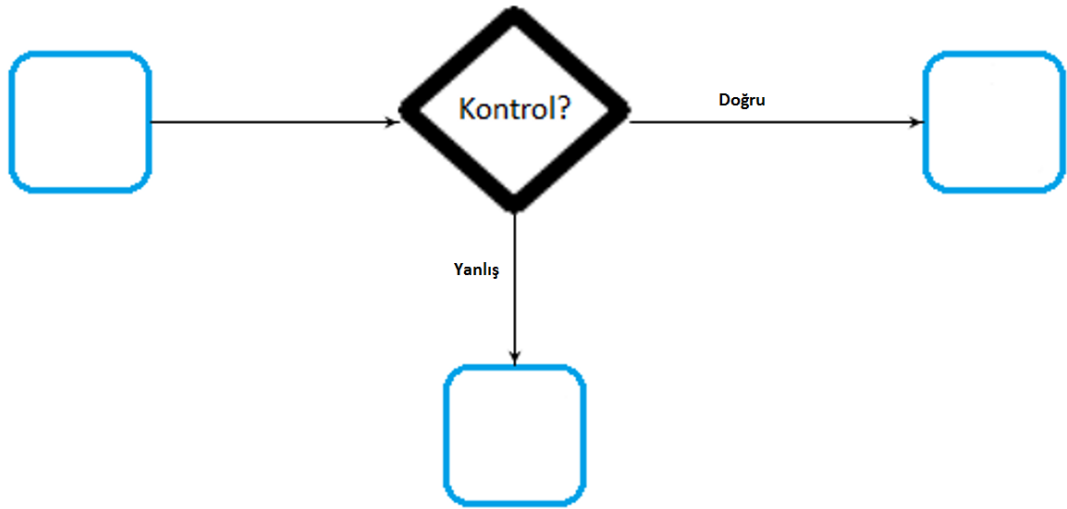
Şekil 5.21’de *Karar Kaynak* (*KararSrc*) kısıtlama yapısı ile *Karar* atomundan diğer model elemanlarına, model elemanlarından sadece iki tanesine bağlantı yapılması sağlanmıştır. Şekil 5.22’de ise gösterildiği gibi, tanımlanan *Karar Hedef* (*KararDst*) kısıtlama yapısı ile sadece bir model elemanından *Karar1* atomuna bağlantı yapılması sağlanmıştır. Şekil 5.23’te ise *Karar1* atomunun model kullanımı verilmiştir.



Şekil 5.21: Karar kaynak kuralı.



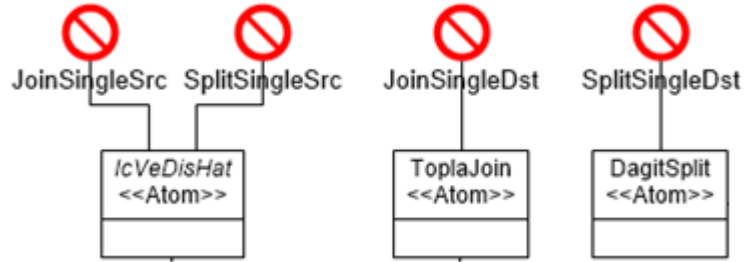
Şekil 5.22: Karar hedef kuralı.



Şekil 5.23: Karar yapısı kullanıcı ekranı.

Topla ve Dağıt ise akış şemasının daha düzenli gözükmesi amacı ile kullanılan akış şeması elemanlarıdır. Eğer bir atoma birden fazla model elemanından hedef bağlantı gerçekleşecekse *Topla* yapısı, bir atomdan birden fazla model elemanına bağlantı yapılması durumunda ise *Dağıt* yapısı kullanılabilir. Metamodelde bu yapıların bağlantıları *Topla (Join)* ve *Dağıt (Split)* bağlantı yapıları ve bu bağlantılar için tanımlanan *Topla Tek Kaynak (JoinSingleSrc)*, *Dağıt Tek Kaynak (SplitSingleSrc)*, *Dağıt Tek Hedef (SplitSingleDst)*, *Topla Tek Hedef (JoinSingleDst)*

kısıtlama yapıları kullanılarak tasarlanmıştır. Şekil 5.24'te bu kısıtlamaların kullanımları gösterilmiştir.



Şekil 5.24: Dağıt ve topla yolcu akış şeması elemanlarında tanımlı kurallar.

Havalimanı Elemanları: Yolcu akış şeması oluşturulurken kullanılacak havalimanı terminal binası birimleri beş grup altında tanımlanmıştır. Her grubun içinde tanımlanan yolcu akış şeması model elemanları ve bu modeller içinde tanımlanan değişkenler Tablo 5.2’de gösterilmiştir. Oluşturulan akış şeması metamodeli için, terminal planı metamodelinde olduğu gibi havalimanında bulunabilecek tüm birimleri metamodelde yerleştirmek yerine benzer birimler gruplandırılmıştır.

Yukarıda belirlenen model elemanlarının birbirleri ile bağlantıları her biri için tanımlanan bağlantı yapıları ile sağlanmıştır. Bu bağlantılar yapılırken yolcunun bulunduğu birimden geçebileceği veya geçemeyeceği birimler dikkate alınarak bazı sınırlandırmalar getirilmiştir. Bu bağlantılara birkaç örnek verilecek olursa; *Kara Tarafı Giden Yolcu* atomu yani kara tarafının giden yolcu bölümü altındaki birimler, Hava ve *Kara Tarafı Giden Bağlantı (ConnectDeparturesLandAir)* yapısıyla grup hava tarafı giden yolcu bölümünün model elemanlarına bağlantı kurabilmektedir. Ancak transfer yolcular yani *transfer* atomu *Transfer Bağlantısı (ConnectionTransfer)* ile sadece *Kara Tarafı Gelen Yolcu* ile *Hava Tarafı Gelen Yolcu* atomu altındaki model elemanlarına bağlantı kurabilir.

Tablo 5.2: Yolcu akış şeması model elemanları ve değişkenleri.

Grup İsmi	Model Elemanları	Tanımlanan Değişken
<i>Kara Taraflı Giden Yolcu (PassengerDepartureLand)</i>	<i>Check-in</i>	Zaman
	<i>Bilet veya Pasaport Kontrol (TicketOrPassportControl)</i>	İç-Dış Hat Seçimi Zaman
	<i>KaraGidenServis</i>	İç-Dış Hat Seçimi, Servis Adı, Zaman
<i>Kara Taraflı Gelen Yolcu (PassengerArrivalLand)</i>	<i>Terminal Çıkış</i>	
	<i>KaraGelenServis</i>	İç-Dış Hat Seçimi, Servis Adı, Zaman
<i>Hava Taraflı Gelen Yolcu (PassengerArrivalAir)</i>	<i>HavaGelenServis</i>	İç-Dış Hat Seçimi, Servis Adı, Zaman
	<i>Bagaj</i>	İç-Dış Hat Seçimi, Zaman
	<i>GelenYolcuKabul</i>	Zaman
	<i>Uçak Çıkış</i>	Gelen Yolcu Tipi, Zaman
	<i>Transfer</i>	Zaman
<i>Hava Taraflı Giden Yolcu (PassengerDepartureAir)</i>	<i>Uçuş Kapısı</i>	Zaman
	<i>Uçuş Salonu</i>	Zaman
	<i>HavaGidenServis</i>	İç-Dış Hat Seçimi, Servis Adı
<i>Transfer</i>	<i>Transfer</i>	Zaman

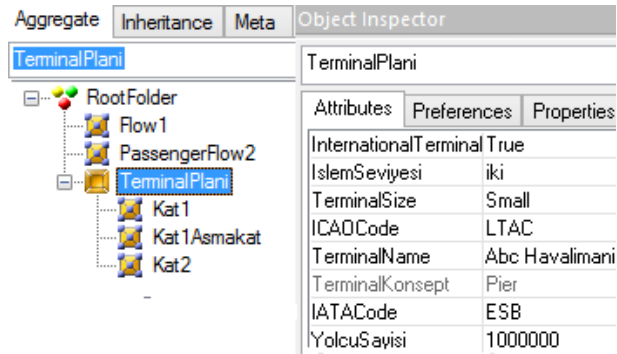
ALTINCI BÖLÜM

MODELLEME EDITÖRÜ VE KULLANICI ARAYÜZLERİ

Bir önceki bölümde açıklanan metamodel ile GME metamodelleme aracı kullanılarak bir modelleme editörü otomatik olarak üretilmiştir. Oluşturulan editör yine GME ortamına benzer bir modelleme aracıdır. Ancak modelleme elemanları ve kuralları tanımlanan metamodelle uyumludur ve görsel açıdan istenildiği şekilde geliştirilebilmektedir. Bu bölümde havalimanı terminal modellemesi için kullanılacak ve önerilen metamodelle tam uyumlu olan modelleme editörü tanıtılacaktır. Modelleme editörü ve bu bölümde sunulan örnek modeller, önerilen metamodelin pratikte uygulanabilirliğini ve farklı ölçekteki havalimanları için başarılı bir şekilde kullanılabildiğini göstermektedir.

6.1. Terminal Yerleşim Modeli

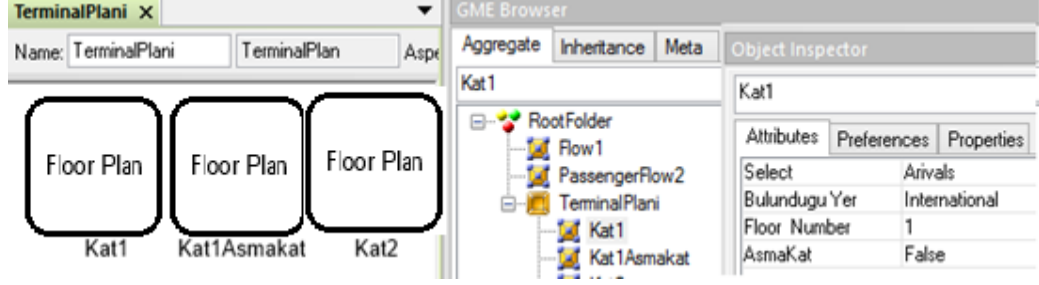
Terminal yerleşim modeli ile havalimanı kat planları çizilebilmektedir. Bu model için metamodelde tanımlanan model özellikleri Şekil 6.1’de görüldüğü gibi, modellemesi yapılacak terminalin değişkenler alanına girilmesi gerekmektedir.



Şekil 6.1: Modelleme ekranı TerminalPlan modeli özellikler ekranı.

6.1.1. Kat Modeli

TerminalPlan modelinde ilk olarak kat modeli eklenmesi gerekmektedir. Eklenen Kat modeli sayısı ile terminalin kaç katlı olacağı belirlenir. Aynı zamanda asma kat olarak tasarlanacak katlarda yine bu seviyede kat modeli olarak eklenmelidir. Kat özellikleri ise Şekil 6.2’de görüldüğü gibi kat modelinin değişkenleri ile tanımlanır.



Şekil 6.2: Modelleme ekranı terminalplan ekranı altındaki kat modeli özellikler ekranı.

Kat planında eklenebilecek birimler Şekil 6.3’te gösterilmiştir ve bu birimlerin İngilizce ve Türkçe karşılıkları Ek-A Tablo A.1’de verilmiştir. Planın bu bölümüne atom olarak 58 tane havalimanı birimi dış hat ile iç hat olmak üzere 2 adet de model eklenebilir.



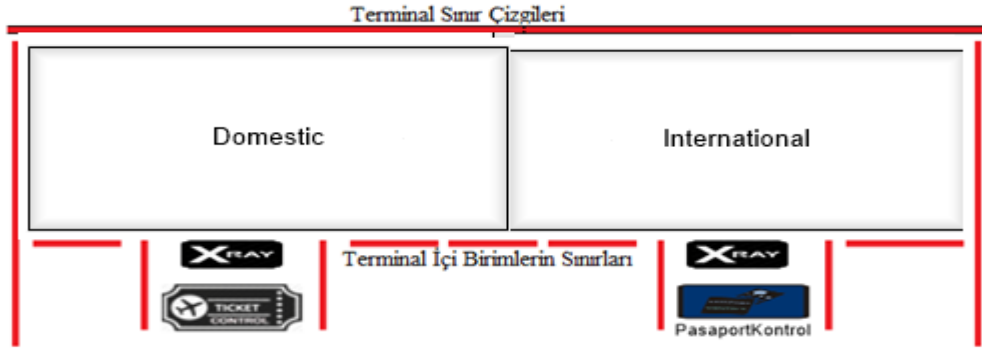
Şekil 6.3: Kat modelinde bulunan havaalanı birimleri model elemanları.

Bu bölümde kullanıcı sadece, havalimanı terminal binasının iç veya dış hatlar kara tarafında bulunabilecek birimleri tasarımına ekleyebilir. Terminalin iç hatlar veya dış hatlar hava tarafında bulunabilecek birimler kat modeli ekranı üzerinde bulunmamaktadır. Metamodeldebu alanlar birimlerin özelliklerine göre dış hat ve iç hat modelleri olarak ayrılmış ve buna göre bu modellerin ayrı tanımlanması istenmiştir. Böylece terminalin kat planı hava ve kara tarafı şeklinde modüler olarak tasarlanması sağlanarak, kullanıcının modeli daha düzenli ve havalimanının tasarım esaslarına da riayet ederek meydana getirmesi amaçlanmıştır.

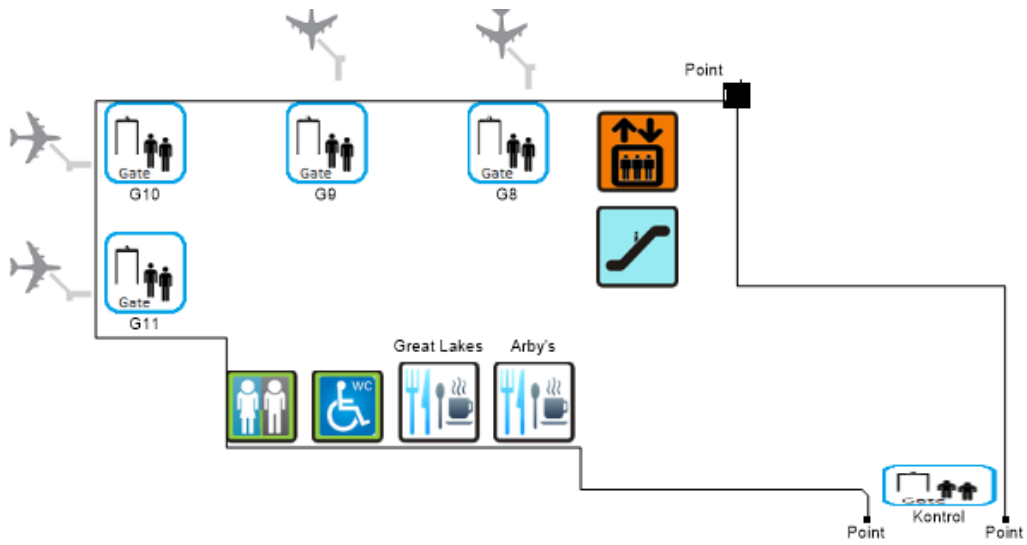
Havalimanının hava tarafı iç ve dış hat model görünümüne herhangi bir renk ya da resim tanımlaması yapılmamıştır. Diğer model elemanlarında olduğu gibi boyutu değiştirilebilir olarak tanımlanmıştır ve renk ya da resim tanımlamaları yapılarak model görünümleri isteğe bağlı olarak kullanıcılar tarafından değiştirilebilir.

Metamodelde tanımlanan kısıtlama tanımlamaları doğrultusunda dış hatlar modelinin kat modelinin içine eklenebilmesi için pasaport kontrol biriminin eklenmiş olması istenmektedir. Aynı kontrol, pasaport kontrol alanı eklendiği zaman dış hatlar modelinin aranması içinde tanımlanmıştır. Pasaport kontrol birimi ile bilet kontrol ve XRay birimleri, hem kara hem de hava taraflarında iki taraf arasında geçiş olarak kullanıldığı için hem kat modelinde hem de iç ve dış hat bölümlerinin karşılık gelen kısımlarına eklenmesi için kullanıcı tasarım ekranı görünüm bölümüne yerleştirilmiştir.

Terminal alanlarını birbirinden ayırmak ve yerleşim planında alanların sınırlarını belirlemek için iki farklı yöntem önerilmiştir. Şekil 6.4'te görüldüğü gibi modelde yatay ve dikey çizgi atomları kullanılabilir. Ya da Şekil 6.5'te olduğu gibi *nokta (Point)* nesnesi üzerinde tanımlanan bağlantılar sayesinde *noktalar* arası veya *noktadan* diğer model nesneleri arasında bağlantılar oluşturularak sınır çizgileri belirlenebilir. Nokta nesnesi üzerindeki bağlantının otomatik yönlendirme özelliği inaktif yapılarak dik açılar dışında sınır çizgileri de belirlenebilir.



Şekil 6.4: Terminal sınır çizgileri dikey ve yatay çizgi uygulaması.



Şekil 6.5: Terminal sınır çizgileri nokta uygulaması.

Mevcut havalimanı terminallerinde bulunan standart birimlerin dışında kullanıcı, tasarımında farklı birimler ya da oluşturduğu havalimanına özgü bir alan tasarlamak isterse, bu alanı Serbest Alan atomu ile boyut ve diğer görüntü özelliklerini de belirleyerek oluşturabilir.

6.1.2. İç Hat ve Dış Hat Modelleri

Havalimanı terminali hava tarafı dış hatlar gelen ya da giden yolcu alanlarının tasarımı için oluşturulan Dış Hat modeli içinde Şekil 6.6'da gösterildiği gibi 10 tanesi hava tarafında bulunabilecek birimler olmak üzere toplam 45 adet havalimanı birimi bulunmaktadır.



Şekil 6.6: Hava tarafında bulunan farklı birimler.

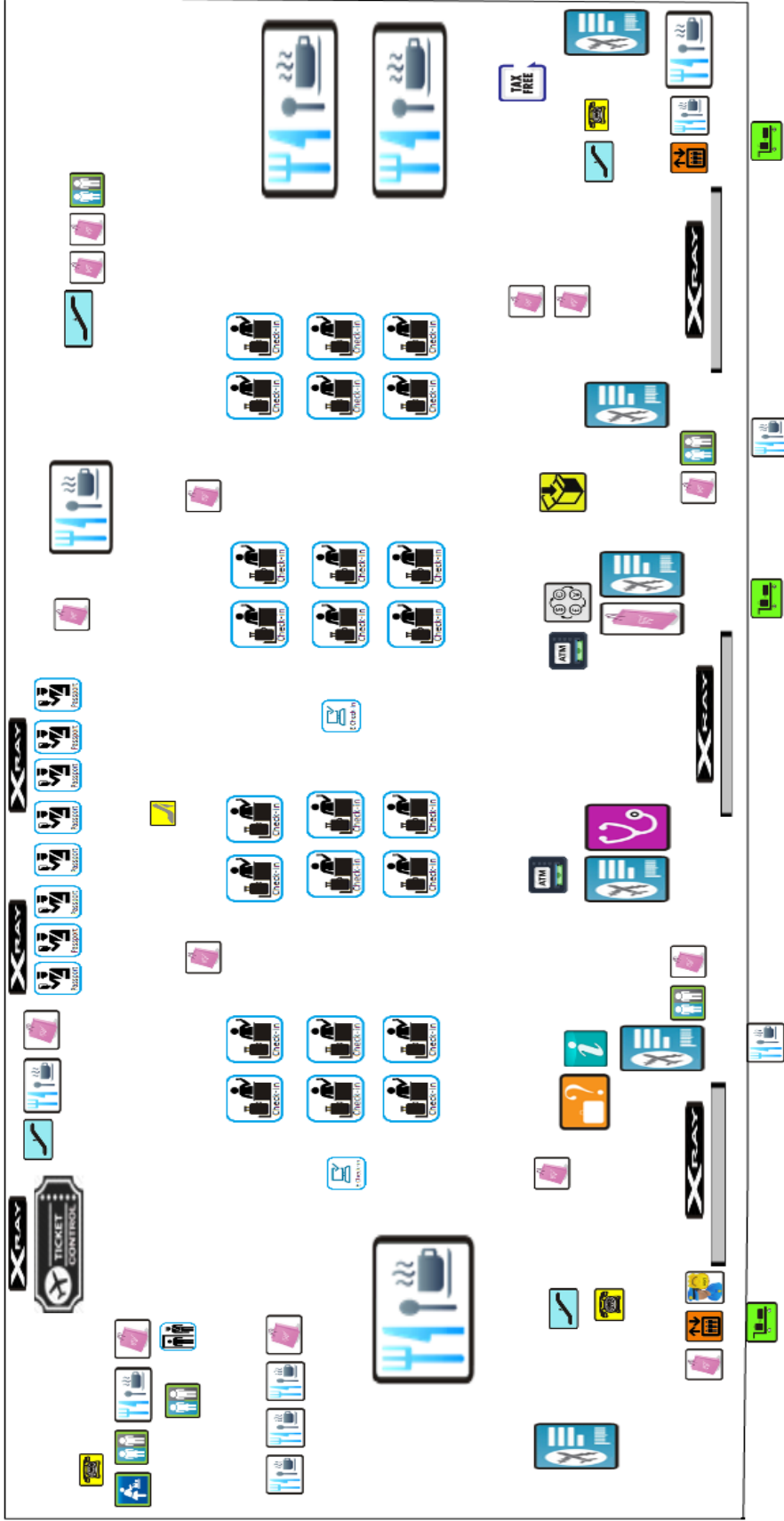
İç hat modelinde ise 9 tanesi hava tarafında bulunabilecek birimler olmak üzere toplam 41 adet havalimanı birimi bulunmaktadır. Dış hat ve iç hat modelleri hava tarafı için hem gelen hem de giden yolcu alanları tasarımında kullanılabilir. Bu nedenle kullanıcılar eklenen modelin değişken seçeneklerinde gelen, giden ya da modeli hem gelen hem de giden yolcu için kullanacaklarsa gelen ve giden seçeneklerinden birini seçmeleri gerekmektedir.

Serbest Alan bu modellerin uçuş tarafı içinde de kullanılabilir. Uçağa binmeden önce yolcuların bekletildiği salonlar *Uçuş Salonları* olarak, uçak ile terminal arasındaki son noktayı belirten yer ise *Uçak Biniş (FlightGate)* olarak tanımlanmıştır. *Uçak* olarak isimlendirilen model elemanı ile, kat planı modelinde uçağın terminale paralel ne kadar yer kaplayacağı ve konumu hesaplanarak, Uçak Biniş ve Uçuş Salonları birimlerinin buna göre nasıl konumlandırılacağı daha gerçeğe uygun olarak tasarlanabilir.

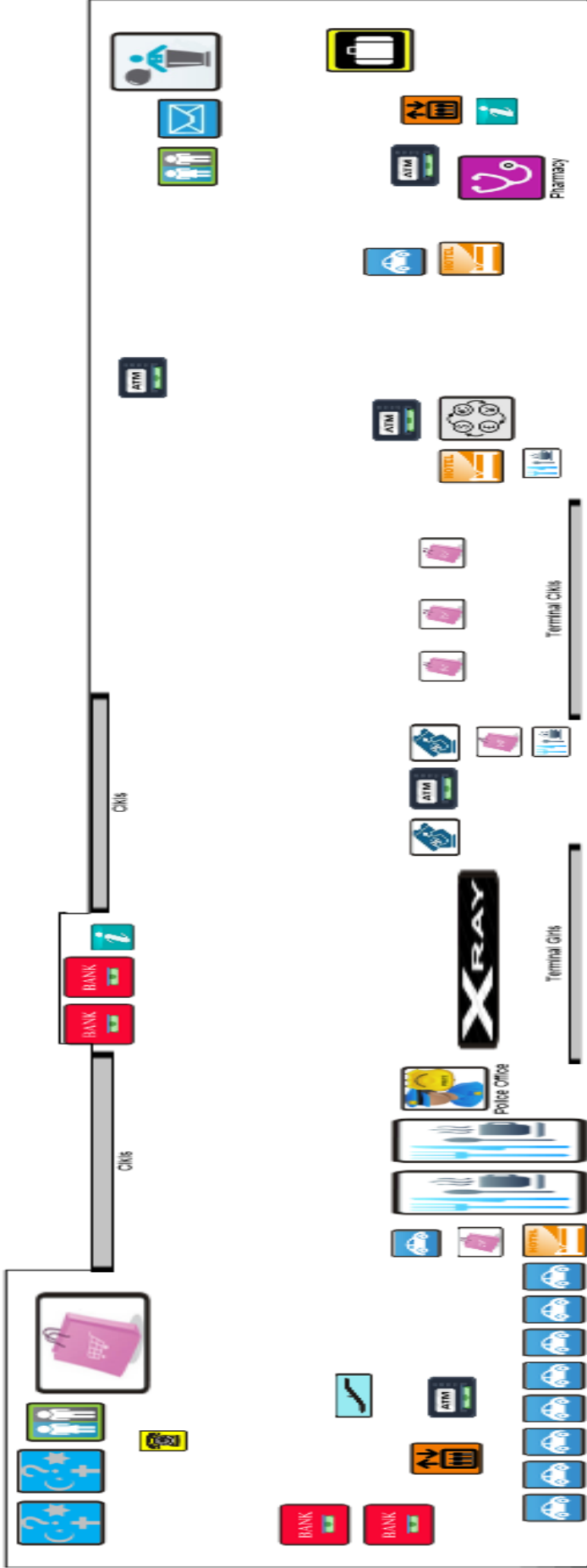
Önerilen metamodel ve otomatik geliştirilen model editörü sayesinde, havalimanı terminal modellemesi için alana özgü bir modelleme dili ve bu dilin kullanımına olanak sağlayan bir araç sunulmuştur. Modelleme dilinin ve editörün kullanımına yönelik çalışmalarda farklı büyüklükte örnek modeller çizilerek modelleme dilinin yeterliliği test edilmiştir. Tezin bu bölümünde, İstanbul Sabiha Gökçen Havalimanı örnek olarak sunulmuştur. Şekil 6.7 ve Şekil 6.8’de Sabiha Gökçen Havalimanı terminali kara tarafının sırası ile giden ve gelen yolcu bölümlerinin çizimi verilmiştir. Şekil 6.9’da hava tarafı gelen yolcu bölümü, Şekil

6.10'da hava tarafı giden yolcu bölümü, Şekil 6.11 ve Şekil 6.12'de ise hava tarafı giden asma kat ve -1. kat örnek çizimleri gösterilmiştir.

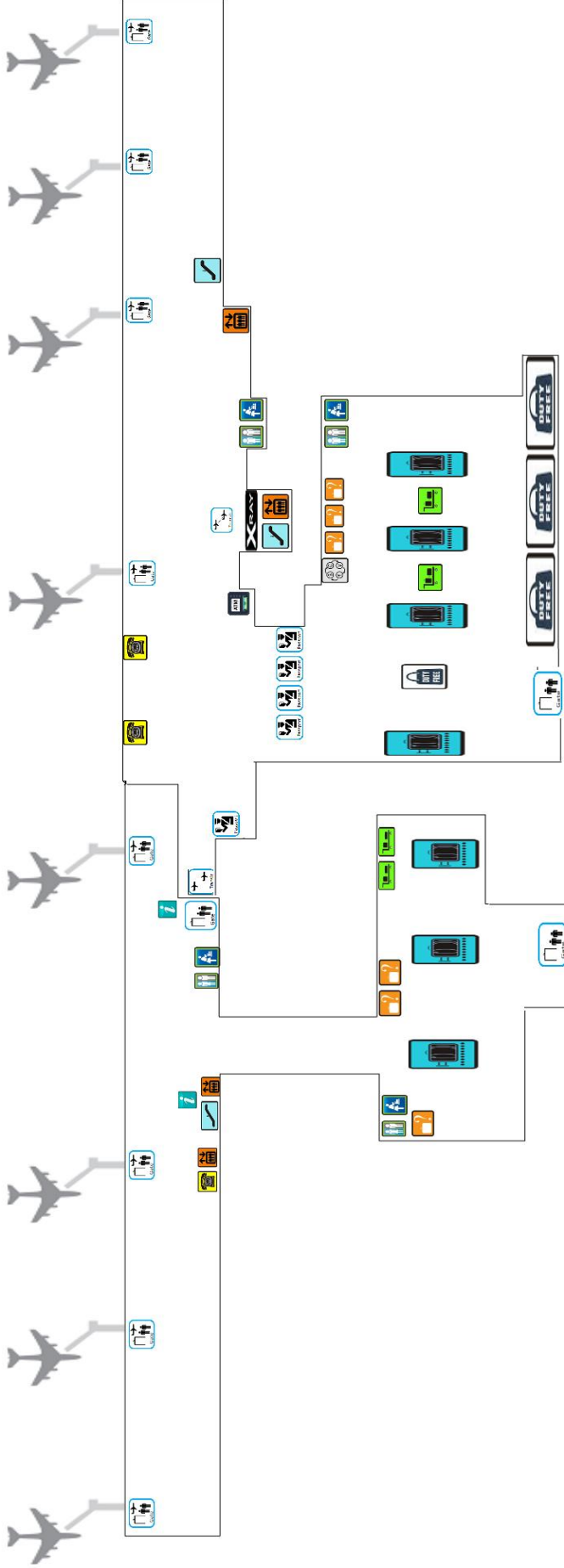
Farklı boyutlarda havalimanı terminal binalarının yerleşim planları için örnek çizimlerden bazıları Ek B ve Ek C'de verilmiştir. Ek B'de, Esenboğa havalimanının kara tarafı giden ve kara tarafı gelen yolcu bölümleri sırası ile Şekil B.1'de ve Şekil B.2'de, hava tarafı gelen ve hava tarafı giden yolcu bölümleri ise sırası ile Şekil B.3'te ve Şekil B.4'te verilmiştir. Küçük bir havalimanına örnek olarak da Ek-C'de Amerika'da özel bir havalimanının kara tarafı giden-gelen yolcu bölümü ve hava tarafı giden-gelen yolcu bölümü sırası ile Şekil C.1'de ve Şekil C.2'de sunulmuştur.



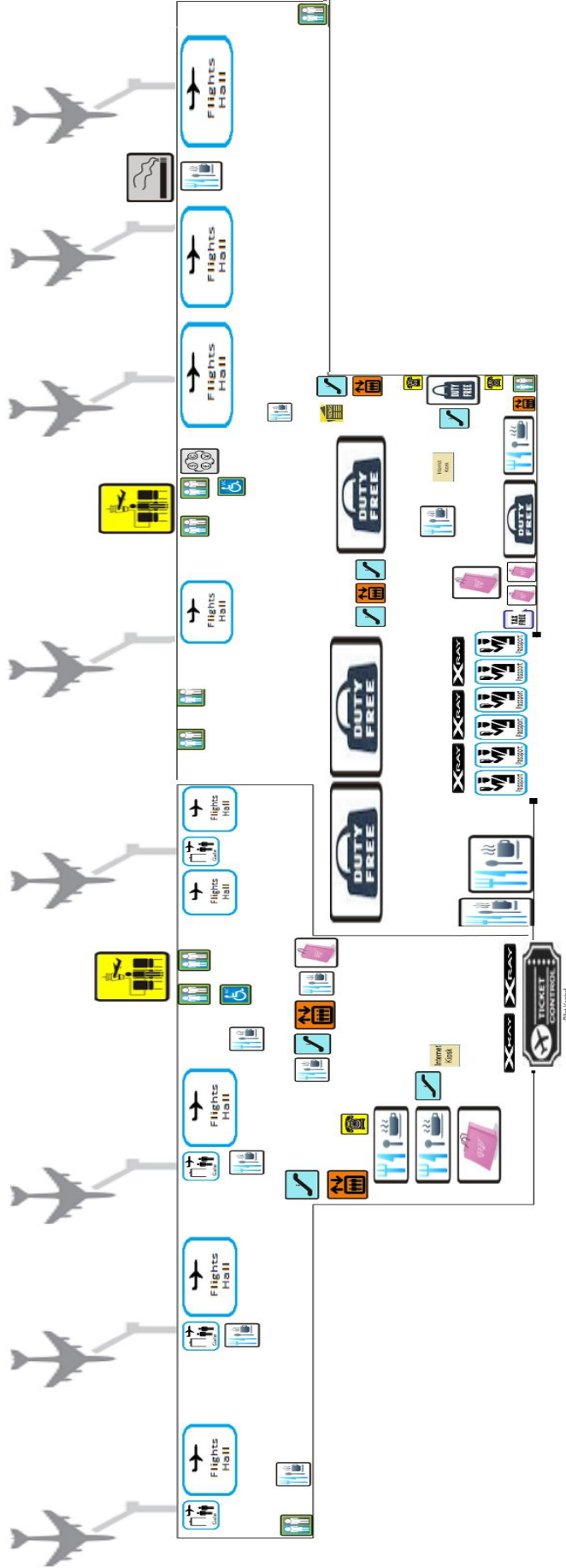
Şekil 6.7: Sabiha Gökçen Havalimanı kara tarafı giden yolcu bölümü.



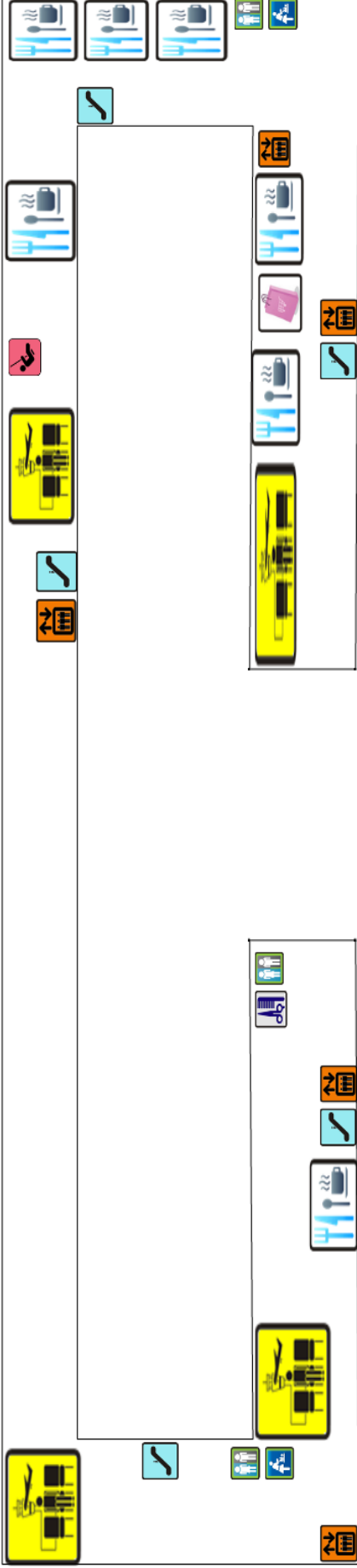
Şekil 6.8: Sabiha Gökçen Havalimanı kara tarafı gelen yolcu bölümü.



Şekil 6.9: Sabiha Gökçen Havalimanı hava tarafı gelen yolcu bölümü.



Şekil 6.10: Sabiha Gökçen Havalimanı hava tarafı giden yolcu bölümü.



Şekil 6.11: Sabiha Gökçen Havalimanı hava tarafı giden yolcu -1. kat.

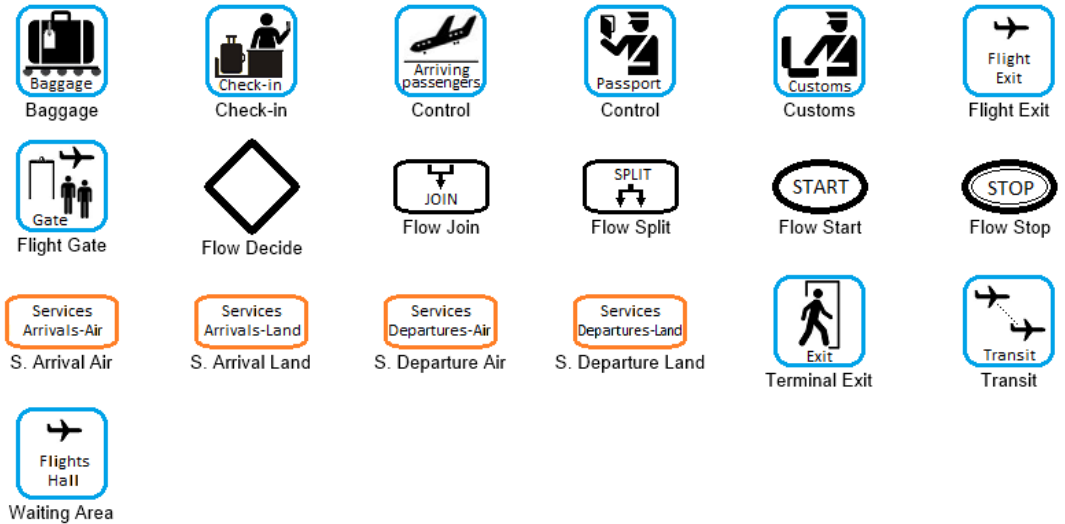


Şekil 6.12: Sabiha Gökçen havalimanı hava tarafı giden yolcu asma katı.

6.2. Yolcu Akış Modeli

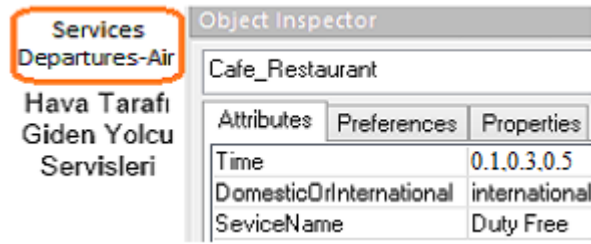
Havalimanı terminal model tasarımında, TerminalPlan modelinden sonra kök klasörde bulunan diğer model ise Akış modelidir. Akış modeli ile kullanıcılar kat planını yaptıkları terminal binasında gelen ya da giden yolcular için bir akış diyagramı tanımlayabilirler. Bu akış modeli yolcuların terminal birimlerini hangi sırayla ziyaret ettiklerini ve bu birimlerde ne kadar süre kaldıklarını göstermek için kullanılabilir. Bu bağlamda, ileride simülasyon modeline dönüşüm için gerekli tüm bilgiler elde edilmiş olmaktadır.

Akış şemasında kullanılabilecek tüm model elemanları akış kök klasörü içinde Şekil 6.13'te resimleri görülen şekilde bulunmaktadır ve İngilizce Türkçe karşılıkları Ek-A Tablo A.1'de verilmiştir. Bunlardan 5 tanesi genel akış şeması elemanları iken 13 tanesi de havalimanı birimleri olmak üzere 18 adet model elemanı bulunmaktadır. Akış şeması elemanları Başla, Bitir, Karar, Topla (birden fazla kaynağı toplar), Dağıt (birden fazla hedefe bağlantı sağlar) olarak isimlendirilmiştir. Havalimanı elemanları ise Check-in, Uçak Çıkış (uçaktan terminale girişi temsil eder), Uçuş Salonu (uçığa binmeden önce), Uçak Biniş (uçığa biniş kapısı), TerminalGate (terminal giriş çıkış kapısı), Transit (transit geçişler), Bagaj, Kontrol (pasaport ya da bilet kontrol işlemleri), Servis (hava tarafı ya da kara tarafı, gelen ya da giden yolcu, terminal ya da özel servisler) olarak tanımlanmıştır.



Şekil 6.13: Akış şeması model elemanları.

Havalimanı terminal binalarında bulunan servisler ve birimler akış şeması genel özellikleri doğrultusunda içeriklerine göre gruplandırılmıştır. Şekil 6.14'teki örnekte gösterildiği gibi gruplandırılmış bir elemanı değişkenler sayesinde detaylandırılabilir. Örneğin, Hava Tarafı Giden Yolcu Servisleri olarak gruplandırılan birim, dış hatlar ve Gümrüksüz Alışveriş olarak kullanıcı tarafından özelleştirilebilmektedir. Simülasyon modeli içinde önemli bir veri olan, birimde geçirilen zaman bilgisi deparametre olarak girilebilmektedir.



Şekil 6.14: Akış şeması hava tarafı giden yolcu servisleri özellikler penceresi.

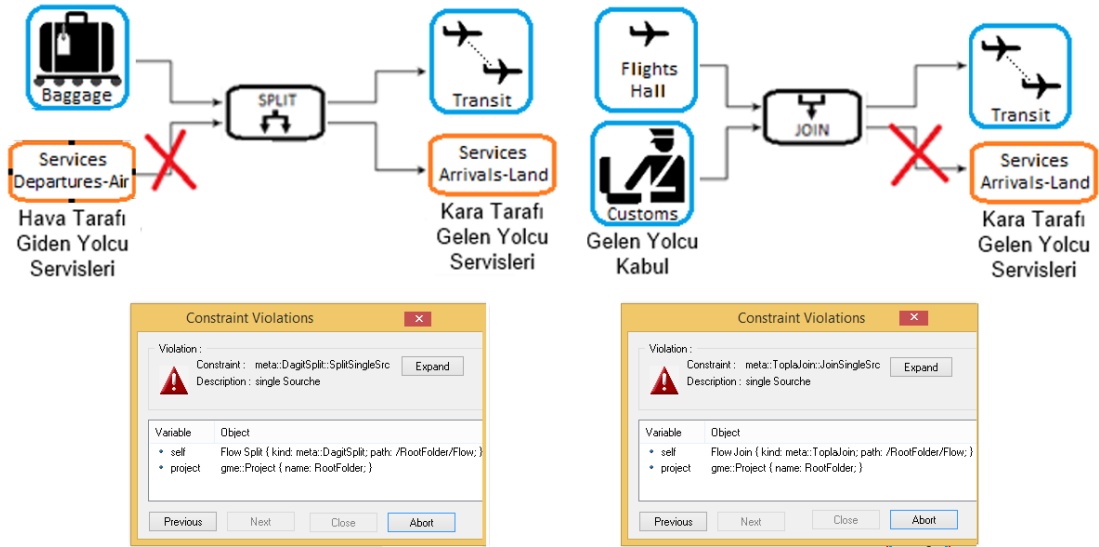
Metamodelde Bağlantı atomları ile model tasarımında, hangi birimlerin birbirleri ile bağlantı sağlayabilecekleri tanımlanmıştır. Kısıtlamalar ile de hangi birimlerin birbirleri ile ne kadar ve hangi yönde bağlantı kurabilecekleri sınırlandırmalar ve kurallar ile tanımlanmıştır. Böylece Şekil 6.15'te gösterilen örneklerde olduğu gibi yolcu akış modelinde hatalı modelleme girişimleri gerekli yerlerde program tarafından verilen uyarı ekranları engellenmiştir.

Örneğin, havalimanı terminali yolcu akış şemasında, hava tarafı giden yolcu tarafında bulunan bir kişinin bir sonraki yapacağı iş bagaj alım işlemi olamayacağı için bu iki birim arasındaki bağlantı sınırlandırılmıştır. Uçuş Salonu alanında bulunan bir yolcunun bir sonraki gideceği yer gelen yolcu kabul birimleri olamayacağı için bu işlem de sınırlandırılmıştır. Ya da transit bir yolcunun kara tarafı gelen yolcu servislerinden faydalanması mümkün olmadığı için bu bağlantılara da izin verilmemiştir. Bu örnekler, bağlantı tanımlaması ile sınırlandırmaya birer örnektir.



Şekil 6.15: Yolcu akış şemasında bağlantı yapılamayacak yerler için örnekler.

Kısıtlama tanımlamaları ile sınırlandırmalar ise genel akış şeması elemanları için tanımlanmıştır. Örneğin Şekil 6.16’da gösterildiği gibi dağıt elemanına birden fazla kaynaktan bağlantı sağlandığında ya da topla elemanından birden fazla hedefe bağlantı yapıldığında program uyarı penceresi ile yanlış kullanımı engellemektedir. Ya da, karar elemanına bir kaynaktan bağlantı gelip en fazla iki hedefe (doğru-yanlış) bağlantı sağlanması da yine kısıtlama sınırlandırmalarına örnektir.



Şekil 6.16: Dağıt ve topla kural dışı kullanım için uyarı pencereleri.

Takip eden alt bölümlerde, havalimanı terminali yolcu akışı ile ilgili bazı senaryolar ve bu senaryolara göre çizilebilecek örnek yolcu akış modelleri verilmiştir.

6.2.1. İç Hatlar Giden Yolcu Senaryosu

Ülke sınırları içinde bir yerden başka bir yere giden yolcu için bir senaryo yazılmıştır. Bu senaryoya göre terminale gelen yolcunun uçağa binene kadar yapacağı işlemler oluşabilecek aksaklıklar göz ardı edilerek sıralanmıştır.

Terminale gelen yolcu bileti olmaması durumunda ilk önce uçuş firması ofislerinden biletini alır. Bagajı olan biletli yolcu bagaj paketlemeye ve Check-in bankosuna geçer. Bagajı olmayan yolcu ise doğrudan Check-in bankosuna gider. Uçuş saatine daha varsa kara tarafı servislerinden yararlanabilir. Sonra bilet kontrol noktasından hava tarafına geçiş yapar. Eğer uçuşa zaman varsa hava tarafı servislerinden faydalanabilir. Ardından uçuş salonuna geçer. Son olarak uçuş kapısından uçağa binerek terminalden ayrılır.

Şekil 6.17’de yukarıda anlatılan senaryoya göre modelleme editörü kullanılarak çizilen akış modeli gösterilmektedir.

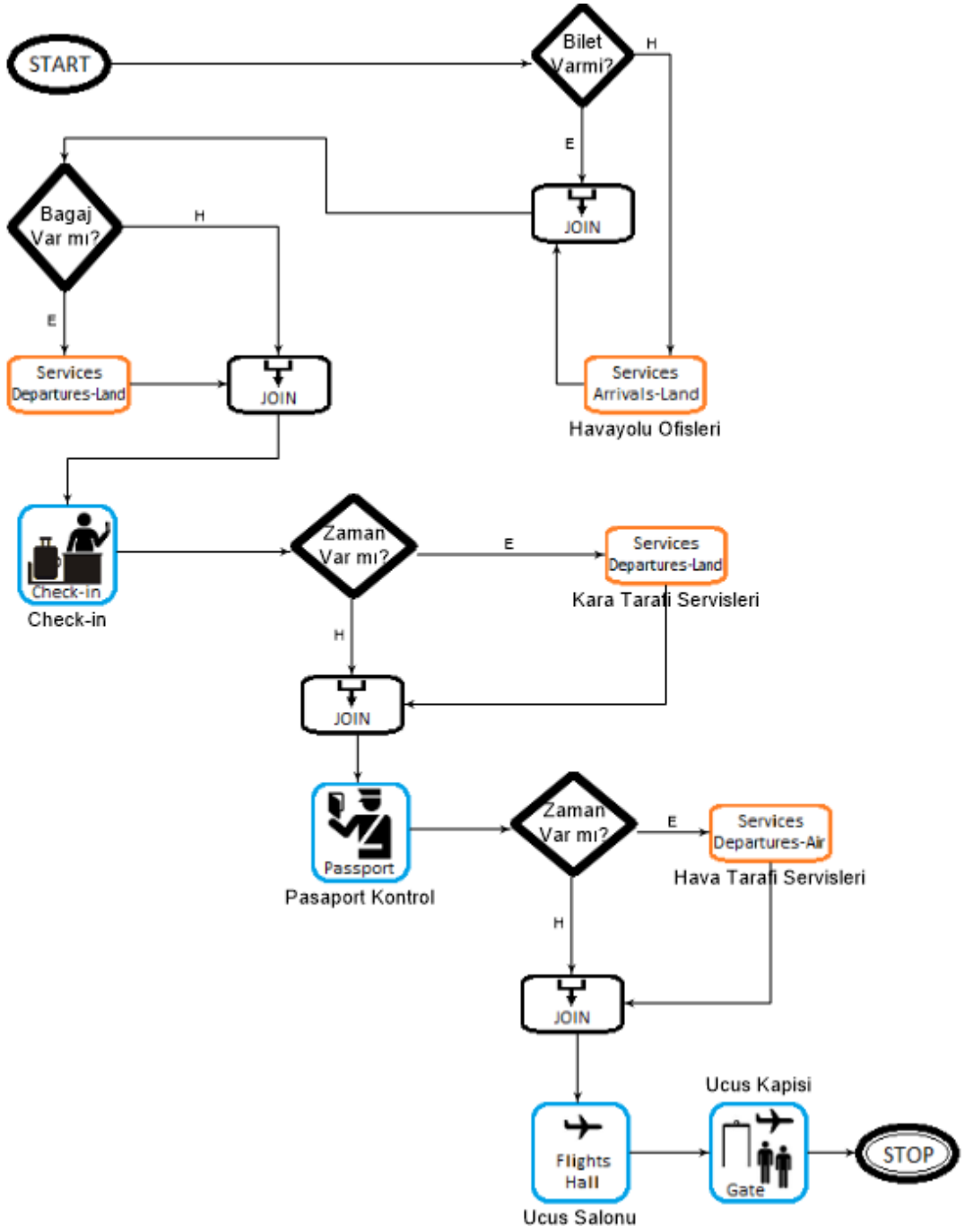
6.2.2. Dış Hatlar Gelen Yolcu Senaryosu

Yurt dışından ülkemize gelen yolcu için bir senaryo yazılmıştır. Bu senaryoya göre uçakla gelen bir yolcunun terminali terk edene kadar yapacağı işlemler oluşabilecek aksaklıklar göz ardı edilerek sıralanmıştır.

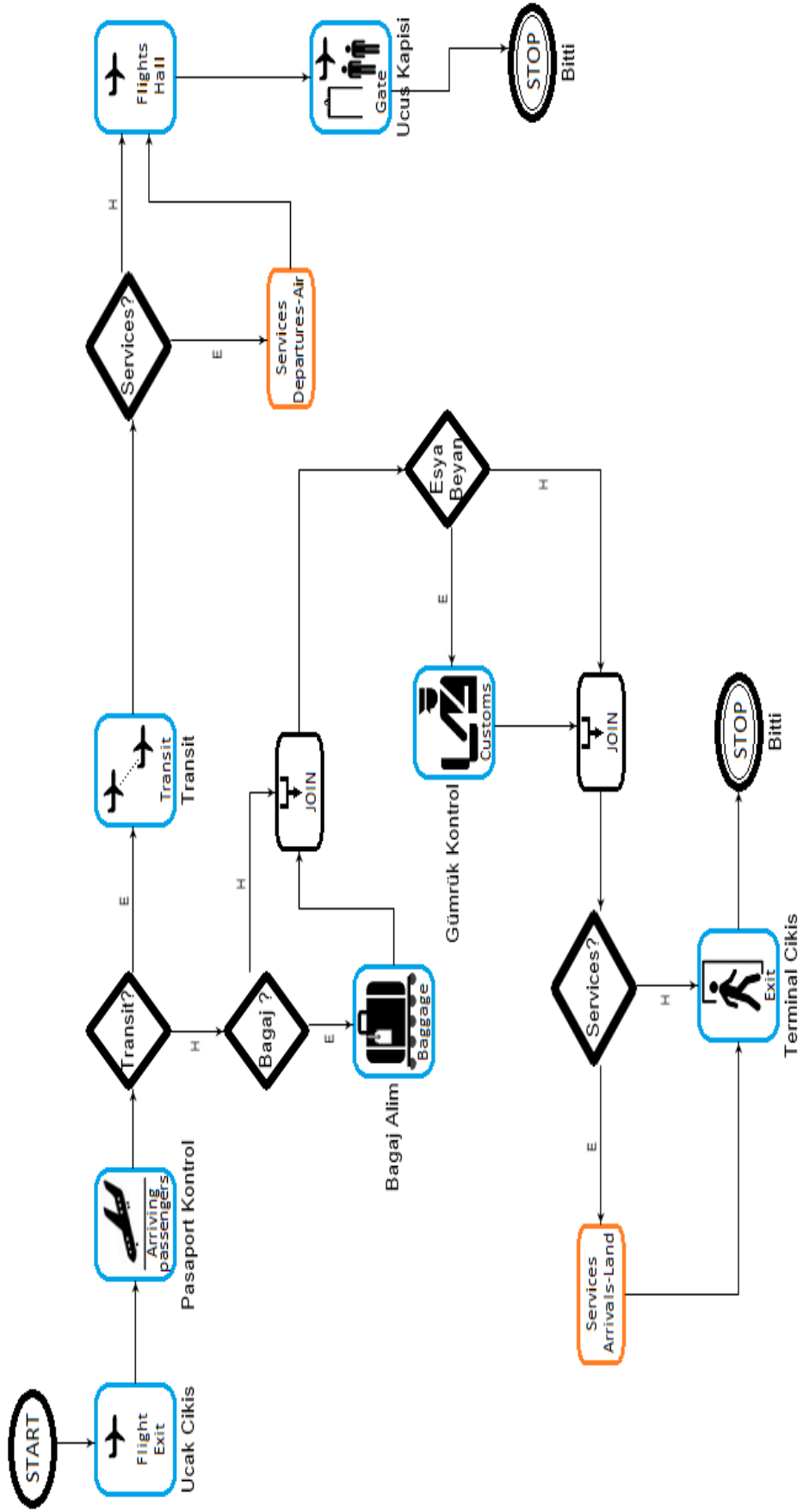
Yurt dışından gelen bir yolcu ilk öce uçaktan çıkar ardından pasaport kontrolüne gider. Eğer transit yolcu ise transit işlemlerini yapar. İsterse hava tarafı servislerini kullanabilir. Sonra uçuş salonuna geçer. Ardından uçuş kapısından uçağa biner ve terminalden ayrılır.

Transit yolcu olmayan yolculardan bagajı olanlar bagajlarını alır. Bagajlı ya da bagajsız yolcuların eğer beyan edilmesi gereken bir eşyası vb. varsa gümrük kontrole uğrarlar. Eğer yolcu isterse hava tarafı servislerinden yararlanabilir. Sonra kara tarafına geçer. İstenirse kara tarafı servislerini kullanabilir. Ardından çıkış kapısından çıkarak terminalden ayrılır.

Şekil 6.18’de yukarıda anlatılan senaryoya göre modelleme editörü kullanılarak çizilen akış modeli gösterilmektedir.



Şekil 6.17: İç hatlar giden yolcu akışı için hazırlanan yolcu akış modeli.



Şekil 6.18: Dış hatlar gelen yolcu akışı için hazırlanan yolcu akış modeli.

6.3. Modellerin XMI formatında saklanması

Havalimanı terminal modelleri GME aracının Genişletilebilir İşaretleme Dili (XML, Extensible Markup Language) ve Meta-Veri Değişimi (XMI, Metadata Interchange) desteği sayesinde farklı araçlarda da kullanılabilir şekilde XMI formatında saklanabilmektedir. XMI, MOF tabanlı metamodellerin ve modellerin nasıl XML kodlarıyla ifade edilebileceğini belirleyen bir OMG standarttır. XML, metamodellerin ifade edilmesinde ve model dönüşümlerinde kullanılabilmesi nedeniyle oldukça yaygın kullanılan bir standarttır. Örneğin, metamodelde kat modeli için tanımlanan değişkenlerden kat numarası (FloorNumber) için üretilen XMI kodları aşağıdaki gibidir. Burada kat numarası için tanımlanan değerler de ([-3,3] aralığında) gösterilmektedir.

```
<attrdef name="FloorNumber" metaref="2507" valuetype="enum" defvalue="0">
  <dispname>Floor Number</dispname>
  <enumitem dispname="-3" value="-3"></enumitem>
  <enumitem dispname="-2" value="-2"></enumitem>
  <enumitem dispname="-1" value="-1"></enumitem>
  <enumitem dispname="0" value="0"></enumitem>
  <enumitem dispname="1" value="1"></enumitem>
  <enumitem dispname="2" value="2"></enumitem>
  <enumitem dispname="3" value="3"></enumitem></attrdef>
```

Metamodel kullanılarak tasarlanan örnek yerleşim planı modelinin XMI kodlarının bir bölümü ise aşağıda verilmiştir. Burada ise, metamodelde tanımlanan değişkenler kullanılarak bu değişkenlere girilen değerler ile tanımlanan model elemanının bilgileri görülmektedir.

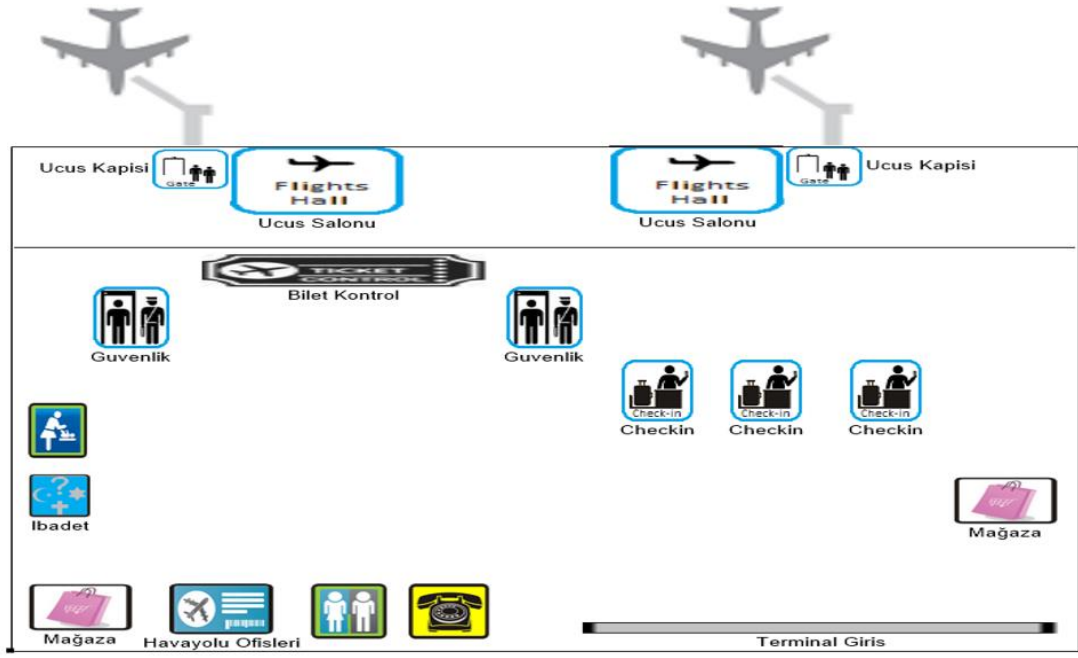
```
<model id="id-0065-00000001" kind="TerminalPlan" guid="{...}">
  <name>TerminalPlan</name>
  <attribute kind="IATA_Code">
    <value>32432</value>
  </attribute>
  <attribute kind="InternationalTerminal">
    <value>true</value>
  </attribute>
  <attribute kind="TerminalName">
    <value>sabihagokcen</value>
  </attribute>
  <attribute kind="Yolcu_Sayisi">
    <value>1000000</value>
  </attribute>
  <attribute kind="islem_Seviyesi">
    <value>iki</value>
  </attribute> </model>
```

6.4. Simülasyon Modellerine Dönüşüm

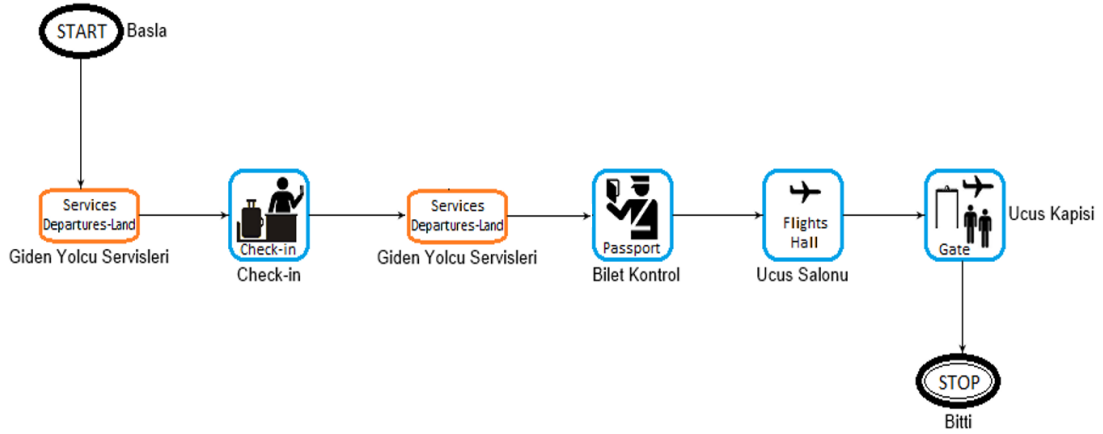
Bu bölümde önerilen metamodel kullanılarak oluşturulmuş bir modelin simülasyon modeline dönüştürülebilmesi için yapılması gerekenler kısaca açıklanmıştır. Sunulan basit bir örnek uygulamanın da ileride yapılacak çalışmalara ışık tutması amaçlanmıştır.

Model dönüşümleri için gerekenler, Şekil 4.4'te açıklanmıştı. Bunlar, kaynak metamodel, hedef metamodel, kaynak model ve model dönüşüm kuralları şeklinde gösterilmişti. Kaynak metamodel, bu tezde önerdiğimiz havalimanı terminal metamodeli ve kaynak model de model editörü ile tanımlanan terminal modeli ve yolcu akış modeli olarak kullanıldığında, geriye hedef metamodel ve model dönüşüm kurallarının belirlenmesi kalmaktadır. Hedef metamodel için daha önceden tanımlanmış metamodeller kullanılabilir ve kaynak modeller herhangi başka bir modele veya dokümana dönüştürülebilir. Simülasyon modeline dönüştürmek istiyorsak, simülasyon için tanımlanmış metamodeller kullanılabilir. İleride yapılacak çalışmalarda asıl yapılması gereken model dönüşüm kurallarının belirlenmesidir. Model dönüşüm kuralları herhangi bir model dönüştürme dili ile yazılabilir. Ancak, asıl önemli olan kaynak ve hedef metamodeller arasındaki eşleştirmelerdir. Burada basit bir örnek ile bu eşleştirmelerin nasıl yapılabileceği hakkında fikir verilmeye çalışılmıştır.

Örnek olarak, küçük bir havalimanının, modelleme editörü kullanılarak çizilen, giden yolcu havalimanı terminal binası yerleşim planı Şekil 6.19'da görüldüğü gibi verilmiştir. Bu havalimanın iç hatlarından bagajsız giden bir yolcunun, terminal girişten uçağa binene kadar asgari yapacağı işlemler temel alınarak oluşturulan akış şeması da Şekil 6.20'de görüldüğü gibidir.



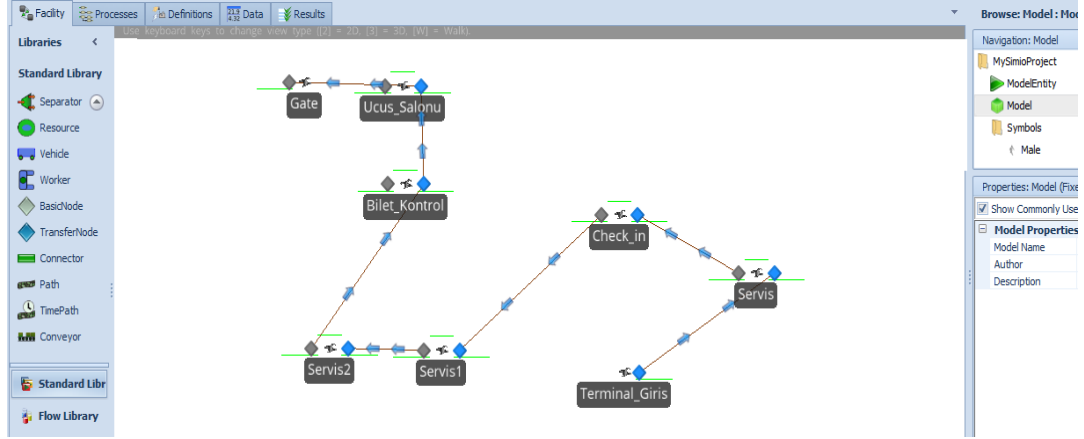
Şekil 6.19: Küçük bir havalimanının terminal yerleşim planı.



Şekil 6.20: Küçük bir havalimanı için yolcu akış şeması.

Çizilen terminal yerleşim planı ve yolcu akış şeması modellerinin XMI formatındaki dosyaları otomatik olarak üretilmektedir. Hem metamodel, hem de XMI formatındaki modeller herhangi bir model dönüştürme aracında girdi olarak kullanılabilir. Örnek hedef ortam olarak Simio adlı bir simülasyon aracı seçilmiştir. Bu aracın seçilmesinde herhangi bir özel neden olmamakla beraber, aracın kolay kullanımı ve XML formatında model saklama ve yükleme desteği olması önemli

avantaj sağlamıştır [43]. Yukarıda akış modeli verilen örnek için Simio programı ile çizilen simülasyon modelinin tasarımı Şekil 6.21’de görüldüğü gibidir.



Şekil 6.21: Yolcu akış şemasına göre oluşturulan simülasyon modeli örneği.

Şekil 6.21’de görülen uygulamada, Terminal Giriş olarak belirtilen model elemanı ile belli zaman aralıklarında terminale girecek yolcuların simülasyon modelinde oluşturulması gerçekleştirilmiştir. Havalimanının her biriminde geçirecek zaman için yine metamodelde üretilen Zaman (Time) verisinin içindeki veriler kullanılarak belirlenmiştir. Uçuş Kapısı (Gate) model elemanı ile de simülasyon modelinde yolcunun terminalde ulaşabileceği son nokta belirlenmiştir.

Bu uygulamada, modelleme editörünün sağladığı ortam sayesinde hazırlanabilen yerleşim planı ve akış şeması modellerinin verileri ve görsel öğeleri Simio programına özel olarak girilerek basit seviyede bir havalimanı terminali için yolcu akış simülasyon modelinin oluşturulabileceği gösterilmiştir. Aşağıdaki Tablo 6.1’de örnek model için yapılan sınıf düzeyindeki eşleştirmeler gösterilmiştir.

Tablo 6.1: Örnek model dönüşümü için eşleştirmeler.

Kaynak modeldeki model elemanı	Hedef modeldeki model elemanı	Kullanılan araçtaki Türkçe-İngilizce karşılığı
Başla (Start)	Terminal Giriş	Üreteç (Source)
Giden yolcu servisleri	Servis	Servis (Server)
Check-in	Check-in	Servis (Server)
Bilet Kontrol	Bilet Kontrol	Servis (Server)
Uçuş Salonu	Uçuş Salonu	Bekleme (Delay)
Bitir (Stop)	Uçuş Kapısı (Gate)	Bitir (Sink)

Hem tasarlanan modelleme editörü sonucunda üretilen modellerde hem de simülasyon modellerinde XML ve XMI formatının kullanılması, kaynak modellerin model dönüşümlerinin yapılabilmesinde önemli bir avantaj sağlamaktadır. Bu sayede, geliştirilen metamodel kullanılarak üretilen terminal yerleşim planı ve yolcu akış şeması modellerinden simülasyon model dönüşümü için otomatik kod üretimi daha kolay sağlanabilecektir. Böylece, modellerden elde edilen verilerin otomatik olarak doğrudan simülasyon modeline aktarılması sağlanarak farklı olasılıkların da denendiği çok daha hızlı ve verimli simülasyon modelleri elde edilebilecektir.

YEDİNCİ BÖLÜM

SONUÇLAR VE ÖNERİLER

Havalimanlarını kullanan kişi sayısının ve uçuş sayılarının artması havalimanlarındaki yoğunluğu da artırmaktadır. Buna paralel olarak havalimanlarının mevcut kapasiteleri ihtiyaca karşılık verememekte ve bu durum havalimanı hizmetlerinde aksaklıklara neden olabilmektedir. Yeni yapılacak havalimanlarının, bu alanları kullanacak kişi sayısının artması ve bu alanlardaki beklentilerin değişmesi nedeniyle, inşa edilmeden önce detaylı bir analiz ve doğru bir planlama ile tasarlanması gerekir. Bu analiz ve planlama çalışmaları, sonradan düzeltilmesi zor hataların önceden fark edilip engellenmesi için oldukça önemlidir. Bu bağlamda havalimanı terminallerinin inşaat başlamadan önce modellenmesi, maliyetleri azaltacak ve bu yapıların daha kaliteli ve daha verimli hale gelmesini sağlayacaktır.

Dünyada ve ülkemizde havacılık alanındaki çalışmaların önem kazandığı görülmektedir. Örneğin, havalimanı mühendisliği (airport engineering) gibi yeni meslek gruplarının oluşması ve uluslararası literatürde, havacılıkla ilgili ve özel olarak havalimanı terminal modellemesi ve simülasyonu ile ilgili çalışmaların son yıllarda artması bu alana olan ilgiyi göstermektedir. İncelenen çalışmalarda havalimanı modellemesinde farklı meslek gruplarının birlikte çalışabilecekleri bir ortam bulunmadığı gözlemlenmiştir. Bu çalışma ile havalimanı terminal binası tasarımının hazırlanmasında mimar, havalimanı mühendisi, inşaat mühendisi, endüstri mühendisi, yazılım mühendisi ve havalimanı yöneticisi gibi meslek gruplarının birlikte çalışmasına olanak sağlanmıştır.

Literatürdeki çalışmalar incelendiğinde havalimanları ile ilgili modelleme ve simülasyon çalışmalarında kullanılan araçlar bulunmasına rağmen, model güdümlü geliştirme yaklaşımı ile geliştirilmiş ve alana özgü genişletilebilir modelleme elemanları sunan bir ortam bulunamamıştır. Bu doğrultuda akademik çalışmalarda

tespit edilen havalimanlarındaki yolcu hareketleri ve havalimanı tasarım kriterleri dikkate alınarak, tespit edilen yolcu istek ve ihtiyaçları doğrultusunda yapısal ve fonksiyonel gereksinimler belirlenerek havalimanı terminal birimlerinin özellikleri, birbirleri ile ilişkileri, sınırları belirlenerek detayları ile bir metamodel geliştirilmiştir. Geliştirilen metamodel ile simülasyon modeline dönüşümü için gerekli bazı parametreleri de içeren modeller geliştirilerek, model dönüşümü için hazır modeller üretilebileceği gösterilmiştir.

Yapılan bu çalışmada, havalimanı terminal modellemesi için hazırlanan metamodelin IATA, ICAO gibi uluslararası standartlara uygun olmasına dikkat edilmiştir. SHGM'nin yayınlamış olduğu, IATA tarafından yayımlanan "Airport Development Reference Manual" dokümanının Türkçeye tercümesi olan Havaalanı Yolcu Terminalleri Tasarım Esasları ve Havaalanı Planlama kılavuzları vb. kaynaklardan yararlanılmıştır. Metamodelde kullanılan havalimanı terminal birimlerinin tamamında SHGM'nin yayınlamış olduğu yayınlardan yararlanılmıştır. Kara tarafı ya da hava tarafı gibi havalimanı terminal birimlerinin dışındaki alanların genel isimleri için de, yine bu yayınlardan ve DHMİ'nin yayınladığı havacılık terimleri sözlüğünden faydalanılmıştır.

Geliştirilen metamodel iki bölümden meydana gelmektedir. Birinci bölümde terminal metamodeli ile terminal yerleşim planı görsel olarak oluşturulabilmektedir. Terminal yerleşim modelinde standartlarda belirtilen havalimanı birimleri tasarlanan modelleme editöründe kullanılabilir. Farklı bir birim eklemek istenirse, modelleme editöründeki özel alan model elemanı kullanılarak veya metamodelde müdahale edilerek yeni bir birim eklenebilmektedir. Bu birim için kural veya kısıtlama tanımlaması da yapılabilmektedir. Terminal yerleşim planı oluşturulurken, metamodelde tanımlanan ilişkiler, kurallar ve kısıtlamalar sayesinde standartlara uygun bir şekilde tasarım yapılabilir.

Terminal yerleşim planı tasarımında kullanılan kısıtlama ve kurallar için de yine SHGM'nin yayınlarında belirtilen standartlardan faydalanılmıştır. Bu bağlamda havalimanı gelen ve giden yolcu terminal bölümlerinin aynı zamanda yurtiçi ve uluslararası yolcu terminal bölümlerine fiziksel olarak ayrılmasını gerektiren kuralların tamamı uygulanmıştır. Havalimanı yolcu terminal binasında bulunması gereken birimler ile kat farkına göre eklenmesi gereken birimlerin kontrolleri gibi

kurallar da metamodelde tanımlanmıştır. Böylece yaklaşık 60 adet birimin yerleşimi ile ilgili kural ve kısıtlamanın modelde otomatik olarak uygulanması sağlanmıştır. Ancak metamodelde mesafe kontrollerinin yapılmasında bazı kısıtlamalar yaşanmıştır. Bu nedenle bu türdeki yaklaşık 15 adet kuralın modelde otomatik olarak kontrolü sağlanamamıştır. Genel olarak terminal yerleşim planı tasarımında bu kural veya kısıtlamaların uygulanmasında belirlenen standartlar %80 oranında sağlanmıştır. Kullanıcıların editör tarafından otomatik olarak yönlendirilmesiyle, tasarım esaslarına riayet edilmesi sağlanmıştır. Geliştirilen metamodelde standartlarda belirtilen, gerçekleştirilebilen ve gerçekleştirilemeyen birim, kısıt ve kuralların tablosu Ek E'de verilmiştir.

İkinci bölümde ise yolcu akış metamodeli ile yolcu akış modelleri oluşturulabilmektedir. Metamodelin bu bölümünde, havalimanı birimleri ve akış modeli elemanları ile bunlar arasındaki bağlantı tanımları ve bağlantı kuralları bulunmaktadır. Yolcu akış modeli, bu tanım ve kurallar ile akış şeması kriterleri dikkate alınarak ve yolcunun gidebileceği yerler kısmen kontrol edilerek çizilebilmektedir. Yolcu akış metamodelinde, havalimanı birimlerinde tanımlanan değişkenler ile simülasyon modelleri için kullanılacak zaman gibi parametre girişleri yapılabilmektedir. Metamodelin yolcu akış modelindeki kurallar ve kısıtlamalar, akış şeması kuralları ve yolcunun terminal girişinden uçağa kadar veya uçaktan terminal çıkışına kadar kullanabileceği birimler ve sıraları dikkate alınarak oluşturulmuştur. Bu kural ve kısıtlamalar dışında, rutin ve geçerli yolcu akışı dışındaki düşük olasılıklı hatalı girişimlerin modellenmesi engellenmiştir. Örneğin, giden yolcunun hava tarafından kara tarafına geçerek uçuştan vazgeçmesi ve terminali terk etmesi seçeneği mevcut değildir. Ancak, model güdümlü geliştirme yaklaşımı ile metamodelin kolay bir şekilde güncellenebilmesi ve model editörünün kolaylıkla yeniden üretilebilmesi mümkündür. Bu ve benzeri farklı özellikler için metamodele müdahale edilerek istenilen eklentiler rahatlıkla yapılabilir.

Sonuç olarak, bu tezde önerilen metamodel ve geliştirilen model editörü alana özgü bir modelleme dili sunmakta ve havalimanı terminal modellemesi için yeni bir yaklaşım ortaya atmaktadır. Yeni akademik çalışmalara temel olabilecek biçimsel bir metamodel sunarak otomatik kod dönüşümleri ile simülasyon modeli oluşturulması konusunda katkı sağlamaktadır. Mevcut akademik çalışmalarda, belirlenen bir havalimanına özgü yolcu akış modelleri bulunmaktadır. Bunlara bağlı oluşturulan

simülasyon modelleri de yine belirli bir havalimanı için kullanılabilmekte ve oluşturulan simülasyon modellerinin verileri de her havalimanı için özel olarak girilmektedir. Bu çalışmada tasarlanan metamodel kullanılarak, farklı havalimanları için farklı modeller daha kısa sürede oluşturulabilecek ve simülasyon modeline dönüşüm için gerekli verinin tamamına yakını sağlanmış olacaktır. Böylece, havalimanlarındaki yolcu akışlarının ve terminal yerleşimlerinin daha kaliteli ve hızlı bir şekilde modellenmesi ve analiz edilebilmesi sağlanacaktır.

KAYNAKLAR

- [1] Devlet Hava Meydanları İşletmesi (DHMI), (2014). Uçak, Yolcu, Yük Serisi ve Tahminleri. <http://www.dhmi.gov.tr/getBinaryFile.aspx?Type=14&dosyaID=1049>, (Son Erişim: Mayıs, 2015).
- [2] Balcı, O., (2001). A Methodology For Certification Of Modeling And Simulation Applications. *ACM Transactions on Modeling and Computer Simulation*, 11(4). 352-377.
- [3] Guizzi, G., Murino, T., Romano E., (2009). A Discrete Event Simulation To Model Passenger Flow In The Airport Terminal. *Proceedings Of The 11th International Conference On Mathematical Methods and Computational Techniques In Electrical Engineering*. 427-434.
- [4] Takakuwa, S., Oyama T., (2003). Modeling People Flow: Simulation Analysis Of International-Departure Passenger Flows In An Airport Terminal. *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation (WSC '03)*.1627-1634.
- [5] Verbraeck, A., Valentin E., (2002). Simulation Building Blocks For Airport Terminal Modeling. *Proceedings of the 2002 Winter Simulation Conference*. 1199-1206.
- [6] Çetinkaya D., Verbraeck, A., (2011). Metamodeling and Model Transformations in Modeling and Simulation. *Proceedings of the 2011 Winter Simulation Conference, IEEE*. 3048-3058.
- [7] Mittal, S., Risco-Martin J. L., (2013). *Netcentric System of Systems Engineering with DEVS Unified Process*. CRC Press, Taylor & Francis Group.

- [8] Sarjoughian, H. S., Markid A. M., (2012). EMF-DEVS Modeling. *Proceedings of the Symposium on Theory of Modeling and Simulation: DEVS Integrative M&S Symposium (TMS-DEVS '12)*, Society for Modeling & Simulation International.
- [9] Selic, B., (2003). The Pragmatics of Model-Driven Development. *IEEE Software*, 20(5),.19-25.
- [10] Seidewitz, E. (2003). What Models Mean. *IEEE Software*, vol. 20. 26-32.
- [11] Kleppe, A., Warmer, J., W. Bast, (2003). MDA Explained - The Model Driven Architecture: Practice and Promise. *Addison-Wesley*.
- [12] Devlet Hava Meydanları İşletmesi (DHMİ), (2011). *Havacılık Terimleri Sözlüğü*. Ankara: Devlet Hava Meydanları İşletmesi Genel Müdürlüğü Yayınları.
- [13] Arusoğlu, Ö., (2010). Havaalanı Yolcu Hareket Simülasyon Model Önerisi. Yüksek Lisans Tezi. İstanbul Teknik Üniversitesi/Fen Bilimleri Enstitüsü, İstanbul.
- [14] Mutlu, Ö. F., (2011). Havalimanı ve Havaalanı Terminal Yapılarında Yolcu Beğenisinin Araştırılması ve Ölçülmesi. Yüksek Lisans Tezi. Gazi Üniversitesi/Fen Bilimleri Enstitüsü/Mimarlık bölümü, Ankara.
- [15] Acımert, M. T., (2003). Çağdaş Havaalanı, Terminal Planlamasında Akış Olgusu, Plan Tipleri, Tip Seçimi Ve Kriterleri. Yüksek Lisans Tezi. Mimar Sinan Güzel Sanatlar Üniversitesi / Fen Bilimleri Enstitüsü/ Mimarlık bölümü, İstanbul.
- [16] Freivalde, L. , Lace, L., (2008). Improvement of Passenger Flow Management In An Airport Terminal, *Proceedings of the 5th International Scientific Conference Business and Management*. 659-664.
- [17] Lazzaroni, M., (2012). Modeling Passenger And Baggage Flow At Vancouver Airport. <http://www.simio.com/case-studies/Modeling-Passenger-and-Baggage-Flow-at-VancouverAirport>, (Son Erişim: Mayıs, 2015).

- [18] Federal Aviation Administration (FAA), (2015). <http://www.faa.gov/>, (Son Eriřim: Mayıs, 2015).
- [19] Norman J. Ashford, Saleh Mumayiz, Paul H. Wright. (2011). *Airport Engineering: Planning, Design and Development of 21st Century Airports* (4. Baskı). Ch15 Airport modelling and simulation (s.602-657).
- [20] Korul, V., (2001). Havaalanlarının Çevre İle İliřkileri Yönetimi ve Türkiye’de Uluslararası Trafığe Açık Havaalanlarında Çevre Kirlilięi Uygulamalarının Analizi. Doktora Tezi. Anadolu Üniversitesi/Sosyal Bilimler Enstitüsü. Eskişehir. 4-5.
- [21] Sivil Havacılık Genel Müdürlüęü (SHGM), (2010). *Havaalanlarında Kapasite Kriterleri*. Ankara: Sivil Havacılık Genel Müdürlüęü Yayınları.
- [22] International Air Transport Association (IATA), (2015), <http://www.iata.org/>, (Son Eriřim: Mayıs, 2015).
- [23] International Civil Aviation Organization (ICAO), (2015), <http://www.icao.int/>, (Son Eriřim: Mayıs, 2015).
- [24] Sivil Havacılık Genel Müdürlüęü (SHGM), (2009). *Havaalanı Yolcu Terminalleri Tasarım Esasları*. Ankara: Sivil Havacılık Genel Müdürlüęü Yayınları.
- [25] Yeřilürdü F., Diri, B., (2006). Model-Güdümlü Mimari Kullanılarak Bir Konum Sunucusu Yazılımının Geliřtirilmesi. 2. *Ulusal Yazılım Mühendislięi Konferansı*.
- [26] Ülgür, İ., N., Akgül, Ö. (2011). Gömülü Sistemler için Otomatik Kod Üretimi-Beyaz Eřya Uygulaması. 5. *Ulusal Yazılım Mühendislięi Sempozyumu - Uyms'11*.
- [27] Özhan G., Oęuztüzün H., (2006). Model-Integrated Development of HLA Based Field Artillery Simulation. *Proceedings of 2006 European Simulation Interoperability Workshop*.
- [28] Allilaire F., Jouault F., (2007). A Simple Illustration of Model to Model Transformation. *ATLAS group, INRIA & University of Nantes* .

- [29] Çelik, T., Sütbaşı, R., İmre, K., (2005). HLA İçin Modelleme, Otomatik Kod Üretme, İzleme ve Sınama Araçları. *Ulusal Savunma Uygulamaları Modelleme ve Simülasyon Konferansı, USMOS-2005*.
- [30] Çetinkaya, D., Verbraeck A., Seck M. D., (2015). Model Continuity in Discrete Event Simulation: A Framework for Model-Driven Development of Simulation Models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. 25: 3, Article 17.
- [31] İşler, V., (2010). Modelleme Simülasyon. *Savunma Sanayii Gündemi*.
- [32] İpekkan, Z., (2010). Simülasyon Tabanlı Tedarik Yönteminin Savunma Tedarik Sürecinde Kullanımı, *Savunma Sanayii Gündemi*.
- [33] Tuğlular, T., Şentuna, N., Tuğ, A., K., Dağhan G., Güven C., (2014). Lojistik Merkez Konteyner Terminal Yönetimi Yazılımı İçin Alana Özgü Dil Geliştirimi Ve Kullanımı. *Ulusal Yazılım Mühendisliği Sempozyumu*.
- [34] Mernik, M., Heering, J., Sloane, A. M., (2005). When and How to Develop Domain-Specific Languages. *ACM Comput. Surv.* 37, 316–344.
- [35] Atkinson, C., Kühne, T., (2003). Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, 20(5):36-41.
- [36] Çetinkaya, D., (2013). Defining and Transforming Conceptual Models into Simulation Models by Using Metamodels and Model Transformations. Doktora Tezi. Delft University of Technology/ The Netherlands.
- [37] Atkinson C., Gerbig R., (2012). Multi-level Modeling and Ontology Engineering Environment. *Proceedings of the 2nd International Master Class on Model-Driven Engineering: Modeling Wizards*.
- [38] Sendall, S., Kozaczynski, W., (2007). Model Transformation – the Heart and Soul of Model-Driven Software Development. *IEEE Software*, vol. 20, no. 5, 42-45.
- [39] Kardaş, G., (2008). Anlamsal Web Ortamında Çalışan Çok Etmenli Sistemlerin Model Güdümlü Geliştirilmesi. Doktora Tezi. Ege Üniversitesi / Fen Bilimleri Enstitüsü/Uluslararası Bilgisayar Anabilim Dalı, İzmir.

- [40] Generic Modeling Environment, (2015). GME Manual and User Guide, <https://forge.isis.vanderbilt.edu/gme/GME>, (Son Eriřim: Mayıs, 2015).
- [41] Egesoy, A., Topalođlu Y., (2009). Yazılım Geliřtirmede Anlam Sorunları ve Model GÜdümlü Yaklařım. *9.Ulusal Yazılım Mühendisliđi Sempozyumu*.
- [42] Sivil Havacılık Genel Müdürlüđü (SHGM), (1987). *Havaalanı Planlama Kılavuzu*. Ankara: Sivil Havacılık Genel Müdürlüđü Yayınları.
- [43] Simulation Modeling Framework Based on Intelligent Objects (SİMİO), (2015), <http://www.simio.com/index.php>, (Son Eriřim: Mayıs, 2015).

EKLER

Ek A Model ve Metamodel Elemanları Çeviri Tablosu

Tablo A.1: MetaModel'deki Terminal ve Yolcu Akış modelinin İngilizce isimlendirilen birimlerinin metamodel ile modelleme kullanıcı ara yüzündeki isimleri ve karşılıkları.

Tablo A.2: Model düzenleyicisinde görünmeyen ancak metamodelde tanımlanan İngilizce kısıt, değişken, atom veya model isimleri.

Ek B Esenboğa Havalimanı Terminal Binası Yerleşim Planı

Şekil B.1: Esenboğa Havalimanı Kara Tarafı Giden Yolcu Bölümü.

Şekil B.2: Esenboğa Havalimanı Kara Tarafı Gelen Yolcu Bölümü.

Şekil B.3: Esenboğa Havalimanı Hava Tarafı Gelen Yolcu Bölümü.

Şekil B.4: Esenboğa Havalimanı Hava Tarafı Giden Yolcu Bölümü.

Ek C Özel Bir Havalimanı Terminal Binası Yerleşim Planı

Şekil C.1: Kara Tarafı Giden-Gelen Yolcu Bölümü.

Şekil C.2: Hava Tarafı Giden-Gelen Yolcu Bölümü.

Ek D GME Metamodelleme Aracının Kullanımı

Şekil D.1: Metamodelde kök klasöre model eklenmesi.

Şekil D.2: Atom ve model nesne denetçisi (object inspector) penceresi.

Şekil D.3: Bağlantı ve bağlayıcı elemanlarının kullanımı.

Şekil D.4: Kısıtlama yapısı.

Şekil D.5: Kısıtlamaların modelde aktif yapılması.

Şekil D.6: Oluşturulan model için üretilen dosyaların listesi.

Ek E Geliştirilen Metamodelin Standartlara Uyumluluk Tablosu

Tablo E.1: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Yolcu Terminalleri Tasarım Esasları kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu

Tablo E.2: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu

Tablo E.3: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Yolcu Terminalleri Tasarım Esasları kitabındaki gerçekleştirilemeyen birim, kısıt ve kuralların tablosu

Tablo E.4: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilemeyen birim, kısıt ve kuralların tablosu

Ek A Model ve Metamodel Elemanları Çeviri Tablosu

MetaModel'deki *Terminal* ve *Yolcu Akış* modelinin İngilizce isimlendirilen birimlerinin metamodel ile modelleme kullanıcı ara yüzündeki isimleri ve karşılıkları Tablo A.1'de verilmiştir.

Tablo A.1: Model elemanları çeviri tablosu.

Havalimanı Birimi	MetaModel'deki Adı	Model Editöründeki Adı
Kat	Floor	Floor
İç Hat	Domestic	Domestic
Dış Hat	InternationalLine	International
Check-İn	CheckBanko	Checkin
Bagajın Teslimi	BagajTeslim	Baggage Claim
Elektronik Check	ElektronicCheck	Electronic Checkin
Terminale Giriş	Terminalinput	Enterance
Terminal Çıkış	TerminalOutput	Exit
Pasaport Kontrol	Pasaport	Passport Kontrol
Terminal Kayıp Eşya	TerminalKayipEsysa	LostProperty
Yürüyen Merdiven	Escalator	Escalator
Asansör	Elevator	Elevator
Yürüyen Yolu	WalkWay	WalkWay
Merdiven	Stair	Stair
Silah Teslim	SilahTeslim	Claim Desk
Emanet	Emanet	Safety Cabinet
Postane	Postahane	Post Office
Bagaj Paketleme	BagajPaketleme	Baggage Packing
Polis	PolisMerkezi	Police Office
Metro	Metro	Transportation
Gümrük Kontrol	Customs	Customs
Banka	Bank	Bank Office
Lostra Salonu	Lostra	Lostra
Eczane	Eczane	Pharmacy

Tablo A.1 (Devam): Model elemanları çeviri tablosu.

Terminal Kapısı	TerminalGate	TerminalGate
Otel	Otel	Hotel info
Uçuş Firması Ofisleri	AirlineOffice	Airline Office
Genel Kapı	Gate	Gate
Bilet Kontrol	BiletControl	Ticket Control
Bilet veya Pasaport Kontrol	TicketOrPassportControl	Control
Çocuk Oyun Alanı	ChildGameArea	Game Area
Tekerlekli Sandalye Servisi	WheelChairService	Wheelchair Help
Danışma	Information	Information
Güvenlik	Security	Security
Araç Kiralama	RentaCar	RentaCar
Tuvaletler	Wc	WC
Engelli Tuvaletleri	WcEngelli	WC Disabled
Çocuk Bakım Odası	WcAndChildcare	WC Childcare
Bagaj Arabası	BaggageTrolleys	Baggage Trolleys
Vip Salonu	VipHall	VipHall
Serbest Alan	FreeSpace	FreeSpace
İbadet Yerleri	Worship	Worship
Otopark Parası Ödeme Noktası	ParkPayment	CarPark Payment
Pilot Dinlenme Salonları	PilotDinlenmeSalonu	PilotHall
Konferans Salonları	ConferanceHall	Conference Hall
Ayrım Çizgisi	YatayCizgi	Wall-H
Ayrım Çizgisi	DikeyCizgi	Wall-V
Kitapevi	Kitapevi	Bookstore
Döviz Bürosu	ChangeOffice	ChangeOffice
Hediyeli Mağaza	HediyelikShop	Gifts

Tablo A.1 (Devam): Model elemanları çeviri tablosu.

Gazete Bayi	Gazete	Newspaper
Yiyecek İçecek Alanları	CafeRest	Cafe-Rest
Diğer Mağazalar	OtherShop	Shop
Duş Salonları	ShowerBath	Shower
Kuaför	Kuaför	HairDresser
Kuru Temizleme	KuruTemizleme	DryCleaner
Spa Alanları	Spa	Spa
Gümrüksüz Alışveriş	DutyFreeShop	DutyFree
Transit Geçiş Noktası	TrasitGecis	Transit Passenger
Dinlenme Alanı	Lounge	Lounge
Uçuş Salonları	FlightsHall	Flights Hall
Sigara Salonları	SigaraSalon	SmokingArea
Uçak Resmi	UcakYatay	Plane1
Uçağa Biniş Kapısı	FlightGate	FlightGate
Bagaj Alım Salonu	BaggageClaim	Baggage Hall
Uçak	Ucak	Plane2
Nokta	Point	Point
Başla	Start	Flow Start
Bitir	Split	Flow Stop
Uçak Çıkış	FlightExit	FlightExit
Uçak Biniş	FlightGate	FlightGate

Model düzenleyicisinde görünmeyen ancak metamodelde tanımlanan İngilizce kısıt, değişken, atom veya model isimleri Tablo A.2’de verilmiştir.

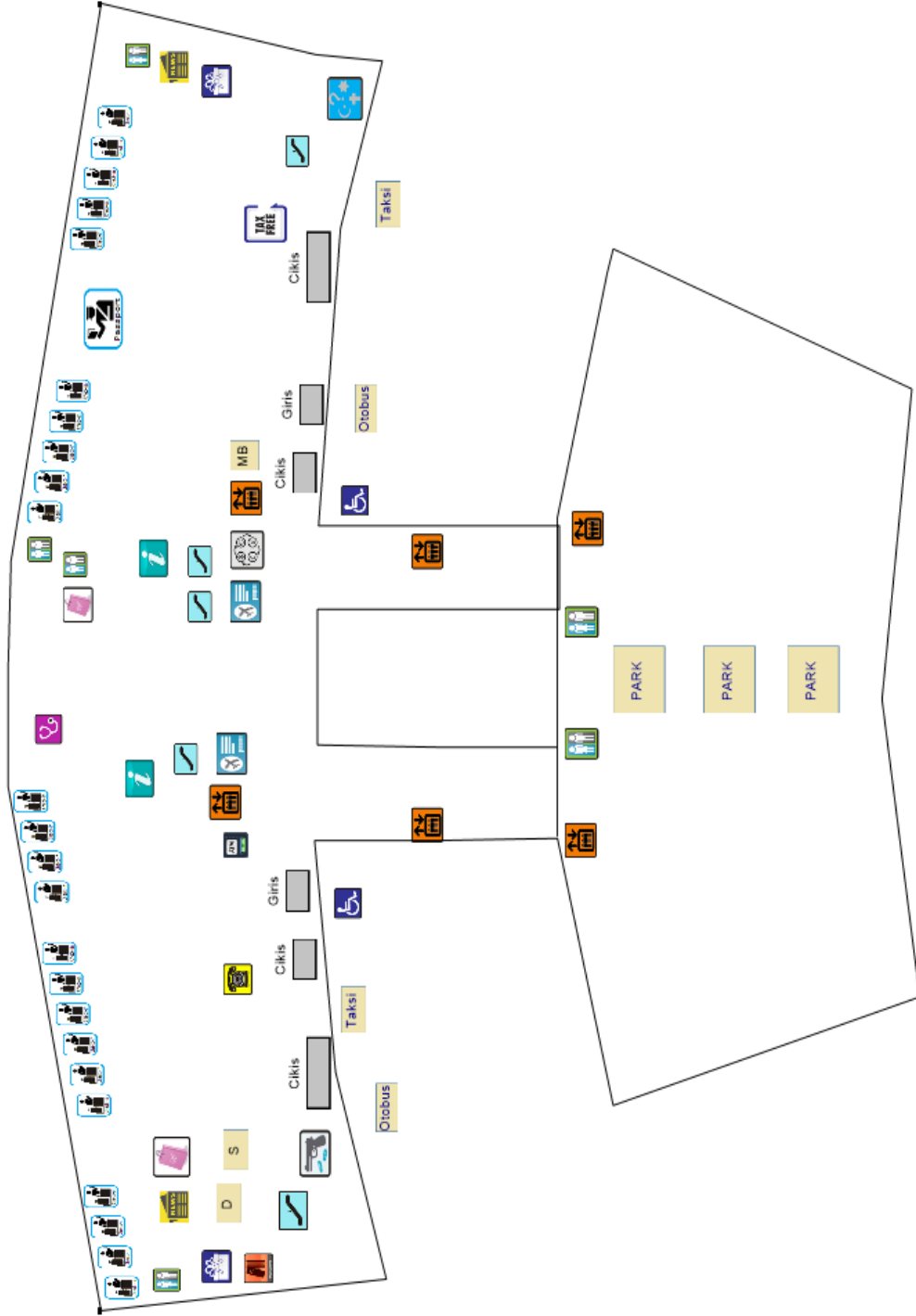
Tablo A.2: Metamodel elemanları çeviri tablosu.

Model Elemanı İsmi	Türkçe Karşılığı
AirPart	Hava Bölümü
AirServiceDomAndInt	İç ve Dış Hat Hava Tarafı Servisleri
AirServiceInternational	Dış Hat Hava Tarafı Servisleri
AreaAndHalls	Alanlar ve Salonlar
ArivalAndDeparture	Gelen ve Giden
BankoID	Banko Numarası
CheckType	Check-in Tipi
CheckinName	Check-in İsmi
DomesAndInt	İç ve Dış Hat
DomesOrInt	İç veya Dış Hat
DutyShopID	Gümrüksüz Mağaza Numarası
FirmName	Firma İsmi
FloorNumber	Kat Numarası
GateID	Kapı Numarası
GateName	Kapı İsmi
HallID	Salon Numarası
IATACode	IATA Numarası
ICAOCode	ICAO Numarası
InternationalTerminal	Uluslararası Terminal
LandService	Kara Tarafı Servisleri
LandAndAirService	Kara ve Hava Tarafı Servisleri
LandOtherTerminalService	Kara Tarafı Diğer Terminal Servisleri
LandSpecialService	Kara Tarafı Özel Servisler
LandTerminalService	Kara Tarafı Terminal Servisleri
Other	Diğer
OtherTerminalService	Diğer Terminal Servisleri

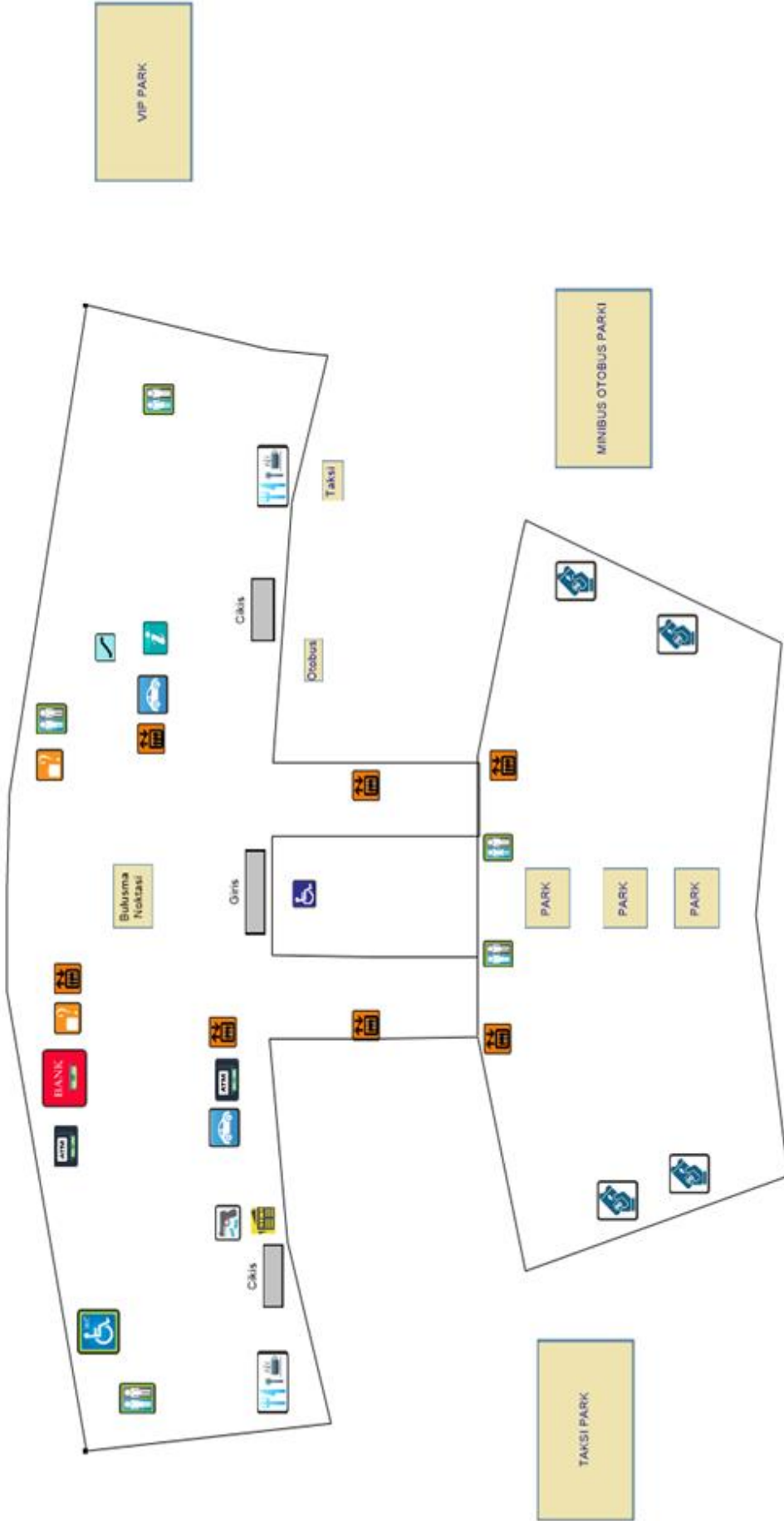
Tablo A.2 (Devam): Metamodel elemanları çeviri tablosu.

PassengerTerminalFlightService	Yolcu Terminal Uçuş Servisleri
PassportControl	Pasaport Kontrol
RequiredFields	Gerekli Alanlar
Shop	Mağaza
TerminalName	Terminal İsmi
TerminalService	Terminal Servisleri
TerminalSize	Terminal Boyutu
ConnectArrivalAirLand	Hava ve Kara Tarafı Gelen Bağlantı
ConnectArrivalLand	Kara Tarafı Gelen Bağlantı
ConnectDeparturesLandAir	Hava ve Kara Tarafı Giden Bağlantı
ConnectStart	Başla Bağlantısı
ConnectStop	Bitir Bağlantısı
ConnectionTransfer	Transfer Bağlantısı
DecisionInfo	Karar Bilgisi
Description	Açıklama
Flow	Akış
Join	Topla
JoinSingleDst	Topla Tek Hedef
JoinSingleSrc	Topla Tek Kaynak
PassengerArrivalAir	Hava Tarafı Gelen Yolcu
PassengerArrivalLand	Kara Tarafı Gelen Yolcu
PassengerDeparturesAir	Hava Tarafı Giden Yolcu
PassengerDeparturesLand	Kara Tarafı Gelen Yolcu
SeviceName	Servis İsmi
Split	Dağıt
SplitSingleDst	Dağıt Tek Hedef
SplitSingleSrc	Dağıt Tek Kaynak
Time	Zaman
PassengerFlow	Yolcu Akışı

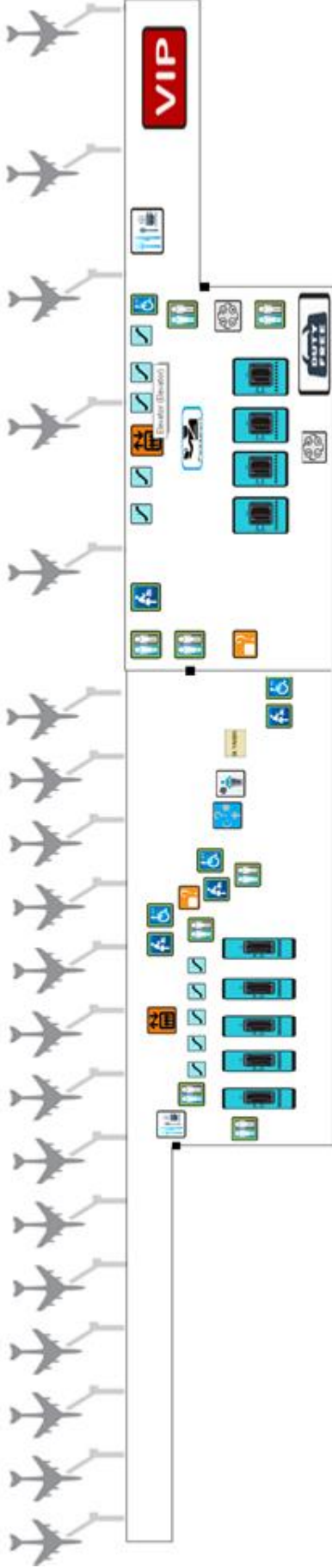
Ek B Esenboğa Havalimanı Terminal Binası Yerleşim Planı



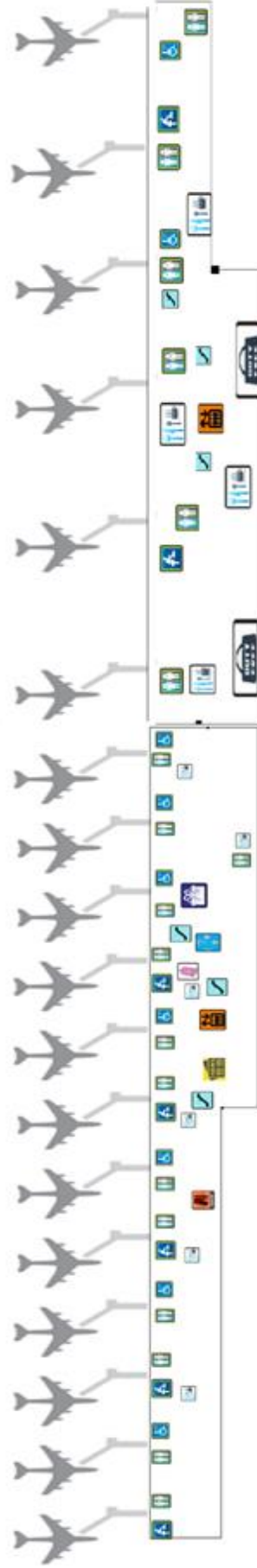
Şekil B.1: Esenboğa Havalimanı kara tarafı giden yolcu bölümü.



Şekil B.2: Esenboğa Havalimanı kara tarafı gelen yolcu bölümü.



Şekil B.3: Esenboğa Havalimanı hava tarafi gelen yolcu bölümü.



Şekil B.4: Esenboğa Havalimanı hava tarafi giden yolcu bölümü.



Şekil C.2: Hava tarafı giden-gelen yolcu bölümü.

Ek D GME Metamodelleme Aracının Kullanımı

Bu bölümde kullanılan GME aracı ile ilgili Türkçe kaynak bulunamaması nedeniyle, ileride yapılacak çalışmalarda yardımcı olması amacıyla GME programının kullanımına yönelik kısa bilgi verilmiştir.

D.1. GME Programı ile MetaGME Metamodeli Oluşturulması

GME’de metamodellere değerler dizisi (paradigm) denir. GME programını kullanarak bir metamodel projesi oluşturmak için ilk olarak GME programında kayıtlı metamodellerden *MetaGME* seçilir. Oluşturulan metamodel dosyası *mga* uzantılı olarak kaydedilecektir.

Mga uzantılı dosya GME programı ile açıldıktan sonra, metamodelin ismi kök klasörün (RootFolder) ismi değiştirilerek tanımlanır. Bu alandaki tanımlama, modelde görünen kök klasörün, metamodel’in XML olarak kaydedildiği dosyanın ve GME programı ilk açıldığında paradigma seçimi (select paradigm) ekranındaki oluşturulan metamodelin de ismi olacaktır. Bu işlemler yapıldıktan sonra MetaGME derleme (Interpreter) butonuna basılarak hazırlanan metamodelin derlenmesi ve XML’in de içinde bulunduğu program için gerekli dosyaların oluşturulması sağlanacaktır.

D.2. GME Programı ile Oluşturulan Metamodelin Kullanımı

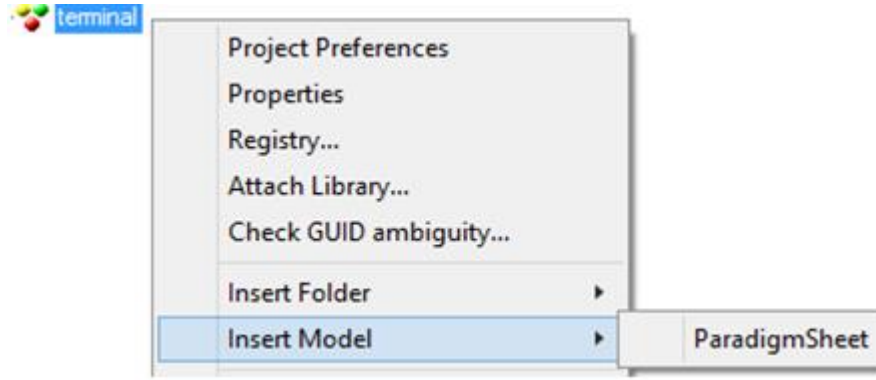
Oluşturulan metamodel içerisinde bölüm tarayıcı (part browser) bölümünde, metamodelde kullanılacak elemanlar bulunmaktadır. Bu elemanlar sınıf diyagramı, değişkenler (attributes), kısıtlamalar (constraints), görüntüleme (visualization) başlıkları altında bulunmaktadır.

D.2.1. Sınıf Diyagramı

Atom; içerisinde herhangi bir nesne bulunmayan, kural, kısıtlama veya özellik eklenebilen nesnelere temsil eden en basit modelleme aracıdır.

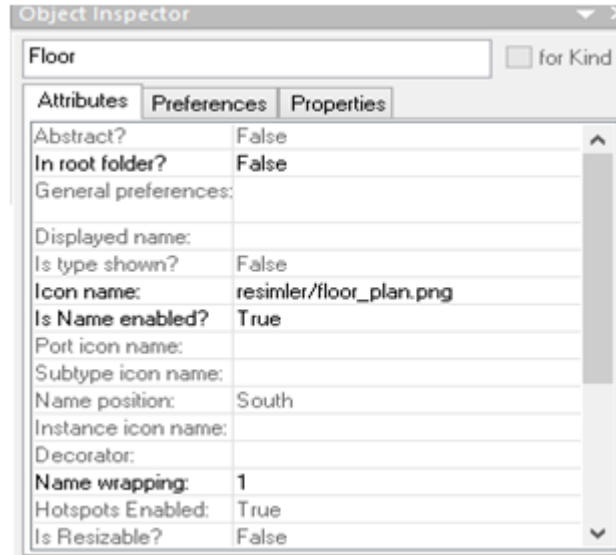
Model; içerisinde başka bir model, atom veya diğer nesnelere barındıran ve bunları temsil eden modelleme elemanıdır. Örneğin bu uygulamada, *check-in bankosu*, *cafe-restaurant* birer atom iken, kat, atom ve diğer nesnelere içinde bulunduran, terminalin herhangi bir katını temsil eden modeldir. Bu nedenle metamodel

oluşturulduktan sonra ilk olarak eklenmesi gereken de kök klasörde tanımlanacak bir model oluşturmaktır. Metamodelde göre üretilecek model tasarımları bu kök klasöre yerleştirilecek model tanımlamalarına göre oluşturulacaktır. Kök klasöre eklenecek modeller, Şekil D.1’de görüldüğü gibi kök klasörün üzerine sağ tıklanarak, model ekle (insert model) sekmesinden paradigma formu (paradigm sheet) seçilerek oluşturulur.



Şekil D.1: Metamodelde kök klasöre model eklenmesi.

Atom gibi Model için de Şekil D.2’de görüldüğü gibi kural ve nitelik eklenebilir.



Şekil D.2: Atom ve model nesne denetçisi (object inspector) penceresi.

Tanımlanan atom ve modeller için nesne denetçisi penceresindeki değişkenler ve tercihler (preference) alanlarındaki seçenekler ile özellik tanımlaması yapılırken,

model ekranındaki nesne görünümleri için de tanımlamalar yapılabilir. Bu tanımlamalardan bazıları şöyledir, *Soyut mu* (abstract) ve *Kök Klasörde mi* (in root folder) ile, eklenen elemanın sırası ile model ekranında ve kök klasörde görünüp görünmemesinin seçimi yapılır. *Görünen İsmi* (displayed name) seçeneği ile model ekranındaki ismi tanımlanır. Kullanıcı ara yüzünde nesne ikonlarının belirlenmesi, ikon ismi alanına görünmesi istenen ikon resminin yolunun yazılması ile olur. Eklenen nesnelerin boyutunun kullanıcı tarafında değiştirilebilmesi ise *Boyutları değişebilir mi* (is resizable) seçeneğinin aktif edilmesi ile olur.

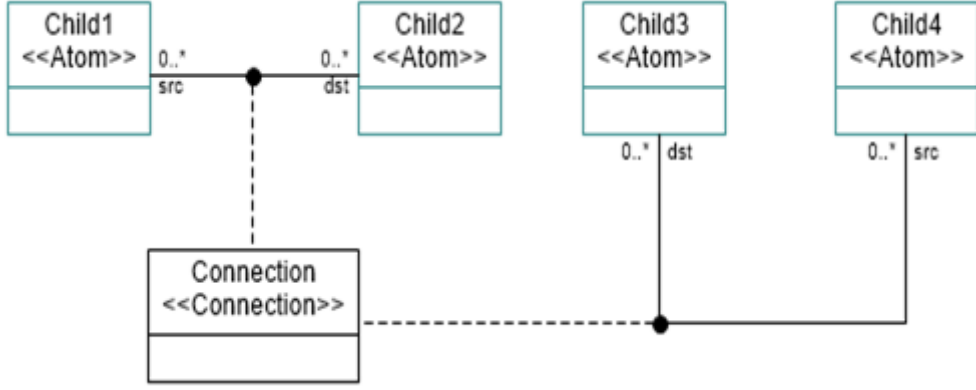
D.2.2. Kalıtım

Atom veya model için tanımlanan özellikleri başka bir atom veya modele aktarılmasını sağlayan sınıf diyagramı elemanıdır. \triangle Simgesiyle gösterilen kalıtım sayesinde, bir atom veya model elemanının özellikleri, kendine bağlanan diğer tüm atom veya model elemanlarına aktarılması sağlanır. Böylece örneğin, birden fazla atom ve model için tanımlanan değişkenler, ortak bir atom veya model altında toplanarak, diğer modelleme elemanları için de otomatik olarak tanımlanabilir.

D.2.3. Bağlantı

Model tasarım ekranında kullanıcıların modelleme elemanları ile yapacağı bağlantıların tanımlanması, metamodelleme aşamasında bağlantı atomu ile yapılabilmektedir. Bağlantı atomu, birbirleri ile bağlantı kurulabilecek elemanlar arasında köprü vazifesi görüp, tanımlanan kısıtlama ve grafiksel öğelerden faydalanılarak, belli kurallar çerçevesinde bağlantıları metamodelde tanımlanmasını sağlar.

Bağlayıcı (Connector) atomu ise bağlantı yapılacak model elemanları arasındaki bağı oluşturup, bağlantı atomuna ilişirir. Şekil D.3'te görüldüğü gibi bir bağlantı ile aynı kurallar çerçevesinde birden fazla bağlayıcı ilişiririp, bağlantı tanımlanabilmiştir. Model tasarımındaki bağlantının yönü ve şekli de, bağlantı tanımlama aşamasında çıkan uyarı pencerelerinde kaynak (source), hedef (destination) seçimi ile oluşturulabilmektedir.



Şekil D.3: Bağlantı ve bağlayıcı elemanlarının kullanımı.

D.2.4. Değişkenler

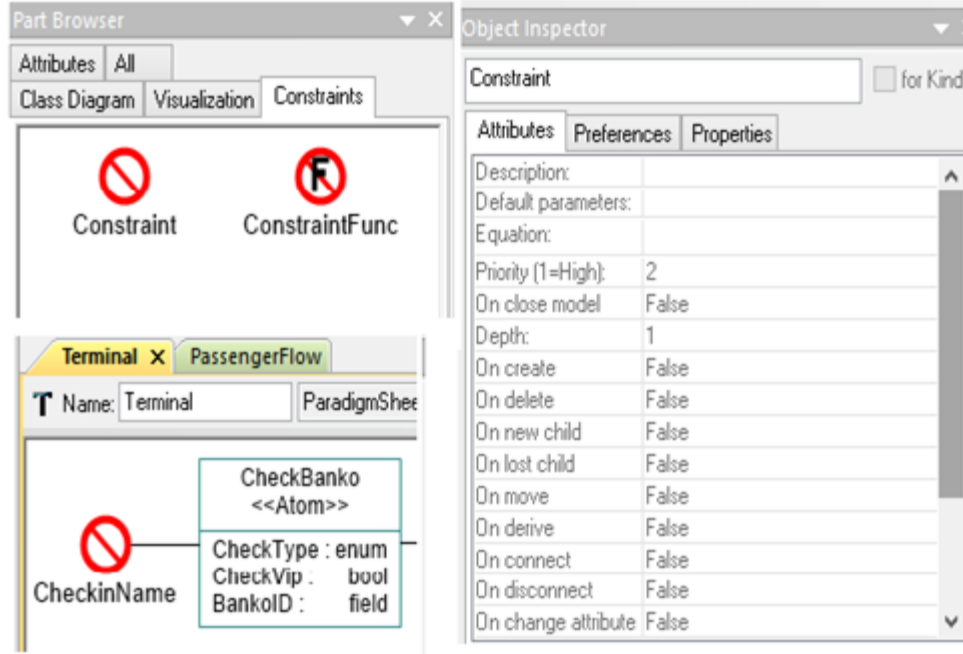
Değişkenler, modelleme elemanları için özellik tanımlamada kullanılır. Bu bölüm altında mantıksal değişken (BooleanAttribute), numaralanmış değişken (EnumAttribute), alan değişken (FiledAttribute) olmak üzere üç adet birim bulunmaktadır. Mantıksal değişken eklendiği birime model tasarım ekranında değişken penceresinde doğru (true) ya da yanlış (false) ifadelerinden birinin seçilmesini sağlar. Numaralanmış değişken, model tasarım ekranında kullanıcının, metamodelde tasarımcının belirlediği liste halindeki özelliklerden birini seçmesini sağlar. Alan değişken ise özellik tanımlamasını, kullanıcının klavyeden metin girişi ile yapılmasını sağlar.

D.2.5. Kısıtlamalar

GME programında model elemanları için, görsel öğelerin veya özellikler panellerinin sunduğu kısıtlamalar dışında farklı bir kural ve kısıtlama tanımlanması gerekirse kısıtlama yapısı kullanılır.

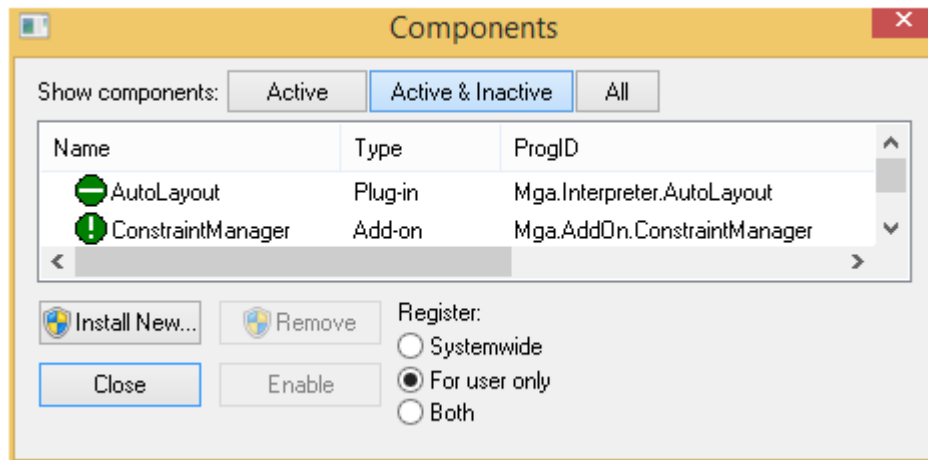
Kısıtlama model elemanı tasarım ekranına eklendikten sonra, kuralın kapsayacağı veya kısıtlamanın geçerli olacağı model elemanı ile bağlantısı sağlanır. Daha sonra eklenen kısıtlamanın nesne denetçisi panelinin eşitlik (equation) alanına OCL kodları kullanılarak kurallar yazılır. Bu panelde; Şekil D.4'te görüldüğü gibi öncelik (priority) alanında kuralın diğer örtüşen kurallara göre öncelik sırası belirlenirken, tanım (description) alanında da hata sonucu çıkan uyarı penceresinde kullanıcı için tanımlanan açıklama yer alır. On ifadesi ile başlayan değerler de ise bu kuralın hangi olaylar sonucu gerçekleşeceği tanımlanır. Örneğin oluşturulurken (on

create) ile kısıtlamanın bağlı olduğu nesne modele eklenmesi sırasında kural kontrolü gerçekleşirken, değiştirilirken (On change) değişkenleri ile ise, nesne için tanımlanan özelliklerde değişiklik meydana geldiği zaman kuralın kontrol ettirilmesi sağlanır.





Şekil D.4: Kısıtlama yapısı.

Kısıtlama tanımlamalarının, model oluşturma ekranında aktif olabilmesi için GME programının araç (tools) menüsü altındaki kayıt bileşenleri (register components) sekmesi altında çıkan pencere için Şekil D.5'te görüldüğü gibi kısıt yöneticisi (ConstraintManager) bölümünün aktif edilmesi gerekmektedir.

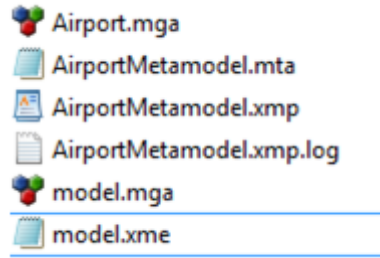


Şekil D.5: Kısıtlamaların modelde aktif yapılması.

Bölüm tarayıcısı alanındaki sınıf diyagram, değişkenler, kısıtlamalar için görünümlerinin otomatik en makul bir şekilde nesne yerleşimi  ikonu ile sağlanırken, nesnelere arası bağlantılar ve bağlantı kaldırma işlemleri  ikonları ile yapılabilmektedir.

D.3. Oluşturulan Metamodel ile Model Oluşturma

Tanımlanan metamodel son hali ile derlendikten sonra model editörü otomatik olarak oluşturulabilir. Oluşturulan model editörü kullanılarak geliştirilen metamodelle uyumlu modeller tanımlanabilir. Oluşturulan modelde dosya (file) menüsünde dışarı aktar (export XML) basılarak Şekil D.6'daki gibi modelin XML dosyası oluşturulabilir. Oluşturulan XML dosyası daha sonra GME ile içeri aktar (import XML) ile açılarak metamodel üzerinde değişiklikler yapılmış olsa bile en son hali ile tekrar açılıp güncellemeler yapılabilir (GME, 2015).



Şekil D.6: Oluşturulan model için üretilen dosyaların listesi.

Ek E Geliştirilen Metamodelin Standartlara Uyumluluk Tablosu

Geliştirilen Metamodelde, SHGM'nin Havaalanı Yolcu Terminalleri Tasarım Esasları kitabı ve Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu sırası ile Tablo E.1 ve Tablo E.2'de verilmiştir.

Geliştirilen Metamodelde, SHGM'nin Havaalanı Yolcu Terminalleri Tasarım Esasları kitabı ve Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilemeyen kısıt ve kuralların tablosu sırası ile Tablo E.3 ve Tablo E.4'de verilmiştir.

Tablo E.1: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Yolcu Terminalleri Tasarım Esasları kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu.

No	Kaynak Metin
1	Memleket dışına ve memleket içine devlet kontrol hizmetleri (gümrük kontrolü). (Sayfa 12)
2-4	Terminal sistemleri (asansörler, yürüyen merdivenler ve yürüyen bantlar), seviye değiştirirken trolleyleri boşaltma ve yeniden yükleme ihtiyacı bulunmaksızın yolcu hareketine olanak vermelidir. (Sayfa 14)
5-7	Hareket gidiş, çeşitli kamusal veya kamusal olmayan alanlardan oluşmaktadır. Bunlar, dolaşım ve bekleme alanlarını, kamu tesislerini, havayolu bilet satış & hizmet bankolarını ve yolcu kabul (checkin) tesislerini içermektedir. (Sayfa 16)
8-10	Dolaşım, sıraya girme ve bekleme için, ve gidiş dahilindeki çeşitli tesislere yönelik alan gereklilikleri aşağıdaki hususlardan etkilenmektedir: Pasaport Kontrol, Güvenlik Konumları, E-bilet kullanımı. (Sayfa 17)
11	Ortak gidiş yolcu salonları, uçuş kapısı salonları ve transit salonları, terminallerde üç ayrı alan olarak, kombinasyon halinde, veya tek bir bütün olarak görülebilir. (Sayfa 17)
12-13	Hava tarafında işleme tabi tutulması gereken, hava tarafına varan ve başka bir uçuşa aktarılan aktarma yolcuları. Her zaman hava tarafında kalması gereken, hava tarafına varan ve yolculuklarına aynı uçuş ile devam eden transit yolcuları. (Sayfa 18)
14	Uluslararası havaalanlarının çoğunda, yolcu kabul işlemleri erken yapılmış ve devlet kontrollerinden geçmiş olup, halen uçağa binmek üzere çıkış kapısı bilgilerini bekleyen çıkış yolcularını barındıracak ortak bir gidiş yolcu salonu sağlanmalıdır. (Sayfa 18)

Tablo E.1 (Devam): Geliştirilen Metamodelde, SHGM yayınları Havaalanı Yolcu Terminalleri Tasarım Esasları kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu.

15	Aşağıdaki fonksiyonlar, ortak gidiş yolcu salonuna dahil edilmek üzere dikkate alınmalıdır: Yolculara yardım sağlayacak havayolu danışma masaları. (Sayfa 18)
16-17	Devlet kontrollerine yönelik gereklilikler ve bu kontrollerin yeri (kara tarafı / hava tarafı) ve bunların yolcu akışı üzerindeki etkisi de göz önünde bulundurulmalıdır. (Sayfa 19)
18	Sirkülasyon alanları, Pier/uydu sonuna ulaşıldığında daralabilir. Pier/uydu sirkülasyon alanları, yürüyüş geçitlerine olanak vermeli ve her iki tarafta bagaj arabaları ile yürüten yolcuları, tekerlekli sandalyeleri ve araç trafiğini barındırabilecek yeterli saha sağlamalıdır. (Sayfa 21)
19-22	Bagajın teslim alma alanına teslimi, yolcu teslimatından önce gerçekleşmesi gerekirken, bu alana para çekme makinaları ve tuvaletler gibi destek tesislerinin, ve de bagaj arabalarının, ücretsiz olarak yerleştirilmesi akıllıca olur. (Sayfa 22)
23	Her havaalanının kendi münferit tasarım özelliklerine sahiptir. Ancak tüm bu tasarımlar, 5 ayrı terminal konseptine indirilebilir: Pier/parmak, Linear, Açık apron, Uydu, Kompakt modül üniteli terminal. (Sayfa 23)
24	Kesin tesis tipine karar vermedeki başlıca bir faktör, kurumsal kimlik konusudur. Birçok havayolları, kendilerinin reklamını yapma ve kendilerini tanıtmaya çalışırken, yolcu kabulü bankolarından, CIP & VIP salonlarına ve bağımsız uçuş kapısı bekleme odalarına uzanan münhasır tesisleri kullanmayı tercih etmektedir. (Sayfa 26)
25	Yolcu işlem sistemlerine ilişkin üç düzenleme mümkündür. (Tek Seviye, ½ Seviye, İki Seviye) (Sayfa 38)
26	Küçük bir havaalanı, uçuşları ve yolcuları kendi pist ve terminal altyapısı düzenlemesi yardımıyla işleme tabi tutma gücü ile tanımlanmaktadır. Tipik olarak, 'küçük' havaalanı olarak tarif edilen bir tesis, azami yılda 1 milyon yolcu (MPPA) işleminden geçirebilecektir. (Sayfa 44)
27	Küçük havaalanının aynı şekilde sınırlı perakende, yolcu ve personel kamusal dinlenme alanları sağlaması gerekecektir. (Sayfa 44)
28	Otomasyon, bilgisayarların, yazıcıların bir yolcunun güzergahı boyunca birçok noktada kurulması anlamındadır. Bunlar, aşağıdakileri kapsayabilir: Bilet/satış bankoları, Check-in bankoları, Uçağa biniş kapıları, Aktarma bankoları, Danışma masaları. (Sayfa 49)

Tablo E.1 (Devam): Geliştirilen Metamodelde, SHGM yayınları Havaalanı Yolcu Terminalleri Tasarım Esasları kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu.

29-30	Herhangi bir CUSS tasdikli uygulamasını (örneğin araç kiralama, otel rezervasyonları, vs.) desteklemektedir. (Sayfa 49)
31-33	Bölüm K içinde tanımlanan çerçeve felsefeye göre, yolcu akış güzergahları: Giden yurtiçi yolcuları için bir akış güzergahı ve uluslararası yolcular için bir başka akış güzergahı. Gelen yurtiçi yolcuları için bir akış güzergahı ve uluslararası yolcular için bir başka akış güzergahı. Giden yolcular, güvenlik kontrol noktalarından sonra, gelen yolculardan fiziksel olarak ayrılmak zorundadır. (Sayfa 71)
34-35	Kısım K3'ün gerekliliklerine göre, ve kontrol şartlarının gerekli kıldığı durumlarda, terminallerin hava tarafında uluslararası ve yurtiçi yolcuların ayrılması için düzenleme yapılmak zorundadır. (Sayfa 72)
36-37	Kısım K3'ün şartlarına göre ve yurtiçi trafiği için, ayırım mutlaka zorunlu olmamalı, fakat belirli durumlar veya düzenlemeler altında gelen ve giden yolcuların ayrılması gerekebilir. (Sayfa 72)
38-57	Ayrıcalıklı alanlar aşağıdakiler dahil olabilir, ancak bunlarla sınırlı değildir. Bankalar, Yiyecek & İçecek, Kahve barları, Postane, Restoranlar, Berber/güzellik salonu, Masaj, Kafeterya, Çocuk Odası, Duty Free, Döviz Bürosu, Hediyelik eşya, Giyim, Paralı telefonlar, Bagaj sarma, Tıbbi hizmetler, Kitaplar, Ayakkabı Boyama, Fitnes Merkezi sauna, Gazete Bayii. (Sayfa 80)

Tablo E.2: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilebilen birim, kısıt ve kuralların tablosu.

No	Kaynak Metin
58-59	Bulunması istenen servis tipleri trafik yoğunluğuna ve birçok diğer piyasa koşullarına göre değişir. Bu tür alanların programlanmasında, her havaalanı için ayrı olmak kaydıyla çalışan ve çalışması muhtemel işletmeler ve ilgililerle yapılan tartışmalar baz alınmalıdır. Belirleyici ana başlıklar aşağıda verilmiştir: Araba kiralama gişeleri, Emanet bagaj dolapları .. (Sayfa 223)
60	Genellikle birçok havaalanına gidiş ve geliş ulaşımı iki temel şekilde gerçekleşir, bunlar özel otomobiller ve toplu taşıma araçlarıdır. Bu iki başlık altında da genellikle taksiler ve otobüsler kullanılır. Bazı havaalanlarına otobüs dışında büyük transit sistemler de (örneğin; tren, metro) servis sağlar. (Sayfa 251)

Tablo E.3: Geliştirilen Metamodelde, SHGM yayınları yayınları Havaalanı Yolcu Terminalleri Tasarım Esasları kitabındaki gerçekleştirilemeyen birim, kısıt ve kuralların tablosu.

No	Kaynak Metin
1	Aktarmalara yönelik uçakların arasındaki, ve farklı şekillerdeki yüzey erişim sistemleri ile hem giden hem de gelen yolculara ait uçakların arasındaki mesafeler minimum düzeyde tutulmalıdır. 300 metreyi aşan mesafeler, yürüyen bantlarla donatılmalıdır. (Sayfa 14)
2	Münferit hareket uçuş kapısı salonlarına yönelik saha gerekliliklerini ve uçuş kapısı salonu genişliklerini tanımlayan tablo. (Sayfa 20)
3	Ana fonksiyonların arasındaki (yani otoparktan check-in/bagaj teslim alma ve check-in/bagaj teslim alma noktasından uçuş kapısı salonuna) önerilen maksimum yürüyüş mesafesi 300 m'dir. (Sayfa 72)

Tablo E.4: Geliştirilen Metamodelde, SHGM yayınları Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilemeyen birim, kısıt ve kuralların tablosu.

No	Kaynak Metin
4	Gazete/kitap ve büfeler: yıllık yolcu miktarının 200 000'ni aştığı havaalanlarında genellikle bu servis birbirinden bağımsız olarak verilir ve daha az trafiği olan havaalanlarındaki servislerle birleştirilebilir. En az 14 m2 ve yıllık bir milyon giden yolcu için 56 ile 65 m2 arasında yere ihtiyaç vardır. (Sayfa 223)
5	Hediye ve giyim eşyası satan dükkanlar: küçük havaalanlarında bazı eşyalar gazete büfelerinde satılabilir, fakat yıllık yolcu sayısının 1 000 000'u aştığı havaalanlarında ayrı olarak oluşturulan hizmetler gerekli hale gelir. Yıllık 1 000 000 giden yolcu için 56 ile 65 m2 arasında yere ihtiyaç duyulur. (Sayfa 223)
6	Berber ve ayakkabı boyacısı: bazı büyük havaalanlarında yıllık 1 000 000 giden yolcu için bir sandalyeye ihtiyaç duyulur. Minimum hizmet için 14 m2 olmak kaydıyla her sandalye başına 10 ile 11 m2 arasında yer ayrılmalıdır. (Sayfa 223)
7	Araba kiralama gişeleri: kullanılacak alan şirket sayısına göre değişiklik gösterir. Yıllık 1 000 000 giden yolcu için 33 ile 37 m2 arasında yere ihtiyaç duyulabilir. (Sayfa 223)
8	Otel telefonlarını da içeren sergiler: yıllık 1 000 000 giden yolcu için 8 ile 9 m2 arasında yer ayrılmalıdır. (Sayfa 223)

Tablo E.4 (Devam): Geliştirilen Metamodelde, SHGM yayınları Havaalanı Planlama Klavuzu kitabındaki gerçekleştirilemeyen birim, kısıt ve kuralların tablosu.

No	Kaynak Metin
9	Sigorta, gişe ve makineler dahil: yıllık 1 000 000 giden yolcu için 14 ile 16 m2 arasında yer ayrılmalıdır. (Sayfa 223)
10	Emanet bagaj dolapları: yıllık 1 000 000 giden yolcu için 6.5 ile 7.5 m2 arasında yer bırakılmalıdır. (Sayfa 223)
11	Genel Telefonlar: yıllık 1 000 000 giden yolcu için 9 ile 10 m2 arasında yer ayrılmalıdır. (Sayfa 223)
12	Satış makineleri genellikle ana trafik akışına sirkülasyon alanı içerisinde engel olmaması açısından grup halinde yerleştirilmelidir. Satış Makineleri için yıllık 1 000 000 giden yolcu için 4.5 m2 alan ihtiyacı vardır. (Sayfa 223)
13	Umumi tuvalet: bölgesel gruplar, ülkeler, vb. için uygulanabilecek kodlara uygun olarak bina kullanıcılarına göre boyutlandırılmalıdır. Ayrılan yerlerin büyüklüğü 500 pik-saat yolcuları (gelen ve giden) için 139 m2'den 167 m2'ye kadar değişiklik gösterebilir. Büyük merkez havaalanlarında ise bu rakam yıllık 1 000 000'lük kullanım için 120m2'ye kadar düşebilir. (Sayfa 224)
14	Yolcu Danışma: bu kapsamda verilen hizmetler büyük değişiklik gösterebilir. Yer gereksinimleri, yıllık kullanıcı sayısının bir milyondan fazla olduğu havaalanları hariç, 7.4 ile 9.3 m2 arasında değişir. (Sayfa 224)
15	Yaya yürüme yolları üzerinde herhangi bir engel bulunmamalı ve en azından 1.5 m. genişliğinde olmalıdır. (Sayfa 227)

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: İsmet CAMCI

Uyruğu: T.C.

Doğum Yeri ve Tarihi: Kulu-18.10.1986

Medeni Hali: Evli

Adres: Yeşiltepe Mah. Keçiören/ANKARA

E-Posta Adresi: ismet.camci@stu.thk.edu.tr

İletişim (Telefon) : 0312 527 00 35



EĞİTİM

Lise : Fethiye Kemal Mumcu Anadolu Lisesi (Ankara) - 2004

Lisans : Gazi Üniversitesi/Bilgisayar Sistemleri Öğretmenliği (Ankara)-2009

Yüksek Lisans : Türk Hava Kurumu Üniversitesi/Elektrik ve Bilgisayar Mühendisliği (Ankara) - Devam Ediyor.

İŞ DENEYİMİ

Teknoloji Barkod Sistemleri, Ankara

Yaz Stajı 2007

Yazılım Geliştirme, Teknik Destek

Milli Eğitim Bakanlığı Mesleki ve Teknik Anadolu Liseleri

Bilişim Teknolojileri Öğretmenliği 2009-2011 ile 2012-2015

Balıkesir Astsubay Meslek Yüksek Okulu

Bilgisayar Öğretmenliği 2011-2012