

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

**ONTOLOJİ TABANLI VERİTABANLARINDA
TERCİHLERİN ELE ALINMASI İÇİN GENEL BİR
MODEL**

DİLEK TAPUCU

Tez Danışmanı: Yrd. Doç. Dr. Murat Osman ÜNALIR

Bilgisayar Mühendisliği Anabilim Dalı

Bilim Dalı Kodu: 619.02.00

Sunuş Tarihi: 10-02-2010

Bornova-İZMİR

2010

Dilek TAPUCU tarafından Doktora tezi olarak sunulan “Ontoloji Tabanlı Veritabanlarında Tercihlerin Ele Alınması İçin Genel Bir Model” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı : Yrd.Doç. Dr. Murat Osman ÜNALIR

Raportör Üye : Prof. Dr. Oğuz DİKENELLİ

Üye : .Prof. Dr. Sıtkı AYTAÇ

Üye : Doç. Dr. Ahmet KOLTUKSUZ

Üye : Yrd. Doç. Dr. Rıza Cenk ERDUR

ÖZET

ONTOLOJİ TABANLI VERİTABANLARINDA TERCİHLERİN ELE ALINMASI İÇİN GENEL BİR MODEL

TAPUCU, DİLEK

Doktora Tezi, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Yrd. Doç.Dr. Murat Osman ÜNALIR

Şubat 2010, 157 sayfa

Günümüz uygulamalarında öne çıkan Anlamsal Web araştırmalarının bir kısmı, mevcut ontolojik veri miktarındaki artışa kişiselleştirilmiş çözüm yaklaşımları sunmaktadır. Çözümler genellikle tercihlerin sorgulanması, sorgu cevaplarının sıralanması ve önceliklendirilmesi yönündedir. Ancak tercihlerin alan bilgisinden bağımsız olarak tanımlanması, benzer tercihlerin paylaşılması, yeniden kullanması ve yönetilebilmesi yönünde eksikler bulunmaktadır. Bu nedenle tez çalışması, kullanıcı tercihlerini temsil edebilen genel, genişletilebilir bir model sunma yönünde Veritabanı ve Anlamsal Web araştırma alanlarında yapılan çalışmaları bir araya getirmekte, literatürde önerilen tercih tiplerini ayrı ayrı ele alarak bir çatı altında toplamaktadır. EXPRESS veri modelleme dili kullanılarak oluşturulan tercih modeli, üst-şema yaklaşımı ile işlenir ontolojik verilere anlamsal olarak bağlanmaktadır. Oluşturulan üst-şema, tercihlerin anlamsal tariflerini desteklemek üzere ontolojileri ve genişletilmiş veri örneklerini yöneten OntoDB Ontoloji Tabanlı Veritabanı (OTV) üzerinde gerçekleştirilmektedir. Kişisel tercihlerin sorgulanabilmesine yönelik olarak ise OntoDB (OTV) mimarisi üzerinde kullanılan OntoQL ontoloji temelli sorgulama ve tanımlama diline, tercih temelli sorgulama gerçekleştirim yaklaşımı getirilmektedir. Mevcut dile yapılan yeni bir eklenti ile geliştirilen tercih modeli kullanılmakta, bu sayede tercihlere ontolojik veriler üzerinden tercih temelli sorgulama gerçekleştirimi ile ulaşılmaktadır. Önerilen eklentinin gerçekleştirimi sonrasında OntoDB üzerinde turizm alanına yönelik bir örnek uygulama çalışma kapsamında sunulmaktadır.

Anahtar sözcükler: Ontoloji Temelli Veritabanı (OTV), Kişiselleştirme, Anlamsal Web, Ontoloji, Kullanıcı Tercihleri.

ABSTRACT**A GENERIC MODEL FOR HANDLING PREFERENCES IN
ONTOLOGY BASED DATABASES****TAPUCU, Dilek**

PhD in Computer Engineering

Supervisor: Yrd. Doç.Dr. Murat Osman ÜNALIR

February 2010, 157 pages

With the emergence of the Semantic Web, the amount of available ontological data (or instances) has increased. To allow personalized access to this information has become a crucial necessity. Users are overwhelmed by the numerous results provided in response to their requests. In order to be usable, these results must often be sorted and filtered. The capture and exploitation of user preferences have been proposed as a solution to this problem. However, the existing approaches usually define preferences for a particular application. Thus, it is difficult to share and reuse the handled preferences in other contexts. Our approach, which defines a sharable and generic model to represent user preferences, based on several models proposed in the Databases and the Semantic Web communities. It incorporates several types of preferences proposed in the literature, but are treated separately. Our idea is to address preferences as a modular way by linking them to ontologies for describing the semantics of the data, handled by the applications. The novelty of our approach is that the defined preferences are attached to the ontologies, which describe the semantic of the data, manipulated by the applications. The preference model is formally defined using the EXPRESS data modeling language, which ensures a free ambiguity definition. Moreover, the proposed is implemented using Ontology Based Databases (OBDB) system, that manages both ontologies and extended data instances, in order to support a semantic description of preferences. Our second approach, which defines preferences in terms of ontologies, that describe the semantics of handled data, provides a mechanism for querying with preferences. Thus, an extension to existing ontology based query languages is proposed, for querying ontological data with preferences. The proposed extension has been implemented onto the OntoDB OBDB, associated to the OntoQL query language, and the approach is illustrated through a case study in the tourism domain.

Keywords: Ontology-based Database (OBDB), Personalization, Semantic Web, Ontology, User preferences.

TEŞEKKÜR

Tez çalışması süresince sonsuz ilgi ve sabır ile değerli katkılarını hiç bir zaman esirgemeyen danışmanlarım sayın Yrd.Doç.Dr. Murat Osman ÜNALIR ve sayın Prof. Dr. Yamine AIT AMEUR başta olmak üzere, tez izlemelerimde beni yalnız bırakmayan sayın hocalarım Prof. Dr. Oğuz DİKENELLİ ve Doç.Dr. Ahmet KOLTUKSUZ'a, İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü Bölüm Başkanı sayın Prof. Dr. Sıtkı AYTAÇ ve Rektör Yardımcımız sayın Prof. Dr. Halis PÜSKÜLCÜ'ye, İYTE Bilgisayar Mühendisliği Bölümü ailesinin değerli öğretim üyesi ve elemanlarına, Ege Üniversitesi Bilgisayar Mühendisliği Bölümü ailesine, Fransa'da bulunan Poitiers Üniversitesi ENSMA/LISI Laboratuvarında ki hocalara sayın Prof. Guy PIERRA, Ladjel BELLATRECHE ve çalışma arkadaşlarım Stephane JEAN ve Gayo DIALLO ile tüm laboratuvar ekibine, 'İkili Doktora Bursu Programı' ile doktora çalışmamı destekleyen kuruma ve hiç bir zaman eksilmeyen sevgi ve desteklerinden dolayı aileme ÇOK TEŞEKKÜR EDERİM.

İÇİNDEKİLER

Sayfa

ÖZET	V
ABSTRACT.....	vii
TEŞEKKÜR	ix
İÇİNDEKİLER	xi
ŞEKİLLER DİZİNİ	xiii
ÇİZELGELER DİZİNİ	xv
ÇİZELGELER DİZİNİ	xvi
SİMGELER VE KISALTMALAR DİZİNİ	xviii
SÖZLÜK DİZİNİ.....	xx
1. GİRİŞ.....	1
1.1 KONUNUN ÖNEMİ	2
1.2 ÇALIŞMANIN HEDEFLERİ.....	4
1.3 TEZİN YAPISI.....	4
2 TEMEL KAVRAMLAR VE ALAN YAZIN.....	7
2.1 TERCİH KAVRAMI	7
2.2 KİŞİSELLEŞTİRME.....	8
2.3 VERİTABANI YÖNETİM SİSTEMLERİ VE KİŞİSELLEŞTİRME.....	9
2.3.1 Tercih Temelli Yaklaşımlar ve İlişkisel Veri Sorgulama.....	9
2.3.2 Değerlendirme.....	13
2.4 ANLAMSAL WEB VE KİŞİSELLEŞTİRME	15
2.4.1 Ontoloji Temelli Kullanıcı Tercihleri ve Ontolojik Veri Sorgulama	16
2.4.2 Değerlendirme.....	18
2.5 DİĞER ARAŞTIRMA ALANLARI VE KİŞİSELLEŞTİRME YAKLAŞIMLARI.....	20
2.5.1 Web Madenciliği.....	20
2.5.2 Veri Ambarları	21
2.5.3 İnsan Bilgisayar Etkileşim Sistemleri	21
2.5.4 Arama Motorları	22
2.5.5 Öneri Sistemleri	23
2.5.6 Kullanıcı Profilleri	23
2.6 İLGİLİ ÇALIŞMALARA AİT GENEL BİR DEĞERLENDİRME.....	24
3 BİLGİ MODELLERİ	27
3.1 MODEL GÜDÜMLÜ MİMARİ.....	27
3.1.1 Meta-Object Facility (MOF)	29
3.1.2 MDA Seviyeleri	30
3.2 MODEL TABANLI YAZILIM GELİŞTİRME.....	32
3.2.1 Modelleme Dilleri	32
3.2.2 Ontoloji Tanımlama Dili (ODM)	34
3.3 DEĞERLENDİRME	38

4	ONTOLOJİ TEMELLİ VERİTABANLARI.....	40
4.1	OTV MİMARİLERİ.....	40
4.2	3. TİP ONTOLOJİ TEMELLİ VERİTABANI: ONTODB	43
4.3	ONTOQL DİLİ	44
4.4	DEĞERLENDİRME	46
5	TERCİH YÖNETİM ÇERÇEVESİ.....	49
5.1	TERCİH YÖNETİM ÇERÇEVESİ VE KAYNAK TANIMI	51
5.2	TERCİH ÜST-ŞEMASI.....	52
5.2.1	Tercih Niteleyici: Preference_URI Tanımı	54
5.2.2	Yorumlanabilir Tercihler: Interpreted_Preferences Tanımı	56
5.2.3	Bağlam Temelli Tercihler: Context_Based_Preference Tanımı	64
5.3	TERCİH BAĞI.....	66
5.4	DEĞERLENDİRME	66
6	ONTODB VE TERCİH MODELİ ENTEGRASYONU.....	71
6.1	TERCİHLERİN ONTODB ÜZERİNDE MODELLENMESİ	72
6.1.1	OntoDB Üzerinde Ontoloji Gösterimi	72
6.2	ONTODB MİMARİSİNİN TERCİH MODELİ İLE GENİŞLETİLMESİ	75
6.2.1	Preference_URI	75
6.2.2	Numeric_Preference	76
6.2.3	Boolean_Preference	77
6.2.4	Interval_Preference	77
6.2.5	Fuzzy_Preference.....	78
6.2.6	Enumerated_Preference	79
6.3	ONTODB MİMARİSİ İÇİN TERCİH - ONTOLOJİ BAĞI.....	79
6.4	TERCİHLERİN ONTODB ÜZERİNDE SORGULANMASI.....	82
6.4.1	Preferring Operatorü Sözdizimi	82
6.4.2	Sorgu Yorumlama	84
6.5	DEĞERLENDİRME	85
7	UYGULAMA GERÇEKLEŞTİRİMİ.....	88
7.1	ÖRNEK-1: KİŞİSEL TERCİHLERİN ALAN ONTOLOJİSİ ÜZERİNDEN ELDE EDİLMESİ.....	89
7.1.1	Alan Ontolojisi Geliştirme	89
7.1.2	Tercih Üst-Şeması Örnekleri	92
7.1.3	Tercih Bağı Oluşturulması	98
7.1.4	Değerlendirme	99
7.2	ÖRNEK-2: ONTODB ÜZERİNDE TERCİH TEMELLİ SORGULAMA GERÇEKLEŞTİRİMİ	102
7.2.1	Alan Ontolojisi Geliştirimi	102
7.2.2	OntoDB Üzerinde Tercih Modeli Gösterimi.....	103
7.2.3	Tercih Modeli ve Alan Ontolojisi Entegrasyonu	104
7.2.4	Sorgulama Örnekleri	105
7.2.5	Değerlendirme	107
8	SONUÇLAR.....	110
9	EKLER	129
EK-1	EXPRESS-G Grafik Notasyonu	129
EK-2	ONTOQL TANIMLAMA DİLİ	133
EK-3	EXPRESS ŞEMA DÖKÜMANI	136
EK-4	TERCİH YÖNETİM ÇERÇEVESİ İÇİN UML ŞEMA	139

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
3.1 Geleneksel Nesne Yönetim Grubu (OMG) Modelleme Altyapısı (Atkinson and Kühne, 2003).....	30
3.2 Kaynak Gösterimi.....	35
3.3 ODM için Kaynak.....	36
4.2 1.Tip OTV Mimarisi.....	39
4.2 2.Tip OTV Mimarisi.....	40
4.3 3.Tip OTV Mimarisi.....	40
4.4 OntoDB Mimarisi.....	41
5.1 Tercih Yönetim Çerçevesi.....	47
5.2 Tercih Yönetim Çerçevesi için Kaynak Tanımı.....	48
5.3 Tercih Üst-Şeması.....	50
5.4 Tercih Bağı.....	63
5.5 Tercih Yönetim Çerçevesi be MDA seviyeleri.....	64
6.1 OntoDB Mimarisi.....	67
6.2 OntoDB için Ontoloji ve Mantık Model Gösterimi.....	69
6.3 Genişletme öncesi OntoDB.....	76
6.4 Tercih Üst-Şeması ile Genişletme sonrası OntoDB Mimarisi.....	76
7.1 Otel Ontolojisi.....	86
7.2 Otel Ontolojisi Örnekleri.....	87

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Veritabanı Araştırma Alanı ve Tercih Yaklaşımları.....	14
2.2 Anlamsal Web Araştırma Alanı ve Tercih Yaklaşımları.....	19
2.3 Araştırma Alanları ve Tercih Yaklaşımları.....	25
3.1 MDA Seviyeleri (Bezivin, 2002).....	29
5.1 Preference_URI.....	52
5.2 Enumerated_Preference.....	54
5.3 Numeric_Preference.....	55
5.4 Interval_Preference.....	56
5.5 Fuzzy_Preference.....	58
5.6 Boolean_Preference.....	59
5.7 UnInterpreted_Preference.....	60
5.8 Context_Based_Preference.....	62
6.1 Otel Ontolojisi.....	70
6.2 Otel ontolojisi Örnek Değerleri.....	70
6.3 Preference_URI.....	71
6.4 Preference_URI Değerleri.....	72
6.5 Numeric_Preference.....	72
6.6 Boolean_Preference.....	73
6.7 Interval_Preference.....	74
6.8 Fuzzy_Preference.....	74
6.9 Enumerated_Preference.....	75
7.1 Ontolojik Örnekler.....	87

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
7.2 Preference_URI Örnekleri.....	89
7.3 Enumerated_Preference Örnekleri.....	90
7.4 Numeric_Preference Örnekleri.....	91
7.5 Interval_Preference Örnekleri.....	92
7.6 Fuzzy_Preference Örnekleri.....	93
7.7 Boolean_Preference Örnekleri.....	94
7.8 Tercih Bağı Örnekleri.....	95
7.9 OntoQL Dili ile Otel Ontoloji Yaratımı.....	99
7.10 Ontolojik Olguların Örneklenmesi.....	99
7.11 Sayısal tercihlerin Örneklenmesi.....	100
7.12 Aralık Tercihlerinin Örneklenmesi.....	100
7.13 Alter Komutu.....	100
7.14 Update Komutu.....	101

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
DDL	Data Definition Language.
DML	Data Manipulation Language.
FOAF	Friends o a Friends
HTML	Hyper-Text Markup Language.
MOF	Meta-Object Facility.
MDA	Model Driven Architecture.
OBDB	Ontology Based Databases.
ODM	Ontology Definition Language.
OWL	Web Ontology Language.
PLIB	Parts Library.
RDF	Resource Description Framework.
RDFS	RDF Schema.
URI	Unified Resource Identifier.
W3C	World Wide Web Consortium.
UML	Unified Modeling Language.
URL	Uniform Resource Locators.

SÖZLÜK DİZİNİ

<u>Kavram</u>	<u>Açıklama</u>
Identifier	Tanımlayıcı
Instance	Örnek
Interval	Aralık
Link	Bağlantı
Nominal	Adlandırmalı
OTV	Ontoloji Temelli Veritabanı
Qualitative	Nitel
Quantitative	Nicel
Ratio	Oransal
Resource	Kaynak

1. GİRİŞ

Web ortamında bilgi üretilmekte, üretilen bilgiye erişim gerçekleştirilmektedir. Web dağıtık bir ortamdır, dinamiktir, çok büyüktür ve herkese açıktır. Anlamsal olarak ifadeleri, ilişkileri ve bilginin kendisini tanımlayan üst verileri içermemektedir. Bu nedenle Web ortamında sunulan bilginin yönetilmesi sorun oluşturmaktadır. Bu sorunun aşılması için bilgi yönetim sistemleri kullanılmaktadır. Veritabanı Yönetim Sistemleri en yaygın kullanılan bilgi yönetim sistemleridir, verinin saklanması, yeniden kullanılması ve sorgulanmasına hizmet ederler. Sahip oldukları şema yapısı gereği kapalı sistemler olarak tariflenmektedirler. Veritabanı Yönetim Sistemleri mantıksal seviyede genişletilebilir, tanımlayıcıları gereği yeniden kullanılabilir değildirler. Kendi içinde verinin anlamı hakkında bilgiyi saklayamadıkları için uygulama geliştirimleri sınırlıdır. Oysa anlam bilimsel veri modelleri ile kavramsal veri modelleri oluşturulmakta, şema yapılarının geliştirilmesi sağlanmakta, veritabanı uygulamalarındaki kısıtlamalar ortadan kaldırılmaktadır. Bu sayede Anlamsal Web araştırma alanına geçiş sağlanmaktadır.

Anlamsal Web, ontolojileri kullanarak kavramları ve kavramlar arasında ki ilişkileri tanımlayan yeni nesil web olarak tariflenmektedir. Standard Web yapısına formal anlam bilgisini, üst veri, üst-şema ve üst üst-şema yapılarını getirir. Kullandığı alt yapı ile Veritabanı Yönetim Sistemlerine tezat; açık, merkezi olmayan ve dağıtık çözümler sunmaktadır. Böylece veriye ve veriler arasında kurulan ilişkilere anlamsal boyutta erişim sağlanabilmekte, URI ler ile temsil edilen ve birbirleri ile ilişkilendirilmemiş bilgiden yeni bilgilere ulaşılabilir. Bu alanda geliştirilen akıllı uygulamalara kişisellik yaklaşımını ekleyerek, kişisel içerik yönetimi, kişisel bilgi yönetimi konularında çözümler sunmaktadır.

Tez çalışması kapsamında anlamsal bir yaklaşım içinde incelenen kişisellik bakış açısı, kullanıcı tercihlerini alan bilgisinden bağımsız olarak ele almaktadır. Oluşturulan kavramsal model ile tercihler Tercih Yönetim Çerçevesi üzerinde tanımlamakta, Veritabanı Yönetim Sistemleri için belirlenmiş anlamsal yapı eksikliğini gidermeye yönelik olarak ortaya konan Ontoloji Tabanlı Veritabanları (OTV) üzerinde bir çözüm önerisi sunabilme hedeflenmektedir. OTV ler Anlamsal Web alt yapısına sahip mimarilerdir. Ontolojileri içererek,

Anlamsal Web in getirdiđi yenilikçi çözüm önerilerini kullanmaktadırlar. Farklı tip mimariler ile açıklanırlar. Bu mimarilerden biri olan OntoDB (Dehainsala et. all, 2007) tez çalışması boyunca kullanılan mimaridir. OntoQL tanımlama ve sorgulama diline sahiptir (Jean et. all., 2006) (Jean et., all 2007), üzerinde Veritabanı Yönetim Sistemi ve ontoloji tanımlama üst-şemasını (ODM) (ODM) (OMG ODM, 2008) temsil eden yapıyı barındırmaktadır. Bu yapı kullanılarak mimari üzerinde yeni kavramsal modeller tanımlanabilmekte, ilgili modellerin mimariye entegre edilmesi sağlanmaktadır.

Tez çalışması kapsamında kavramsal bir model olarak geliştirilen Tercih üst-şemasının OntoDB üzerinde oluşturulması ve var olan üst-şema yapısının genişletilmesi hedeflenmektedir. Tercih temelli sorgulama gerçekleştirimi için ise yeni bir sorgu operatörü tanımlanmasına yönelik çalışılmakta, bu operatörün Tercih üst-şeması elemanları için yapılandırılması sağlanmaktadır. Son olarak geline nokta kişisel tercihlerin sorgulanmasına yönelik bir uygulama gerçekleştirimi sunulmakta ve çalışmanın çıktıları değerlendirilmektedir.

1.1 Konunun Önemi

Tercihler genel olarak karar verme süreci içinde kullanıcıların istek ve gereksinimlerini belirtmeye yarayan kısıtlar olarak tariflenmekte, kişiselleştirme bakış açısı içinde değerlendirilmektedir. Alan bilgisi üzerinde yer alan örnek değerlerin kişisel bilgi olarak ifade edilebilmesi için kişisel sorgular oluşturulmalıdır. Sorgulardan dönecek cevapların kişisel tercihleri temsil edebilmesi için ise olası tercih tiplerinin belirlenmesi, ilgili tercih tanımlamalarının oluşturulması ve alan bilgisini temsil eden üst-şema modelinin incelenmesi gerekmektedir. Bu nedenle kavramsal modellerin yapılandırılmasında kullanılan üst-şema oluşturma yaklaşımı ve üst veri kullanımının önemi tez çalışması kapsamında öne çıkartılmaktadır.

Alan yazın çalışmasında da gösterildiđi gibi henüz literatürde bu bakış açısını kullanan, kullanıcı tercihlerini bir üst-şema ile sunan benzer bir araştırmaya rastlanılmamaktadır. Ancak farklı araştırmalarda farklı tercih tiplerinin tanımlandıđı gözlenmiş, belirlenen tercih tiplerinin ortak bir model altında sunulması gerektiđi fikri desteklenmiştir.

Veritabanı arařtırmaları iin tercihler sorgu kısıtları olarak tariflenmektedir. Bu kısıtlar zerinde sorgu performansının arttırılması ynnde mantık katmanı seviyesinde zmler aranmaktadır. Ancak ilgili arařtırma alanı iinde kabul edilmiř genel bir yaklařıma rastlanılmamaktadır. İncelenen arařtırmalarda, uygulamalarda kullanılan sorgulama tekniklerini iyileřtirici zmler zerine yoęunlukla alıřıldıęı gzlenmektedir. Anlamsal Web alanında ise tercihlerin anlamsal seviyede incelendięi ve řema seviyesinde ontolojik bilginin sorgulanmasına ynelik zm yaklařımları oluřturulduęu belirlenmiř ancak st-řema seviyesinde olmayan zm nerilerine dikkat ekildięi tespit edilmiřtir. Ayrıca, bu alanda kiřisel sorgulamalar iin yeni bir kısıt operatrnn tanımlanması ynnde oluřturulacak bir zm nerisinde bir st-řema oluřturulması fikri kullanılmamıřtır. Bu nedenle oluřturulacak tercih operatr yapısına tercih st-řeması iin bir zm nerisi sunulması gereklilięi belirlenmiřtir.

Gnmz teknolojilerini kullanan uygulamalara gelindięinde ise kiřiselleřtirme ve tercihlerin ynetilmesine ynelik olarak ihtiyaları belirleyen alıřmalara sıklıkla ulařılamamaktadır. Yapılan arařtırmaların bir kısmı tercihlerin kiřisel bilgi olarak saklanabilmesi ynnde FOAF (Brickley and Miller, 2004) dkmanlarına dikkat ekmektedir. FOAF dokmanları iinde kiřisel bilgiler ve arkadaşlık iliřkileri saklanmakta, bu alana zm nerileri sunma ynnde kiřisel bilgi ynetiminin nemi vurgulanmaktadır. Ancak zme ynelik genel bir st modelin oluřturulması ve bu modelin FOAF dokmanları iinde kullanılması ynnde yeni bir fikir sunulmamaktadır. Dięer bir kısım arařtırmada tercihlerin tanımlanacak politikalar ile kısıtlanabileceęi vurgulanmakta, ancak bu yaklařıma uygun bir zm yaklařımı verilmemektedir. Tez alıřması kiřiselleřtirme alanında yapılan arařtırmaları bir araya getirebilmeye ynelik bir zm nerisi sunmayı hedeflemektedir. Bu nedenle oluřturulacak tercih st řeması ve bu řemayı kullanacak yeni bir sorgu operatr, tercih temelli sorgulama geliřtirmede katkı saęlayıcı olacaktır. Bu kapsamda tez, kiřisel bilgi ynetimi konusunda Anlamsal Web uygulamaları iin (Veritabanı Ynetim Sistemlerinin eksik bir yanını giderebilecek řekilde) OTV lerin kullanılabileceęini gstermektedir.

Yapılan alan yazın taramasında OTV ler zerinde kiřiselleřtirme ve tercih temelli sorgulama gerekleřtirmesi konuları alıřılmamıř bir alan olarak belirlenmiřtir. Bu eksiklik tez alıřmasının neminin arttırmaktadır. Bu sayede Veritabanı Ynetim Sistemi ve Anlamsal Web alanı iin ortak bir perspektif

sunma ve eksiklikleri giderme yönünde çözüm geliştirme imkanı elde edilmektedir.

1.2 Çalışmanın Hedefleri

Çalışma kapsamında ele alınan kişiselleştirme bakış açısı yukarıda belirtilen eksiklikleri giderme yönünde kullanıcı tercihlerini tanımlama, alan bilgisi üzerinden bağımsız bir Tercih üst-şeması sunma ve Tercih Yönetim Çerçevesi için kişisel tercihlerin yönetimini gerçekleştirebilme yönünde bir alt yapı oluşturmayı hedeflemektedir.

Tez çalışması boyunca tercihlerin alan bilgisini temsil eden ontolojiler üzerinde yer alan örnek ve öznitelik değerleri ile gösterildiği kabulü yapılmaktadır. Bu kabul doğrultusunda öncelikle ontolojik bilginin, Ontoloji Tanımlama Üst-Şeması (ODM) (OMG ODM, 2008) içinde üst-şema seviyesinde nasıl ifade edildiği incelenmeli, daha sonra ODM de tanımlandığı şekliyle tercihlerin sınıf ve öznitelik varlıkları ile temsil edilebilecek şekilde nasıl modellenmesi gerektiği belirlenmelidir. Tercih üst-şeması oluşturabilme yönünde ise farklı konular, bu alana özgü yapılan çalışmaları bir araya getirebilmek yönünde araştırılmalıdır.

Tercih üst-şeması ile alan bilgisini temsil eden ontoloji üst modelinin entegrasyonun gerçekleştirilebilmesi için ise kullanıcı tercihlerinin Tercih üst-şeması elemanları kullanılarak sorgulanabileceği düşünülmektedir. Bu nedenle tercih temelli sorgulamalarda ihtiyaç duyulan sorgu operatörünün geliştirilmesine yönelik olarak ilgili operatörün yeteneklerinin belirlenmesi gerekmektedir. Kullanıcı tercihlerinin oluşturulan Tercih Yönetim Çerçevesi elemanları kullanılarak sorgulanabilmesi için ihtiyaç duyulacağı öngörülen anlamsal sorgu operatörünün Tercih üst-şeması ile birlikte nasıl kullanılabileceği değerlendirilmelidir.

1.3 Tezin Yapısı

Tez çalışması üç ana kısımdan oluşmaktadır. Kısım I de, çalışılan konunun önemi vurgulanmakta, çalışmanın hedefleri ortaya konmaktadır. Kısım II de, tezde kullanılan temel kavramlar ve çalışmanın oluşturulmasında kullanılan

mimariye ait özellikler gerekli alan yazın taraması ile birlikte sunulmaktadır. Bu kısım için oluşturulan,

Bölüm 2, çalışma boyunca kullanılan tercih, kişiselleştirme gibi temel kavramlar açıklamakta, farklı araştırma alanları için öne çıkan kişiselleştirme yaklaşımlarını, kişisel sorguların oluşturulması konuları ile birlikte incelemektedir.

Bölüm 3, tez çalışması süresince üzerine çalışılan üst-şema oluşturma yaklaşımını model güdümlü mimariler ile birlikte ele almaktadır. Bu sayede var olan bir üst-şemanın genişletilebilmesi konusu bilgi modelleme konusu ile birlikte açıklanmaktadır.

Bölüm 4, tez çalışması için seçilen OntoDB Ontoloji Tabanlı Veritabanı (OTV) mimarilerini birbirleri ile kıyaslayarak açıklamakta, OntoDB mimarisi için tanımlama ve sorgulama dili olarak tariflenen OntoQL dilini tanıtmaktadır.

Kısım III, tez çalışması kapsamında oluşturulan Tercih Yönetim Çerçevesini, oluşturulan çerçevenin OntoDB mimarisine entegrasyonunu ve mimari üzerinde tercih tabanlı sorgulama gerçekleştirimi için sunulan PREFERRING operatörünü açıklamaktadır. Bu kısımda yer alan,

Bölüm 5, EXPRESS modelleme dili kullanılarak yapılandırılan Tercih Yönetim Çerçevesi elemanlarını (Kaynak, Tercih üst-şeması ve Tercih Bağı) tanımları ve söz dizimleri ile birlikte sunmaktadır.

Bölüm 6, Tercih Yönetim Çerçevesinin gerçekleştiriminin anlatıldığı bölümdür. Bu bölümde OntoDB mimarisine Tercih üst-şemasının nasıl entegre edildiği OntoQL dili yetenekleri ile birlikte açıklanmaktadır. Ayrıca tercih tabanlı sorgu gerçekleştirimi için oluşturulan PREFERRING operatörüne ait sözdizimi gösterilmektedir.

Bölüm 7, oluşturulan tercih modelini ve tercih tabanlı sorgulama gerçekleştirimini örnek iki uygulama ile açıklamaktadır. Elde edilen verilerin çözümlenmesi ve ulaşılan bulguların değerlendirilmesi yine bu bölümde yer almaktadır.

Sonuçlar kısmında çalışma boyunca karşılaşılan zorluklar, ileri ki çalışmalara verebilecek katkılar ile birlikte değerlendirilmektedir.

Ekler bölümünde yer alan EK-1 EXPRESS veri modelleme dili yeteneklerini, EK-2 OntoDB mimarisi üzerinde kullanılan OntoQL tanımlama ve sorgulama dilini, EK-3 tercih üst-şemasını, EK-4 UML dili ile çizilen Tercih Yönetim Çerçevesi elemanlarını sunmaktadır.

2 TEMEL KAVRAMLAR VE ALAN YAZIN

Tez çalışması boyunca tercih kavramı kişiselleştirilmiş bilgi, kişisel sorguların oluşturulması ve kişisel tercih yönetimi konuları için değerlendirilmektedir. Bu bölümde farklı yaklaşımlar incelenmekte, bu yaklaşımları kullanan araştırma alanları bir arada sunulmaktadır. Veritabanı Yönetim Sistemleri ve Anlamsal Web araştırmaları başta olmak üzere birçok çalışmada kişiselleştirme yaklaşımları kullanıcılara tercihleri doğrultusunda seçim yapabilme imkanı sağlayabilmeyi hedeflemektedir. Bu nedenle sorgulama dilleri önem kazanmakta, tercih temelli sorgulama gerçekleştirimine yönelik olarak yeni kısıt tarifleri ve yeni sorgu operatörleri oluşturulmaktadır.

Aşağıda öncelikle çalışmaya yön veren temel kavramlar incelenmekte, daha sonra bu kavramları kullanan çalışma örnekleri sunulmaktadır. Tez çalışmasında ortaya konan çözüm önerisine yönelik olarak incelenen tüm çalışmalar için ortak bir değerlendirme ise bölüm sonunda yer almaktadır.

2.1 Tercih Kavramı

Tercihler, karar verme sürecinde belirsiz durumların çözülmesi için kullanılan kavramlardır. Bir seçim uzayı içinde bir varlığı, bir özelliği ya da bir kısıtı diğerine göre daha iyi ya da üstün sayma hali olarak açıklanmaktadır (Boutilier et al. 2004). Kullanıcılar için istekleri ve gereksinimleri tarifleyen kısıtlardır (Hasson, 2001). Farklı disiplinlerce araştırılan bu konu özellikle ekonomistler, matematikçiler ve bilgisayar bilimleri konusunda çalışan bilim adamları tarafından incelenmektedir. Bu nedenle ekonomi, karar süreçleri, sosyal seçim teorisi ve yöneylem araştırmaları gibi geleneksel alanlar ile günümüz uygulamalarında öne çıkan yapay zeka, çoklu ajan sistemleri, kısıt memnuniyeti, karar teorisi gibi hesaplamalı alanlarda, veritabanları, Anlamsal Web, veri madenciliği, insan-bilgisayar etkileşimi, arama motorları gibi bilgi temsilli araştırmalarda önem kazanmaktadır.

Tercihler fayda kuramı (Fisbrn, 1970) içinde ikili ilişkiler olarak ele alınmakta, *psikoloji bilimi* için istek, görev ve değerlere yönelik olarak sebep gösterme hali (Hansson 2001) ya da karar verme sürecinin etkilenmesi durumu olarak açıklanmaktadır. *Matematiksel karar teoremi* içinde kişilerin ekonomik

davranışlarını göstermekte kullanılan bir model (Fishburn, 1988), bilgi temsilli araştırmalarda kullanıcı sorgularına yönelik veri yığınınından cevap olarak dönecek veriyi belirlemede kullanılan kısıtlar (Lacroix, 1987) olarak tariflenmektedir.

Tercihler genel olarak, *nicel* ve *nitel* olmak üzere iki farklı yaklaşım ile açıklanmaktadır. *Nicel* yaklaşımlar tercihlerin dolaylı yoldan ortak bir işlev puanına uygun olarak ifade edilebildiklerini göstermek için kullanılmaktadır. Bu sayede kullanıcılar sorgu cevaplarını sayısal olarak ifade edebilmekte, skor fonksiyonlarına uygun olarak tercihlerini belirtebilmektedirler (örneğin, en iyi 3 yıldızlı oteli arayan bir kişinin bu otel için 5 yıldız skor derecesini kullanması). *Nitel* yaklaşımlar ise klasik karar verme kuramından yola çıkarak, tercihlerin ikililer olarak ifade edilebildiğini göstermektedir. Bu durum da tercihler birbirleri ile kıyaslanabilmekte, bir tercihin diğerine göre üstünlüğü belirtilebilmektedir (örneğin, 4 yıldızlı bir otelin 5 yıldızlı bir otele tercih edilmesi durumu).

Belirtilen yaklaşımlara göre sınıflandırılan tercihler, farklı tercih tipleri ile genel tercih kategorileri altında tanımlanmaktadır. Bu kategoriler *sembolik*, *sıralamalı* ve *oran ölçekli* tercihler olarak gruplandırılmaktadır. Sembolik tercih kategorisi ile metinsel bir ifadeye dayandırılabilen tercihler (örneğin; renk tercihi), sıralamalı tercih kategorisi ile birim içermeyen oluşturulan tercihler (örneğin; büyük, orta, küçük), oran ölçekli tercih kategorisi içinde ise oranı ve sayısal bir etki aralığı ile temsil edilebilen tercihler (örneğin; yakın 50mt, uzak 1 km) belirtilmektedir. Tercih kategorileri içinde yer alabilecek tercih tipleri ise farklı çalışmalarda farklı tanımlamalar ile verilmektedir. Yapılan araştırmalarda kullanıcıların sıklıkla sembolik ve aralık/oran kategorilerini kullandıkları tespit edilmiştir. Bu durum benzer tercihlerin kullanıcılar tarafından farklı söz dizimleri ve farklı sıralama ifadeleri ile kullanıldığını göstermektedir. Bu nedenle benzer söz dizimleri için farklı değer aralıkları ile tariflenen tercihlerin anlamsal olarak incelenmesi gerektiği belirlenmiştir. Tespit edilen bu durum tez çalışması boyunca göz ardı edilmemekte, Bölüm 5 de sunulan Tercih üst-şeması içinde değerlendirilmektedir.

2.2 Kişiselleştirme

Kişiselleştirme, kullanılan uygulamaların ya da araçların kişisel tercihler doğrultusunda kişiye özgü duruma getirilmesi olarak açıklanmaktadır. Bu nedenle kişisel gereksinimlerin bulunmasına yönelik olarak yapılan aramalardan kişiye

özel cevapların getirilmesi olarak da tariflenmektedir. Kişiselleştirme, bilişim teknolojilerinde önem kazanan konular arasında yer almakta ve kullanılan sistem ile kullanıcıların etkileşimini azaltmayı, basitleştirilmeyi hedeflemektedir (Fischer, 2001).

Kişiselleştirme araştırmaları *dışsal* ya da *içsel* olmak üzere iki farklı yaklaşımı içermektedir. *Dışsal* yaklaşımlarda kullanıcıların tercihlerini ifade edebilecekleri ara yüzler kullanılmakta ve bu ara yüzler üzerinde yer alan seçim alanları yardımı ile kişisel seçimlerin yapılması ve bu seçimlerin kayıtlanması sağlanmaktadır. *İçsel* yaklaşımlarda ise kullanıcıların hareketleri ve sistem üzerinde tutulan kayıtları incelenmekte, tercih madenciliği teknikleri kullanılarak yeni bilgilere ulaşması hedeflenmektedir.

Tez çalışması kapsamında tercihlerin sorgulanmasına yönelik olarak sorguların kişiselleştirilmesi konusu araştırılmaktadır. Bu nedenle Veritabanı Yönetim Sistemleri ve Anlamsal Web araştırmaları için kullanılan sorgulama dilleri (SQL ve SPARQL) incelenmekte, tercih tabanlı sorgulama gerçekleştirilmesine yönelik olarak sunulan çözüm önerileri değerlendirilmektedir. Bu doğrultuda Bölüm 6 da ortaya konan çözüm yaklaşımı Ontoloji Tabanlı Veritabanlarında kullanılan OntoQL dili için oluşturulmaktadır.

2.3 Veritabanı Yönetim Sistemleri ve Kişiselleştirme

Veritabanı Yönetim Sistemleri için kişisel sorguların oluşturulması ve kullanıcı tercihlerinin sorgulanmasına yönelik yoğun olarak çalışılmaktadır (Kießling, 2000 -2002), (Chomicki, 2003), (Agrawal, 2000), (Koutrika, 2004), (Viappiani, 2006). Bu nedenle bu bölümde, öncelikle Veritabanı Yönetim Sistemleri için tercih temelli yaklaşımlar ele alınmakta, daha sonra kişisel tercihlerin sorgulaması konusuna yönelik çözümler incelenmektedir.

2.3.1 Tercih Temelli Yaklaşımlar ve İlişkisel Veri Sorgulama

Veritabanı Yönetim Sistemleri tablolardan oluşmaktadır. Bir tablo içinde yer alan satırlar tablonun kayıtlarıdır. Sütunlar tablonun dikey alanlarıdır ve her sütunun mutlaka bir ismi vardır, işlemler bu isimlere göre yürütülür. Veritabanı uygulamaları mantık modeli seviyesinde yapılmaktadır. Modele uygun tutulan

verilerin sorgulanmasında hedef, kullanıcı tercihlerine uygun eşleşmelerin bulunmasıdır. Bu nedenle Veritabanı Yönetim Sistemleri için uygun eşleşme *nitel* ve *nicel* yaklaşımlar kullanılarak yapılmaktadır. Aşağıda öncelikle her iki yaklaşım ayrı ayrı ele alınarak açıklanmakta, daha sonra tercihlerin sorgulanması konusu incelenmektedir.

2.3.1.1 Nitel Yaklaşımlar

Veritabanı Yönetim Sistemleri üzerinde oluşturulan sorgulara yönelik kullanılan *nitel yaklaşımlar*, veriler arasında kıyaslama ifadeleri kullanılarak sonuçların bir sıra düzeni içinde sunulabildiğini göstermektedir. (örnek; otel tercihi için kullanıcının havuzu saunaya, saunayı spor salonuna tercih etmesi gibi). Birçok araştırmacı tarafından (Lacroix and Lavency 1987; Kießling and Güntzer 1994; Köstler et al. 1995; Börzsönyi et al. 2001; Govindarajan et al. 2001; Chomicki 2002; Kießling 2002; Kießling and Hafenrichter 2002; Kießling and Köstler 2002) incelenen *nitel* yaklaşımlar tez çalışması kapsamında Chomicki (Chomicki, et. all., 2003) ve Kießling (Kießling, et. all., 2000, 2002) in çalışmaları doğrultusunda ele alınmaktadır. Her iki araştırmacı tercih temelli sorgu gerçekleştirmeye yönelik olarak farklı tercih tipleri tanımlarını vermekte ve bu tanımları sorgulama yaklaşımları içinde kullanmaktadır.

Chomicki ve ekibi araştırmalarında kullanıcı tercihlerinin ikililer şeklinde tutulduğunu ve ele alınan değerlerden birinin diğerine göre daha üstün olarak ifade edilebileceğini göstermektedirler. Bu nedenle en çok tercih edilen kaydın bulunmasına yönelik olarak tanımladıkları *winnow* tercih operatörünün kullanımını açıklamaktadırlar. *Winnow* operatörü, Börzsönyi (Börzsönyi et al. 2001) tarafından geliştirilen *Skyline* sorgularına benzer bir yaklaşım sunmaktadır. *Skyline* sorguları, SQL sorgularının tercih sorgulaması gerçekleştirimi yönünde genişletilmesi için oluşturulmaktadır. Aşağıda gösterildiği gibi SQL diline ait söz dizimine yeni kısıt operatörlerinin dahil edilmesi sonucu birbiri ile ilişkili olarak belirtilen kayıtlar arasından tercih edilecek kaydın neye göre tercih edildiğini göstermektedir.

```

SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ...
SKYLINE OF A1 [MIN | MAX | DIFF]
...
An [MIN | MAX | DIFF]

```

Bu sayede tercih edilecek özniteliklere ilişkin kayıtlar arasında en düşük (MIN) ya da en yüksek (MAX) değerlerin bulunması için karmaşık sorgulamalar gerçekleştirilebilmektedir. Örneğin; farklı turizm acenteleri tarafından fiyatlandırılan otellere ait en ucuz fiyatlamasının bulunması için yapılacak bir sorgu, aşağıda ki gibi oluşturulmaktadır.

```
SKYLINE OF Hotel_name DIFF, PRICE MIN.
```

Chomickinin çalışmasından sunulan çözüm önerisine benzer bir öneri bu çalışmadan bağımsız olarak Kießling (Kießling, 2002) in çalışmasında da yer almaktadır. Kießling *nitel tercih yaklaşımını* Preference SQL dilini oluştururken kullanmaktadır. Preference SQL içinde kullanılan tercih tipleri *basit tercihler* ve *karmaşık tercihler* olmak üzere iki grupta ele alınmaktadır. Basit tercihlerin *pozitif* ve *negatif* tercihlerden oluştuğu ve atomik durumlar olarak tariflenen bu tercihlerin sorgulanabilmesinde *Around(k)*, *Between(k and l)*, *Lowest(k)*, *Highest(k)* gibi fonksiyonların kullanıldığı belirtilmektedir. *Karmaşık* tercihlerin ise basit tercihlerin AND operatörü kullanılarak bir araya getirilmesinden oluştuğu açıklanmakta, sorgulanmaları için ise *Pareto* (Kießling, et. all., 2002) ve *Priority* (Kießling, et. all., 2002) tercih sorgulayıcılarına ihtiyaç duyulduğu belirtilmektedir.

Pareto tercih sorgulayıcı, birden çok özeliğin eşit ağırlıklı olarak tercih edildiği durumlarda (örneğin; bir otel tercihi yapılırken eşit önemde olduğu düşünülen havuzu olsun, saunası olsun, kapalı otoparkı olsun, sineması olsun vb.), *Priority* tercih sorgulayıcı ise çoklu tercihlerin bir araya getirilmesi durumunda önceliklikli tercihlerin belirtilmesinde kullanılmaktadır (örneğin; bir otel tercihi yaparken öncelikle havuz olsun, sonra kapalı otoparkı ve sineması olsun vb.).

Preference SQL (Kießling, 2002) (Kießling, 2003) *Skyline* sorgularında olduğu gibi SQL sorgularının tercih sorgulaması gerçekleştirimi yönünde genişletilmesi ile sağlanmaktadır. Aşağıda Preference SQL diline ait söz dizimi bu söz dizimine eklenen yeni kısıt cümlecikleri ile birlikte sunulmaktadır.

```

SELECT < selection>
FROM < table_references>
WHERE <hard-conditions>
PREFERRING <soft-conditions>
GROUPING <attribute_list>
BUT ONLY <but_only_condition>
ORDER BY <attribute_list>

```

Belirtilen söz diziminde yer alan Where yapısı içinde <hard-conditions> olarak temsil edilen tercih edilmesi zorunlu kısıt değerlerinin bulunmasına yönelik bir kesinlik ifadesi kullanıldığı belirtilmektedir. PREFERRING yapısı içinde belirtilecek olan kullanıcı tercihleri <soft conditions> olarak temsil edilmekte ve belirtilen tercihlerin bir esneklik ifadesi ile birlikte kullanıldıkları gösterilmektedir (örn: yaklaşık 14 günlük otel rezervasyon tercihi). Tercihleri gruplayabilmek adına GROUP BY cümlecği, bir tercihe yönelik yeni bir zorunluluğu belirtmek için BUT ONLY cümlecği ve dönecek cevapların belirli bir sıralamaya göre alınmasını sağlamak amacıyla ORDER BY cümlecği kullanılmaktadır. Son olarak Preference SQL ile sorgulamadan dönen değerlerin istenen tercihlere en uygun cevaplar olup olmadığının belirtilebilmesi için ise özel fonksiyonlara başvurulmaktadır. Bu fonksiyonlar arasında yukarda bahsedilen *Pareto*, *Priority* ve *Top(k)* fonksiyonları bulunmaktadır (Koutrika et. al., 2004). *Top(k)* fonksiyonu ile dönen cevaplar arasından kullanıcının tercihi doğrultusunda belirlenen en iyi k sonucun listelenebilmesi hedeflenmektedir.

2.3.1.2 Nicel Yaklaşımlar

Nicel yaklaşımlarda tercih edilen durumlara yönelik olarak sayısal ifadeler kullanılmaktadır. Aralık veya oransal tarifler ile temsil edilen sayısal skor fonksiyonları tercihlere yönelik elde edilen sorgu sonuçlarının sıralı olarak alınmasını sağlamaktadır. Bu yaklaşım üzerine çalışan araştırmacılar (Agrawal and Wimmers 2000; Hristidis et al.2001; Koutrika and Ionnis, 2004) skor fonksiyonlarının dışsal olarak tanımlanması gerektiğini vurgulamaktadır. Tercih Yönetim Çerçevesi içinde kullanılan bu yaklaşıma yönelik olarak Agrawal

ve Koutrika'nın çalışmaları üzerinde bir inceleme yapılmakta ve bu yaklaşım için belirlenen tercih tipleri incelenmektedir.

Agrawal ve ekibi çalışmalarında veri tiplerini (string, float, boolean, ..) incelemekte ve skor adını verdikleri yeni bir sayısal tercih tipi tanımlamaktadırlar. Tanımladıkları yeni veri tipinin 0-1 aralığında değer aldığını açıklamakta, bu sayede en çok tercih edilen özelliğe (1) ve en az tercih edilen özelliğe (0) skor değerinin atandığını göstermektedirler. Farklı tercihlerin bir araya getirilebilmesinden oluşan *karma tercihler* için ise yeni bir değer fonksiyonu tanımına ihtiyaç duyulduğunu belirtilmektedirler. Ancak çalışmada tercihlerin sorgulanabilmesine yönelik bir çözüm önerisi sunmamaktadırlar.

Koutrika ve ekibinin yaptığı araştırmada ise öncelikle *atomik* ve *karmaşık* tercihlerin tanımları yapılmaktadır. *Atomik* tercihler ile kısıt değerlerine yönelik olarak SQL diline ait atomik sorgu elemanlarının kullanımı tarif edilmekte, *karmaşık* tercih yapısı ile ise atomik tercihlerin bir araya getirilmesinde bir derecelendirme fonksiyonunun kullanılması gerektiği belirtilmektedir. Bu nedenle $\langle \text{condition}, \text{score} \rangle$ ikililerinin kullanımı ile *karmaşık tercih* yapısı gösterilmektedir. Oluşturdıkları tercih sorgularında ise derecelendirme fonksiyonu kullanımını, sorgu sonucu dönen cevapların sıralandırılmasında *nitel* yaklaşımlarda oldu gibi *Top(k)* algoritmasının kullanımı ile açıklamaktadırlar.

Nitel yaklaşımlar için incelenen çalışmalarda gözlenen durum, *nicel* yaklaşımlarda olduğu gibi bir sorgu operatörü tanımı yapılmadığı, SQL dili yeteneklerini artırıcı katkı sağlayıcı bir eklentinin bulunmadığı yönündedir.

2.3.2 Değerlendirme

Veritabanı Yönetim Sistemleri için incelenen çalışmalara ilişkin elde edilen bulguların değerlendirilmesi Çizelge 2.1 de verilmektedir. Belirtilen çizelge tercihlerin hangi kişiselleştirme yaklaşımları ile tariflendiğini model seviyesinde göstermekte, tanımlanan tercih tiplerinin genel bir tercih modeli altında sunulup sunulmadığını belirtmekte, örneklenen çalışmaların SQL dili için tanımlanan yeni bir sorgulama operatörü kullanımı ile desteklenip desteklenmediğini göstermektedir. Belirtilen çizelge üzerinde elde edilen bulgular kişisel tercihlerin iki farklı yaklaşım ile ele alındığı yönündedir. Her iki yaklaşım içinde tercihler,

tercih edilen ve tercih edilmeyen durumları açıklamak için olumlu ya da olumsuz tercihler olarak tarif edilebilmekte, tek tek ya da bir arada ele alınmaktadır. Kişisel sorguların gerçekleştirilmesine yönelik olarak ise sadece *nitel* yaklaşımlar için bir çözüm önerisinin SQL dili üzerinde oluşturulduğuur. Belirtilen çözüm önerisi ile tercihler tek tek ya da birlikte sorgulanabilmektedir.

Çizelge 2.1 Veritabanı Araştırma Alanı ve Tercih Yaklaşımları.

Yazar	Alan veritabanı	Yaklaşım	Model Seviyesi	Sorgu Oluşturma	Tercih Modeli
Kiebling 2002-2003	DB	Nitel	Mantıksal	PREFERRING	Hayır
Chomicki 2003	DB	Nitel	Mantıksal	Winnnow	Hayır
Agrawal-Wimmers 2000	DB	Nicel	Mantıksal	Hayır	Hayır
Koutrica-Ionnidis 2006	DB	Nicel	Mantıksal	Hayır	Hayır

Ancak *nicel* yaklaşımlar için ortaya konan sayısal tercih tiplerine yönelik olarak yeni bir sorgu operatörü tanımı oluşturulmamaktadır. Bu yaklaşım ile sadece belirtilen tercih tiplerine yönelik standart SQL sorguları gerçekleştirilmektedir. Fakat bu durumda da sorgu sonucunda hiçbir eşleşmemenin yapılamadığı ve boş cevap kümesinin kullanıcıya döndürüldüğü durumlar ile karşılaşılmalıdır. Tespit edilen bu olumsuzluğun Veritabanı Yönetim Sistemlerine özgü olarak olduğu belirlenmiş, mantıksal model seviyesinde esnek olmayan bir şema yapısı içinde tercih temelli sorgulama gerçekleştirilmesine yönelik çözüm önerilerinin yetersiz kaldığı tespit edilmiştir. Bu olumsuz durumun ancak anlamsal model seviyesinde oluşturulacak çözümler ile aşılabileceği düşünülmektedir.

Aşağıda Anlamsal Web araştırma alanı için kişiselleştirme çalışmaları kişisel sorguların oluşturulması yönünde incelenmektedir. Benzer bir çizelge ile elde edilen bulguların değerlendirilmesi ilgili bölümün sonunda ayrıca sunulmaktadır.

2.4 Anlamsal Web ve Kişiselleştirme

“World Wide Web Konsorsyumu (W3C)” tarafından, web üzerinde verinin anlamsal gösterimini sağlamak amacıyla “Anlamsal Web (Semantic Web)” kavramı Şubat 2001 tarihinde, ortaya atılmıştır. World Wide Web’in kurucusu Tim Berners Lee, Anlamsal Web’in şimdiki web teknolojisinin bir uzantısı olduğunu, bilginin anlamının tanımlandığını, böylelikle bilgisayar ve kullanıcıların daha iyi bir şekilde birlikte çalışabilirliklerinin sağlandığını ifade etmiştir.

Anlamsal Web’in çalışabilmesi için, bilgisayarlar yapısal bilgiye ve çıkarsama kurallarına ihtiyaç duymaktadırlar. Bu durum, ontoloji kullanımı ile sağlanmaktadır (Doerr, 2003). Ontoloji içinde varlıklar ve ilişkiler birlikte tanımlanmaktadır. Sözlük anlamı olarak “varlık bilimi” demektir. Belli bir alandaki bilgilerin paylaşımını ve yeniden kullanımını sağlayan “kavramlaştırmaların biçimsel ve açık tanımlanması” olarak tanımlanmaktadır (Gruber, 1993). Kavramsallaştırma, belli bir tasarım aşamasında soyut model oluşturma anlamına gelmektedir. Bilgisayar bilimleri içerisinde ontoloji kelimesi, varlıkları, varlıklar arasındaki ilişkileri ve varlıkları içeren tasarımların nasıl oluşturulacağını tanımlamak amacıyla kullanılmaktadır. Ontolojiler ile hiyerarşiler oluşturulmakta, bilginin temsiline bir standartlaşma getirilmektedir. Ortak bir dilde, üst verisine uygun olarak tanımlanan kavramlar, tanım açısından aynı temel türden gelmektedir (Gruber, 1994). Bu temelin açık ve paylaşılabılır olması, verinin temsilde karşılaşılan uyumsuzlukları ortadan kaldırmaktadır. Bu sayede ortak üst veri kullanılarak tanımlanan kavramlar, tüm farklılıklarına rağmen hiyerarşik olarak aynı küme ya da benzer kümelerde yer alırlar. Bu yakınlık ilişkisi, farklı ontolojiler içerisinde tanımlanan kavramların insanlar ve bilgisayarlar tarafından aynı şekilde anlaşılabilmesini sağlamaktadır.

Ontolojiler, ontoloji dilleri ile tanımlanmaktadır. W3C tarafından 2002 yılında geliştirilen OWL (Web Ontoloji Dili) (Mike, 2004) dili yaygın olarak kullanılan ontoloji geliştirme dilidir. Anlamsal Web’i oluşturan önemli öğeler arasında RDF (Kaynak Tanımlama Çerçevesi) (Hayes, 2003) ve RDF şema gelmektedir. RDF bilgisayarlarca işlenecek verinin anlamını temsil edecek veri modelinin düzenlenmesini sağlamakta, RDFS (RDF şema) RDF veri modelini genişleterek, alanda kullanılacak sözcük kümesini nesneler, nesneler arası

ilişkiler, özellikler ve özelliklerin alabileceği değerler açısından tanımlamaktadır (Brickley, 2000).

Web Ontolojisi, web üzerindeki bir alanda (özel bir konuya ait bilgi alanı), paylaşılabılır bilgiye ulaşmak isteyen ihtiyaç sahiplerine nesnelerin kurallı tanımını yaparak ortak kelimeler ve anlamlar sunmaktadır. Bir ontoloji pratik olarak (Davies, 2002):

- kavram veya varlık sınıflarının tanımlanması,
- sınıfların hiyerarşilerinin düzenlenmesi,
- varlıklara ait örnekler ve özellikler arasındaki ilişkilerin tanımlanması ile geliştirilmektedir.

Tez çalışması boyunca kişiselleştirme bakış açısı içinde ontolojilerin kullanılması konusu önemli bir yere sahiptir. Tercih Yönetim Çerçevesi bu doğrultuda oluşturulmakta ve kişisel tercihler ontolojik bilgi ile temsil edilmektedir. Bu nedenle ortaya konan çözüm yaklaşımı Anlamsal Web araştırma alanını desteklemektedir.

Bir sonra ki kısımda ontolojik bilginin sorgulanmasında kullanılan kişiselleştirme yaklaşımları incelenmektedir.

2.4.1 Ontoloji Temelli Kullanıcı Tercihleri ve Ontolojik Veri Sorgulama

Tıpkı Veritabanı Yönetim Sistemlerinde olduğu gibi Anlamsal Web alanında da kişiselleştirme çalışmaları önem kazanmaktadır. Bu amaçla farklı çalışmalarda, farklı tercih tipleri tanımlanmakta, kişisel sorguların gerçekleştirilmesinde tercih tiplerinin kullanılması gerektiği vurgulanmaktadır. İncelenen çalışmalarda tercih temelli yaklaşımlar, tercih temelli ontolojik veri sorgulama konusu ile birlikte SPARQL sorgulama dili için değerlendirilmektedir. SPARQL veri kaynakları için bir protokol ve sorgulama dilidir. RDF dili ile tanımlanmış üçlülere sorgulamaktadır. Tıpkı Veritabanı Yönetim Sistemlerinde kullanılan SQL sorgulama dilinde olduğu gibi SPARQL dili için de tercih temelli sorgulama yaklaşımı araştırılmaktadır. SQL sorgulama diline benzer tercih operatörü tanımlanması yönünde Siberski'nin çalışması Anlamsal Web

araştırmalarına katkı sağlamaktadır (Siberski, 2006). Bu çalışmada SPARQL in uzantısı olarak yeni bir dil tanımlanmamakta ancak standard SPARQL dilin yeni bir sorgulama dili yapı taşı ile nasıl genişletildiği açıklanmaktadır. Bu doğrultuda öncelikle SPARQL dili yapısal özellikleri aşağıda gösterilmektedir.

```

      SolutionModifier ::=      OrderClause?      LimitClause?
      OffsetClause?

      OrderClause ::= 'ORDER' 'BY' OrderCondition+

      OrderCondition ::=      ( ( 'ASC' | 'DESC' ) '(' Expression ')' )
                              | ( FunctionCall | Var | '(' Expression
                              ')' )

      LimitClause ::=      'LIMIT' INTEGER
      OffsetClause ::=      'OFFSET' INTEGER

```

Yukarıda verilen söz diziminde kullanılan *SolutionModifier* tanımı ile sonuç düzenleyiciler tanıtılmakta, özellikle *Order* ve *Limit* kısıt tanımlayıcıları kullanılarak sorgulardan dönen cevapların belirtilen belirli bir sıra ya da sınır değerine uygun olarak düzenlenebildiği açıklanmaktadır.

```

      SolutionModifier ::=      PreferringClause? OrderClause?
                              LimitClause? OffsetClause?

      PreferringClause ::=      'PREFERRING' MultidimensionalPreference

      MultidimensionalPreference ::=
      CascadedPreference
      ('AND' CascadedPreference)*
      CascadedPreference ::= AtomicPreference
      ('CASCADE' AtomicPreference)*

      AtomicPreference ::=      BooleanPreference
                              | HighestPreference | LowestPreference

      BooleanPreference ::=      Expression
      HighestPreference ::=      'HIGHEST' Expression
      LowestPreference ::=      'LOWEST' Expression

```

Tercihlerin sorgulanmasına yönelik olarak her hangi bir tercih operatörü yapısının SPARQL dili içinde tanımlanmamış olması Siberski'nin çalışmasında yer almaktadır. Bu nedenle yukarıda yer alan söz dizimsel sonuç düzenleyicilere kullanıcı tercihleri sorguları oluşturabilmek için yeni bir *sıralı sonuç düzenleyici*

tanımının eklenmesi gerektiği ilgili çalışmada belirtilmekte ve `<soft constraint>` olarak tanımlanan kullanıcı tercihlerinin sorgulanabilmesine yönelik `PREFERRING` operatörü tanımı SPARQL sorgulama dilinin söz dizimine dahil edilmektedir.

Yukarıda belirtilen söz dizimi sonuç düzenleyicilerine yönelik yapılan tanımlamayı genişletmektedir. Kullanıcı tercihlerinin *atomik tercih* tanımı ile gösterilen *boolean tercihlerden* ya da atomik tercihlerin bir araya getirilmesi ile *çok boyutlu tercihlerden* oluştuğu belirtilmekte, oluşan *çok boyutlu tercih* yapısı için `AND` ve `CASCADE` tanımlarının kullanıldığını gösterilmektedir. Bu yapıyı kullanan tercih temelli SPARQL sorgularında ise aşağıda ki gibi bir sorgu cümlesi yapısının kullanması gerektiğini vurgulanmaktadır.

```
SELECT <projection-var-list>
FROM <ontology-reference>
WHERE <var-bindings>
FILTER <hard-conditions>
ORDER BY <var-list>
PREFERRING <soft constraints>
```

Belirtilen söz dizimi içinde yer alan `PREFERRING` cümlecisi tıpkı SQL dilinde ki tanımı gibi kişisel tercihleri içerecek şekilde `<soft constraints>` olarak tariflenmektedir. Sorgudan dönen cevapların sıralanabilmesi için ise `ORDER BY` cümlecisinin kullanıldığı gösterilmektedir.

2.4.2 Değerlendirme

Kullanıcı tercihlerinin tanımlanmasına yönelik olarak yapılan bir kısım Anlamsal Web araştırmaları kişisel sorguların oluşturulması yönünde değil, tercihlerin nasıl tariflenebileceğinin gösterilmesi yönündedir. Örneğin Dubois araştırmasında *nitel tercihleri*, *durum temelli* ve *ihtimal temelli* olmak üzere ikiye ayırmaktadır (Dubois, 2004). İhtimal temelli tercihleri bir olasılık değeri ile birlikte ifade etmekte, durum temelli tercihleri ise bir durumun oluşması koşuluna bağlı olarak açıklamaktadır. Başka bir araştırmada Fischer ve grubu, tercihleri öznitelik tipine bağlı olarak *sayısal tercihler* ve *sayısal olmayan tercihler* olarak ikiye ayırmaktadır (Fischer et al., 2006). Toninelli ve grubu ise anlamsal bir model oluşturma yönünde tercihlerin tanımlanması gerektiğini vurgulamakta, bu

nedenle tercihlerin ontolojiler üzerinde nasıl temsil edildiğini açıklamaktadırlar (Toninelli et. al., 2008). Çalışmalarında ayrıca *sınıflar* arasında tanımlanan ve ilişkileri gösteren *nesne* özellik değerinin de bir tercih ögesi olduğu belirtilmektedir. Bu öge üzerinden kullanıcı tercihlerinin tanımlanabileceği fikri çalışmanın sonunda verilmekte, anlamsal bir modele duyulan ihtiyaç ortaya koyularak, bu ihtiyacı karşılayan bir yapının oluşturması gerektiği vurgulanmaktadır. Ancak araştırmalarına bu ihtiyaca cevap veren bir model sunma yönünde devam etmemektedirler.

Anlamsal Web araştırma alanı için kullanıcı tercihlerinin bulunması yönünde yapılan çalışmaların genel bir değerlendirilmesi Çizelge 2.2 de sunulmaktadır.

Çizelge 2.2 Anlamsal Web Araştırma Alanı ve Tercih Yaklaşımları.

Yazar	Alan (Anlamsal Web)	Yaklaşım	Model Seviyesi	Sorgu Operatörü	Tercih Modeli
Siberski et al. 2006	AW	Nitel	Anlamsal	PREFERRING	Hayır
Dubois, 2004	AW	Nitel	Anlamsal	Hayır	Hayır
Fischer et al. 2006	AW	Nicel	Anlamsal	Hayır	Hayır
Tonielli et al. 2008	AW	Üst-Model	Anlamsal	Hayır	Hayır

Çizelge içinde yer alan yaklaşımların yine *nitel* ve *nicel* yaklaşımlardan oluştuğu dikkat çekicidir. Tercihler Anlamsal Web araştırma alanı içinde anlamsal seviyede ele alınmaktadır. Bu alanda genel bir modele ulaşabilme yönünde bir üst model oluşturma yaklaşımı sadece bir çalışma içinde değerlendirilmektedir. Kişisel tercihlerin sorgulanabilmesi için ise SQL dilinde olduğu gibi SPARQL diline yeni eklenen bir operatörün tanımı gösterilmiş ancak bu operatörün kullanımı ve iyileştirilmesi için diğer çalışmalar ile desteklenmediği tespit edilmiştir. Bölüm 2.3.2 de yer alan Çizelge 2.1 de sunulan verilerde olduğu gibi yine bu alanda belirlenen kriterler içinde en dikkat çekici yan farklı tercih tiplerini bir araya getiren genel bir modelin literatürde bulunmayışıdır.

Toninelli'nin araştırmasından elde edilen bulgular tez çalışması için katkı sağlayıcı niteliktedir. Üst-şema oluşturma yönünde tercihlerin ve alan

ontolojilerinin incelenmesi gerektiği fikrinin Tercih Yönetim Çerçevesi içinde yer alması gerektiği, bu çalışmanın incelenmesi sonrasında belirlenmiş, bu yönde oluşturulması gereken çözüm yaklaşımının ontolojik varlıkların sorgulanmasında ihtiyaç duyulan tercih temelli sorgulama gerçekleştirimi konusu içinde de yer alması gerektiği tespit edilmiştir.

Ancak unutulmamalıdır ki Veritabanı ve Anlamsal Web araştırmaları kullanıcı tercihleri ve kişiselleştirme yaklaşımlarına yönelik yapılan çalışmaların sadece bir kısmını kapsamaktadır. Aşağıda tez çalışmasına yön veren diğer araştırma alanlarına kısaca değinilmekte, seçilen önemli çalışmalar örneklendirilmektedir.

2.5 Diğer Araştırma Alanları ve Kişiselleştirme Yaklaşımları

Bu bölümde öncelikle kişiselleştirme araştırmalarının örneklendiği Web Madenciliği, Veri Ambarları, İnsan Bilgisayar Etkileşim Sistemleri araştırmalarına yönelik kısa bir değerlendirme sunulmakta, daha sonra arama motorları, öneri sistemleri ve kullanıcı profilleri gibi uygulama gerçekleştirmeleri için kişiselleştirme yaklaşımları incelenmektedir.

2.5.1 Web Madenciliği

Web Madenciliği, Web dokümanlarından yeni bilgi elde edebilmek üzere *Web yapı madenciliği*, *Web içerik madenciliği* ve *Web kullanım madenciliği*, tekniklerinin kullanılması ile gerçekleştirilmektedir. *Web yapı madenciliği*; Web sitesi ve Web sayfalarının birbirleri ile olan bağlantılarına bakılarak yeni bilgi üretilmesinde kullanılmaktadır. Bu sayede Web dokümanları arasında bağ yapısı incelenmekte, Web sayfaları hakkında yeni bilgilere ulaşılmaktadır. *Web içerik madenciliği*; Web dokümanlarından tercih edilen bilgiye ulaşabilmek için kullanılmakta, *Web kullanım madenciliği* ise veri madenciliği teknikleri kullanılarak, Web kullanıcılarının İnternetteki davranışları için örüntüler oluşturmak olarak tariflenmektedir.

Bu alanda kişiselleştirme yaklaşımları Web kullanım madenciliği ile birlikte değerlendirilmektedir. Web üzerinde doğrudan erişilebilen veriyi kullanmak yerine, kullanıcıların Web'de dolaşırken hareketlerinden oluşturulan verilerden bilgi üreterek kişisel çözüm önerileri oluşturma hedeflemektedir

(Mobasher 2000). Bu alanda en çok kullanılan yöntemler, Markov modelleri (Sarukkai, 2000), (Deshpande vd., 2001), sıralı örüntüler (Pitkow vd., 1999), bağıntı modelleri (Mobasher vd. 2001) ve demetleme (Mobasher vd., 2000) yöntemleridir. Bu çalışmaların çoğunda kullanıcı davranışları ele alınmaktadır.

2.5.2 Veri Ambarları

Veri ambarları ilişkili verilerin sorgulanabilmesi için gerekli analizlerin yapıldığı depolar olarak tariflenmektedir. Bu alanda yapılan çalışmalar veritabanları ile yakından ilişkilidir ve sorgulamaların olabildiğince otomatik olarak yapılabilmesi için çözüm önerileri sunmaktadır. Veri ambarında verinin analizi OLAP (Online Analytical Processing) programları ile yapılmaktadır. Bu programlar veriye çok boyutlu bir küp olarak bakmayı sağlamaktadır (Han, 2006). Böylece boyut bazında yapılan guruplamalar, boyutlar arasındaki korelasyonlar ile incelenmektedir. Oluşturulan OLAP sorguları için kişiselleştirme çatısı sunma yönünde araştırmalar literatürde yer almaktadır. Bu araştırmalar kullanıcıların kişisel tercihlerini OLAP sorgu sonuçları üzerinden ele almalarını sağlamaktadır (Bellatreche, 2005). Ancak bu tür sorgulara yönelik olarak herhangi bir tercih operatörünün geliştirimi literatürde yer almamaktadır.

2.5.3 İnsan Bilgisayar Etkileşim Sistemleri

1960 larda başlayan bu araştırma alanı 1980 lerde kullanıcı modelleri geliştirme yönünde devam etmiştir. Öncelikle bilgi sistemleri için kullanıcı modellerinin kullanılması ve kullanıcı davranışlarına yönelik kayıtların tutulmasına çalışılmıştır. Günümüz araştırmalarında ise insan bilgisayar etkileşim sistemleri için bağlam temelli çözüm önerileri, kullanıcı modelleri araştırmaları ile birlikte sunulmakta, tercih temelli yaklaşımları oluşturulmaktadır (Suh, 2007).

Literatürde yer alan birçok farklı insan bilgisayar etkileşim sistemleri uygulamalarında kişiselleştirme yaklaşımları ve tercih temelli uygulamalar, arama motorları, kullanıcı profilleri ve kullanıcılara öneri sunan sistemleri ile bir arada örneklenmektedir. Bu nedenle aşağıda bu alanları kapsayan araştırmalar için kısa bilgilere yer verilmektedir.

2.5.4 Arama Motorları

Genel anlamda, Web tüm dünyaya dağılmış ve birbirine bağlanmış, bir bilgi havuzudur. Web'de bulunan bilgiler arama motorları kullanıcıları için çoğu zaman sorun yaratmaktadır. Ancak “örümcek”, “robot”, “ajan” sistemler ile arama motorlarına yönelik çözümler sunulmakta, Web sayfalarına etkin erişimlerin gerçekleştirilebilmesi yönünde araştırmalar yapılmaktadır (Dupret, 2007). Bu nedenle arama motorları iki grupta ele alınmaktadır:

- *Tekli Arama Motorları* arama yapmak için kendi veritabanlarını kullanmaktadırlar. Belirli adreslerden başlayarak tüm web sayfalarını kendi veritabanına indirerek bu sayfalarda geçen kelimeleri indeksleme yoluna giderler. Bu arada son indirme zamanı, son güncellenme zamanı, özet kelimeler, sayfaların başlıkları gibi bilgileri de saklarlar. Ancak kişisel istekler doğrultusunda oluşturulmuş içeriğe sahip sayfalara ulaşılabilmesi yönünde çözüm önerilerini içermezler.
- *Üst Seviyeli Arama Motorları* kendi veritabanını oluşturmak yerine tekli arama motorlarının veritabanlarını kullanarak arama yaparlar. Böylece arama yapılan konu hakkında birden fazla arama motorunu tarayarak arama sonuçlarını gösterirler (Deng, 2004). Bir ya da birden çok arama motorunun ortak arama yaptığı üst arama sayfaları için ortak amaç kullanıcılara istekleri, tercihleri doğrultusunda uygun sorgu cevaplarını döndürebilmektir. Bu nedenle farklı sorgulama kriterleri için derecelendirme sistemleri, kullanıcıların ortak yanlarına uygun cevap döndürmeyi hedefleyen profil yaklaşımları bu tür arama motorları için kişiselleştirme yaklaşımlarını sunma yönünde çözümleri içermektedir. Her ne kadar birçok yeni arama motoru geliştirilmiş olsa da Web'in ivmelenerek artan büyümesi tüm Web sayfalarını toplayıp indekslemeyi ve bu indeksleri de güncel tutmayı imkânsızlaştırmaktadır. Birçok arama motoru genelde eksik ve güncel olmayan sonuçlar döndürerek kullanıcıların isteklerine yanıt verememektedir. Bu yüzden son zamanlarda yapılan çalışmalar kişisellik kavramına odaklanmış arama motorları üzerine yoğunlaşmaktadır.

2.5.5 Öneri Sistemleri

Kişiselleştirme tekniklerinin uygulanmasında öne çıkan sistemlerden bir diğeri de öneri sistemleridir (Agarwal, 2005). Genel olarak bu sistemler kullanıcılara geçmiş tercihleri doğrultusunda otomatik öneri yapmayı hedeflemektedir. Birçok elektronik ticaret sitesi bu sistemleri müşterilerine kişisel tercihlerine uygun ürünleri sunabilme yönünde kullanmaktadır.

Öneri sistemleri için, Web kullanım madenciliği teknikleri kullanılmakta, Web kullanıcılarının davranışları incelenerek uygun örüntülerin bulunması sağlanmaktadır. Doğru ve hızlı öneri yapabilmek için ise bulunan örüntülerin kullanıcı davranışları ile etkin biçimde modellenmesi ve hızlı öneri kümesi üretilmesi gerekmektedir. Bu nedenle öneri sistemleri farklı yaklaşımları içermektedir. Örneğin işbirlikçi öneri sistemleri ilgili sistemlerinin en yaygın bilinen tipidir (Adomavicius, 2005). Patern yaklaşımlarını kullanarak kullanıcıların tercihlerini kayıtlamalarını, benzer tercihler doğrultusunda yeni önerilerde bulunmayı hedeflemektedir Shardanand'ın (Shardanand, 1995) çalışmasında belirtilen süreç,

- kullanıcı bilgilerinin toplanması,
- bir tercih profilinin oluşturulması,
- düzenli güncellemelerin yapılması,
- hedef kullanıcıya öneri sistemleri ile kişisel tercihleri doğrultusunda önerilerinin iletilmesi olarak açıklanmaktadır.

2.5.6 Kullanıcı Profilleri

Profil, benzer özelliklere sahip kullanıcıların aynı isim ile referans edilebilmesini sağlayan bir yapıdır. Profil oluşturma, kullanıcılara ait bilgileri bir profil tanımı altında toplama sürecidir. Profil sayesinde, kullanıcı özellikleri ile bu özelliklerin alabileceği değerler kümesini ifade eden bir yapı sunulmaktadır. Profiller özellik ve değer ikililerini içermektedir. Literatürde yer alan profil yaklaşımları *içsel* - *dışsal* ve *durağan* - *devingen* olarak çeşitlenmektedir (Jrad, 2007).

İçsel profil yaklaşımı ile kullanıcıların Internet üzerindeki davranışlarına yönelik kayıtlarının tutulması, *dışsal profil yaklaşımı* ile ise, kullanıcıların

web sayfaları üzerinde yapacakları seçimler ile oluşturdıkları profilleri anlatılmaktadır. Bu nedenle *içsel profiller* sistem üzerinde tutulan geçmişe yönelik kullanıcı hareketlerinin bir araya getirilmesinde kullanılmaktadır ve kullanıcılar tarafından bu tür profiller değiştirilememektedir.

- *Durağan profil yaklaşımı* kişinin bilgilerinin sistem üzerinde saklanması sonucunda oluşurken, *devingen profil* yaklaşımı dışsal profil yaklaşımını içinde barındırarak zaman içinde geliştirilebilen profiller olarak açıklanmaktadır.

Tercih seçimleri, kullanıcının kendine uygun profili belirlemesi sonrasında, tercih uzayının ilgili profiller doğrultusunda daraltması fikrini öngörmektedir. Bu fikir farklı çalışmalarda ontolojiler kullanılması fikri ile desteklenmektedir (Razmerita, 2003), (Trajkova, 2000).

2.6 İlgili Çalışmalara ait Genel bir Değerlendirme

Bu bölümde incelenen araştırmalardan elde edilen bulgular doğrultusunda tez çalışması için aranan çözüm yaklaşımının Veritabanı Yönetim Sistemleri ve Anlamsal Web araştırmaları için tespit edilen eksikleri giderme yönünde olması gerektiğidir. Bu nedenle çalışmanın Ontoloji Temelli Bilgi Sistemleri üzerinde gerçekleştirilmesine karar verilmiştir. Bu doğrultuda Anlamsal Web araştırmaları ile paralellik gösteren ancak mimarisi gereği içinde veritabanı tablolarını da barındıran Ontoloji Tabanlı Veritabanları (OTV) tez çalışmasına dahil edilmektedir. OTV mimarileri üzerinde bilgi, ontolojik olarak ifade edilebilmekte ve geleneksel veritabanı mimarilerinin sahip olduğu tablo yapısında tutulmaktadır. OTV mimarileri ile Veritabanı Yönetim Sistemlerinin eksik kaldığı noktalara (esnek olmayan tasarım, sorgu sonuçlarının hata döndürdüğü durumlar gibi) anlamsal boyutun eklenmesi ile yeni çözüm önerileri sunulabilmektedir.

Çizelge 2.1 ve 2.2 de gösterildiği gibi alan yazında incelenen ve tez çalışmasına temel teşkil eden araştırmalar Çizelge 2.3 de aynı kriterler kullanılarak bir arada gösterilmektedir. Belirtilen çizelgede gözlenen en önemli durum, ortaya konan kişiselleştirme yaklaşımları için *nicel* ve *nitel* yaklaşımlarda ayrı ayrı yoğun olarak çalışıldığıdır. Kişisel sorguların oluşturulmasında *nitel* yaklaşımlar için sorgulama dillerine katkı sağlayıcı çözümlerin getirildiği fakat iki yaklaşımın bir araya getirilmesi yönünde bir çözüm önerisinin sunulmadığıdır.

Tez çalışması kapsamında *nicel* yaklaşımlara yönelik bir çözüm önerisi sunma fikri *nitel* olarak ifade edilen yaklaşımlar ile birlikte ele alınmakta, iki durumu da kapsayan bir çözüm önerisi geliştirilmesi yönünde çalışılmaktadır.

Ayrıca diğer çalışmalardan farklı olarak tercihlerin tercih tipleri olarak kategorize edilmesi gerektiği belirlenerek, oluşturulan yapının bir tercih modeli sunabilme yönünde üst-şema yapısı içine taşınması gerektiği vurgulanmaktadır.

Çizelge 2.3 Araştırma Alanları ve Tercih Yaklaşımları.

Yazar	Alan (DB/AW/..)	Yaklaşım	Model Seviyesi	Sorgu Operatörü	Tercih Modeli
Kiebling Chomicki Agrawal et al. Koutrica et al. Das et al.	DB	Nitel Nicel	Mantıksal	PREFERRING	Hayır
-	OTV	-	-	-	-
Siberski et al. Sieg et al. Gurský et al. Tonielli et al.	AW	Nitel Nicel Üst-model	Anlamsal	PREFERRING	Hayır

Bu doğrultuda kişisel tercihlerin kavramsal olarak modellenebilmesi ve tercih temelli sorgulama gerçekleştirimi ile sorgulanabilmesi yönünde Tercih Yönetim Çerçevesi oluşturulmasının katkı sağlayıcı bir çözüm önerisi olabileceği düşünülmektedir. Böylece Tercih Yönetim Çerçevesinin OTV üzerinde ki gerçekleştirimi ile anlamsal seviyede bir çözümü sunulabilecek ve anlamsal veri sorgulama dillerinden bir olan OntoQL dili için tercih temelli sorgulama gerçekleştirimine yönelik yeni bir çözüm yaklaşımı getirilebilecektir.

Bu bölümde çalışmanın dayanağını oluşturan konular ile ilgili temel kavramlar incelenmiştir. Bir sonraki bölümde Tercih Yönetim Çerçevesi için önemli olduğu düşünülen bilgi modelleme yaklaşımları ve bu yaklaşımları için kullanılan veri modelleme dillerine yönelik alan yazın çalışmaları sunulmaktadır.

3 BİLGİ MODELLERİ

Model, gerçeğin basitleştirilmiş halidir. Yazılım mühendisliği bağlamında ise gerçek dünyanın bir bölümünü temsil eder, bir modelleme diliyle ifade edilir ve temsil ettiği gerçek dünyanın ilgili bölümü hakkında belirli bir amaç için bilgi verir. Bir modelde temsil edilen gerçek dünya bölümüne ise sistem, nesne sistemi, hitap edilen uzay, hedef sistem, incelenen sistem gibi isimler verilebilmektedir (Atkinson and Kühne, 2003). Bu tanımda önemli olan modelin temsil ettiği gerçek dünya bölümünün sınırlarının belirlenmiş olmasıdır. Modeller, gerçek dünyanın bir bölümünün soyutlanması sonucu ortaya çıkar, doğal dil ile biçimsel ya da biçimsel olmayan şekilde ifade edilebilmektedir. Soyutlama sırasında modellenen gerçek dünya bölümünün bazı özellikleri modele yansıtılmakta, bazı özellikleri ise ihmal edilmektedir. Hangi özelliklerin modele yansıtılacağı, hangi özelliklerin ihmal edileceği modelin oluşturulma amacı tarafından belirlenmektedir.

Bilgi Modelleme, veri modellemede karşılaşılan sorunları çözmek ve veriyi en etkin şekilde ifade etmek için geliştirilen yaklaşımdır. Bilgi modelleri değişik veri/bilgi ve süreçlerin standart bir yapıda biçimlendirilmesi sonucunda oluşturulmaktadır. Veri modelleri (ilişkisel, nesnel, ağ, hiyerarşik, vb) kendisi ile tanımlanan verinin doğrudan bilgi işleme sürecine girebilmesini hedeflemektedir. Bilgi modelleri kavramsal modellerdir, ilişkisel ve nesnel veri modellerinin bir üst seviyesinde yer almaktadır (Dori, 2004). Verinin tekil yapı içerisinde ifade edilmesine olanak veren bu modeller gerektiğinde nesnel veya ilişkisel modellere dönüştürülebilmektedir. Modellere yazılım geliştirme sürecinin odağına koyan yaklaşıma Model Güdümlü Mühendislik adı verilmektedir.

3.1 Model Güdümlü Mimari

Model Güdümlü Mühendislik yaklaşımının en çok bilinen iki gerçekleştirimi Object Management Group (OMG) tarafından 2001 yılında duyurulan “Model Güdümlü Mimari” dir (Bezivin et al., 2004; Bezivin and Gerard, 2002). Model Güdümlü Mimari, modellerin yazılım geliştirme sürecinde kullanılabilmesi için önerilen ve sistemin çalışma belirtimlerini, sistemin kullandığı platformun detaylarından ayırmak amacıyla geliştirilen bir yazılım tasarım yaklaşımıdır. Bu nedenle de,

- sistemi platformdan bağımsız tanımlamak,

- platformları tanımlamak,
- sistem için platform seçmek,
- sitem tanımlamasını belirli bir platforma dönüştürmek için gerekli yaklaşımları içermek,
- hesaplama bağımsız model, platform bağımsız model, ve platform bağımlı modelleri ve bunların arasındaki dönüşümleri tanımlamaktadır.

Model GÜdümlü Mimari ilkeleri, kavramsal modelin temelini oluşturur. Kavramsal Model, tüm uygulamaların ortak gereksinimlerini ortaya koymaktadır (Dori, 2004). Model üzerinde kullanılması gereken varlıkları içermekte, varlıkların sahip oldukları karakteristikleri ve birbirleriyle olan ilişkilerini tanımlanmaktadır.

Literatürde farklı Model GÜdümlü Mimari yaklaşımları ve farklı standartlar bulunmaktadır. Her Model GÜdümlü Mimari yaklaşımı ve standardı, kendisine uyumlu araçlarla hayata geçirilir. Model GÜdümlü Mühendislik yaklaşımının amaçları, yazılım kalitesini arttırmak, yeniden kullanımı teşvik etmek, yazılım geliştirme verimliliğini yükseltmek, yazılım geliştirme maliyetlerini azaltmak, yazılım geliştirme sürecindeki karmaşıklıkları mümkün olduğu kadar ortadan kaldırmak olarak özetlenebilmektedir. Model GÜdümlü Mimari modelleri, yazılım geliştirme sürecinin merkezinde yer almaktadır. Bu süreçlerinin soyutlama seviyesini yükseltmeyi hedeflemektedir. Temel özellikleri arasında (Selic, 2003):

- modellerin iyi tanımlanmış gösterimlerle ifade edilmesi ve,
- modeller arasındaki dönüşümler için, bu modelleri biçimsel olarak tanımlayan üst-şemaların kullanılması gelmektedir.

OMG'nin Model GÜdümlü Mimarisi kaynak ve hedef platformlardaki modellerin bütünlüğünü kaybetmeden, uygulamalardan ayrılmasını sağlamaktadır (Dustdar and Schreiner, 2005). Bu nedenle de üst-şemaları kullanmaktadır. Üst-şemalar, modeller arasındaki dönüşümleri ve modellerin birbirleriyle bütünlleştirilmesini sağlar. Bir modelin üst-şeması, o model oluşturulurken kullanılabilecek yapı elemanlarını ve bu yapı elemanlarının kullanım kurallarını tanımlar. Bir model, üst-şemasında tanımlı olan kurallara uymak zorundadır. Dolayısıyla üst-şema “modelleme dilinin bir modeli” olarak tanımlanabilir. Her üst-model özel bir dil ile tanımlanır. Sonuçta bütün üst-modeller bir üst-üst-

modele bağılıdır. Model GÜdümlü Mimarilerde bu üst-üst-model *Meta-Object-Facility (MOF)*'dir (OMG MOF, 2002). Aşağıda *MOF* yapısı kısaca açıklanmaktadır.

3.1.1 Meta-Object Facility (MOF)

Meta-Object Facility (MOF) (OMG MOF2, 2003), yazılım modelleyiciler tarafında sıklıkla kullanılan *UML* çekirdeğinin MGM'nin ihtiyaçlarına uygun olarak adapte edilmiş şekli olarak geliştirilmiştir. *MOF* temel olarak diğer modelleme dillerinin tanımlanabilmesi için gerekli olan minimum kavram setidir (Gasevic et al., 2006). Yapının modellenmesinde kullanılan *UML* kısmına benzerdir (ancak birebir aynı değildir). En son versiyonunda (*MOF 2.0*), *MOF* ve *UML* üstyapısı kavramları *UML* kavramlarından çıkartılmıştır. *UML* altyapı (OMG *UML*, 2003) isimli bir *Object Management Group (OMG)* standardı da mevcuttur. Bu standart diğer üst-modellerde kullanılmak üzere geliştirilmiş temel kavramları içermektedir.

MOF'da birçok farklı kavram olmasına karşın, dört ana modelleme kavramı aşağıdaki şekilde sınıflandırılabilir (Gasevic et al., 2006):

- Sınıf (*Class*) – *MOF* üst-nesnelerini, üst-modellerde varlık (*entity*) olarak bulunan kavramları (Ör., *UML* Sınıfı, Özellik ve İlişki, *ODM* Sınıfı ve Özelliklerini ve ayrıca Örnek ve Öznitelik değeri gibi *MOF* kavramlarını) modellemektedir.
- İlişki (*Association*) – İkili ilişkileri (Ör., *UML* veya *MOF* üst-sınıflarını ve tiplerini veya *ODM* üst-özelliklerini) modellenmektedir.
- Veri tipi (*DataType*) – Temel tipleri (String, Integer, vb.) modellemektedir.
- Paket (*Package*) – Diğer kavramları modüler hale getirmektedir (Ör., benzer kavramları gruplamak).

Model GÜdümlü Mimari yazılım mimarileri için farklı model seviyelerini MDA seviyeleri ile açıklamaktadır. Aşağıda Model GÜdümlü Mimari içinde tanımlanan farklı model seviyeleri incelenmektedir.

3.1.2 MDA Seviyeleri

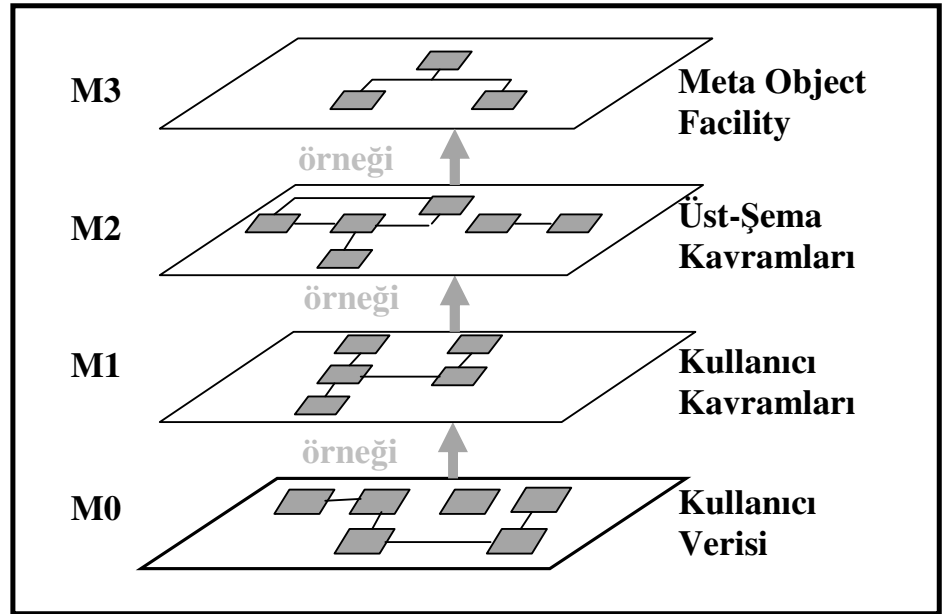
OMG tarafından MOF standartlarında (OMG MOF, 2002), (OMG MOF2, 2003) 4 seviyeli olarak tanımlanan üst-şema yapısı Çizelge 3.1 de özetlenmektedir (Bezivin, 2002). Belirtilen çizelge üzerinde yer alan seviyeler bir üst numaralı seviyenin *örneği* olarak düşünülmelidir.

Çizelge 3.1 MDA Seviyeleri (Bezivin, 2002).

Kısa Ad	Seviye	Açıklama
M0	Model Örneği	Uygulama örneğini içerir, örneğin bir sınıfın koşturma zamanındaki örneği veya veritabanındaki bir satırdır. (ör. Nesneler)
M1	Model	Uygulamayı içerir, örneğin nesne yönelimli bir sistemdeki sınıflar veya ilişkisel veritabanındaki tablo tanımlamalarıdır. (ör. Kullanıcı Modeli)
M2	Üst-şema	Modelleme dilini içeren üst-şema, örneğin UML’de tanımlı sınıf, özellik, operasyon gibi model elemanlarının tanımlandığı seviyedir. (ör. Entity, Attribute)
M3	Üst üst-şema	Tüm üst-şemaların sahip olabileceği davranış ve özellikleri tanımlayan seviyedir. (ör. MOF)

Çizelge 3.1 de gösterildiği gibi bir üst-şema, bir modelleme dilinde geçerli olan modelleri tanımlamaktadır. Üst-şema, modelin üstünde yer alır ve modeller bu üst-şemanın “örnekleri” (“instance of”) olur (Djuric, 2005). Ancak model ve üst-şema arasındaki “örnekleme” (“instance of”) ilişkisi sadece üst-şema dilinin teorisi için geçerlidir. Üst-şema dili elemanları ile modelleme dili elemanları arasında geçerli değildir. Bu nedenle üst-şema ve model arasındaki örnekleme ilişkisini nesneye yönelik yaklaşımdaki sınıf ve örneği arasındaki ilişki gibi düşünmemek gerekir. Tanımdan da anlaşılacağı üzere üst-şema da bir modeldir. Ancak temel iki farklılığa sahiptir.

Öncelikle üst-şema tanımladığı dilin söz dizimsel ve anlam bilimsel kuralları gibi temel özelliklerini içermelidir. Bir modelin söz dizimsel yapısının nasıl olacağı o modelin üst-şeması ile belirlenmektedir. İkinci olarak üst-şema bir üst-şema mimarisinin parçası olmalıdır. Bu nedenle de üst-şema bir model olarak ele alınabilir. Örneğin bir modelleme dili olan UML (OMG UML, 2003), yine OMG tarafından geliştirilmiş olan MOF üst-şema mimarisi içinde yer almaktadır. Şekil 3.1’de model güdümlü mimari teknolojilerinde yer alan 4 katmanlı altyapı görülmektedir.



Şekil 3.1 Geleneksel Nesne Yönetim Grubu (OMG) Modelleme Altyapısı (Atkinson and Kühne, 2003).

Bu dört katmanlı mimari sayesinde birçok yeni modelleme standardı da (Ör., Ortak Depo Üst-modeli (*Common Warehouse Metamodel (CWM)*) (OMG CWM, 2003) MOF örnekleri olarak M2 katmanında geliştirilmiştir. MOF tabanlı araçlar bu yüzden bu tip yeni modelleme standartlarını destekleyebilmekte ve aralarında bilgi değiş tokuşu yapabilmelerine olanak vermektedir.

3.2 Model Tabanlı Yazılım Geliştirme

Model tabanlı yazılım geliştirme, yazılım mühendisliğinde nesneye yönelik programlamadan daha üst düzeyde soyutlama olanağı sunan bir yaklaşımdır. Model tabanlı yazılım geliştirmenin ilk noktası, çalışma alanıdır. Bu kelime, ilgili alana yönelik tanımlanan kavram ve ilişkiler hiyerarşisi olarak açıklanmakta, çalışma alanının sınırlarını belirlemekte, ilgili alana ait terminolojiyi tanımlamakta ve alanın söz konusu alt alanlara bölünerek karmaşık bir problemin, basit anlaşılabilir sistemler haline getirilmesini sağlamaktadır. Model tabanlı yazılım geliştirme konusu farklı modelleme dilleri ile desteklenmektedir. Aşağıda tez çalışması boyunca kullanılacak olan modelleme dillerine yönelik açıklamalar yer almaktadır.

3.2.1 Modelleme Dilleri

Modeller sistem karmaşıklığını yönetilebilir bir boyuta indirerek anlaşılabilirliği arttırmaktadır. Bu nedenle bir model üzerinde tasarıma yönelik temel özelliklerin açıklanması gerekmektedir. Belirtilen gereksinim modellemede ortak bir gösterim biçiminin, ortak bir modelleme dilinin kullanılması ile giderilebilmektedir. Yazılım alanında bu nedenle geliştirilmiş çeşitli modelleme dilleri bulunmaktadır. Tez çalışması içinde kullanılan UML ve EXPRESS modelleme dillerinin genel olarak açıklanması aşağıda sunulmaktadır.

3.2.1.1 UML ve UML Profilleri

UML bir programlama (ya da yazılım geliştirme) dili olmaktan çok iş sistemlerinin nasıl modellenebileceğini belirleyen ve açıklayan yöntemlerin bir araya toplanmış halidir. Yazılım mühendisliğinde nesne tabanlı sistemleri modellemede kullanılan açık standart olmuş bir görsel modelleme dilidir. Nesne tabanlı yazılım geliştirme ve yazılım geliştirme sürecinin önemli bir kısmını oluşturmaktadır. UML, yazılım projelerinin tasarımlarının ifade edilmesi için genellikle grafiksel notasyonlar kullanmaktadır (Braun ,2001).

Kurumsal bilgi sistemlerinden, dağıtık ağ-tabanlı uygulamalara ve gerçek zamanlı gömülü sistemlere kadar birçok sistemi modellemek için uygun bir

dildir. Sahip olduđu UML diyagramları ile bir sistem modeli 3 farklı açıda ele alınmaktadır. Bunlar modelin,

- **İşlevsel gereksinimler açısından,** kullanıcının bakış açısını ve sistemin gereksinimlerini vurgulamakta kullanılmaktadır. Kullanım Senaryosu (Use-Case) diyagramını içerir.
- **Statik yapısal açısından,** nesneler, nesnelere ait özellikler ve ilişkiler kullanarak sistemin statik yapısı incelenmektedir. Sınıf (Class) ve Birleşik Yapı (Composite Structure) diyagramlarını içerir.
- **Dinamik davranış açısından,** nesneler arası ortak çalışmalar ve nesnelerin durumlarındaki değişiklikleri gösteren sistemin dinamik davranışlarını incelemektedir. Sıralama (Sequence), Faaliyet (Activity) ve Durum (Statechart) diyagramlarını içerir.

Kavramsal model dediğimiz Sınıf diyagramları, gereksinimlerin oluşturulmasında ve tasarımıda kullanılmaktadır. Sınıf diyagramlarının tasarımı ilgilendiren kısmı oldukça kapsamlıdır, bu sebeple tasarlanacak diyagramlarında, türetme, soyut sınıflar, sınıf hiyerarşileri, kullanılacak tasarım şablonları gibi detayların düşünülmesi gerekmektedir (Falkovych, 2003).

UML profilleri mevcut UML dil yeteneklerini özelleştirmek ve çalışma alanına göre basit adımlarla gerçekleştirmek için kullanılmaktadır. Özelleştirme yöntemi, mevcut nesnel tasarımlardaki genişletme tekniğine benzetilebilir. Bu nedenle UML profilleri, mevcut UML ifadeleri ve kuralları kullanarak ilgili kavramlara değişik anlam ve özellikler ekleme ya da çıkarmayı sağlamakta, model seviyesinde, genişleme mekanizmalarını ve eşlemeleri desteklenmektedir. Tez çalışması içinde oluşturulan Tercih üst-şeması ile UML profillerinin sağladığı yeteneklere paralel olarak bir ontoloji modelini nasıl genişletebileceği gösterilmektedir.

Bir sonra ki bölümde tez çalışması sırasında kullanılan modelleme dilleri açıklanmaktadır.

3.2.1.2 EXPRESS Veri Modelleme Dili

Bilgi yapıları ISO10303 tarafından tanımlanan EXPRESS bilgi modelleme dili ile ifade edilmektedir. Bu dil ile ifade edilen yapılar bir üst-şema olarak değerlendirilmektedir. ISO10303, uluslararası standardizasyon kuruluşu (ISO) na ait TC184/SC4 teknik komitesinin bilgi modeli yaklaşımını kavramsal yapıdan fiziksel (uygulanabilir) yapıya getirmek amaçlı geliştirdiği standartlar serisi olarak anılmaktadır. ISO10303 standartlar serisi bilgi modellerinin tanımlanacağı dilleri (EXPRESS, EXPRESS-G, EXPRESS-X), bilgi modeli ile tanımlanmış verinin ifade edilme şeklini, bilgi modellerine erişmek için standart bir ara yüzü içermektedir.

EXPRESS dili formal bir modelleme dili olarak tanımlanmaktadır. Belirtilen dil alan bilgisine ait kavramları bir *kavramsal model*, ve belirtilen kavramlar için gerekli veri yapısını *mantık modeli* altında sunmaktadır. EXPRESS dili, soyutlama ve nesnelerin sınıflandırılması için *varlık* kavramını kullanmaktadır. Varlık nesneye dayalı programlama dilinde kullanılan *sınıf* kavramına karşılık gelmektedir. Ancak varlıklar sınıf kavramında olduğu gibi metotlar ile birlikte tanımlanmazlar. Fakat varlıklar arasında, sınıflar arasında olduğu gibi bir miras ilişkisi tanımlı yapılabilir. Belirtilen ilişki çoklu, tekli ya da yinelemeli olarak çeşitlenmektedir. Bir varlık kesinlikle öznitelik içermektedir. Anılan öznitelikler varlığın sahip olduğu olgu dokümanına ilişkin karakteristik özellikleri değerleri ile birlikte göstermektedir. Belirtilen değerler ancak tanımlı olduğu veri tipine uygun olan yapıda ifade edilebilmektedir.

Tez çalışması içinde oluşturulan Tercih Yönetim Çerçevesi elemanları EXPRESS modelleme dili yetenekleri ile oluşturulmuştur. Bu dile ait genel özellikler EK-1 de sunulmaktadır.

3.2.2 Ontoloji Tanımlama Dili (ODM)

Anlamsal Web araştırma alanında, alan bilgisi ontolojiler ile temsil edilmektedir. Ontoloji kavramı, kavramlar arasındaki ilişkileri formal olarak göstermektedir. Kısım 2.4 de tariflendiği gibi ontolojiler, dağıtık verilerin üst verilerine dayalı olarak modellenmiş ortak iletişim aracıdır (Gruber and Olsen, 1994). Sınıflandırmanın bir boyutu olarak da tariflenen ontolojiler bu özelliği ile geliştirilecek yazılıma için tanımlanan problem alanını ya da geliştirme tanımının

alt yapısının modellenmesi için önemlidir. Ontolojilerin modellediği alan, yazılımın gerçek dünyadaki problem alanı ya da yazılım geliştirme sürecindeki çeşitli altyapı alanları (yazılım bileşenleri kütüphanesinin ontolojiler ile modellenmesi gibi) olabilmektedir. Bu iki boyuta göre Happel ve Seedorf (Happel and Seedorf, 2006), ontolojilerin yazılım mühendisliğindeki kullanımları için üç sınıf belirlemiştir:

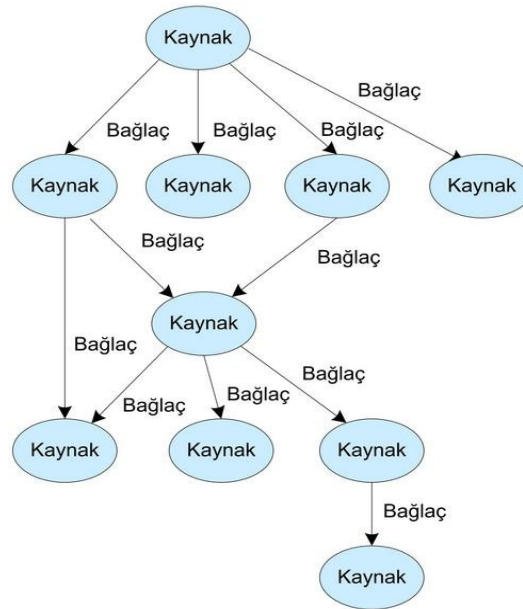
- *Ontoloji güdümlü geliştirme*, ontolojilerin, yazılımın problem alanını tanımladığı ve gerçekleştirim zamanında kullanıldığı sınıfı temsil etmektedir. Bu kullanım sınıfına örnek olarak alan modellerinin ontolojilerle temsil edilmesi, ontoloji modelleme dillerinin diğer modelleme dilleriyle bütünleştirilmesi verilebilir.
- *Ontoloji destekli geliştirme*, ontolojilerin gerçekleştirim zamanında ve yazılım geliştirme sürecinin iyileştirilmesinde kullanımını öngörmektedir. Bu kullanım sınıfına örnek olarak bileşenlerin yeniden kullanımı, kodlama desteği ve test verilebilir. Bu kullanım sınıfında ontolojiler yazılım geliştirme süreci kapsamındaki alanları modellemektedirler.
- *Ontoloji tabanlı mimari*, için ise ontolojiler çalışma zamanında kullanılmakta ve uygulama mantığının temelini oluşturmaktadır. İş mantığının ontoloji destekli yapılarla temsil edilmesi bu uygulama sınıfı için bir örnek teşkil edebilir. Ontoloji destekli mimarilerde ontolojiler, yazılıma çalışma zamanında altyapı desteği verirler.

ODM Ontoloji Yönetim Grubu (*Ontology Management Group (OMG)*) tarafından geliştirilen bir Ontoloji Tanımlama Üst-modeli (*Ontology Definition Metamodel (ODM)*) spesifikasyonudur. (OMG ODM Adopted Specification, 2006). Bir ontolojinin geliştirilebilmesi için gerekli olan tüm kavramları bir üst-sema seviyesinde anlatmaktadır. İlgili tüm ODM kavramlarının modellenmesi için MOF kullanılmaktadır. Yazılım mühendisliği alanında ve MOF'la birlikte, Model Güdümlü Mimari teknolojik uzayındaki standartlar ve bu standartlara göre geliştirilmiş araçlar, ontoloji mühendisliği araçlarının çözüm getirmediği gereksinimleri karşılama, bu gereksinimlere yönelik çözümler üretebilme için kullanılmaktadır.

3.2.2.1 ODM için Kaynak Tanımı

Web temelli uygulamalar için Kaynak terimi ilk kez hedef dokümanlar için URL'ler olarak kullanılmış, süreçte Kaynak Tanımlayıcısı URI'ler olarak genişletilmiştir (RFC 3987 ve RFC 3986). Kaynaklar günümüz web uygulamalarında internet ortamı üzerindeki bir web sitesi, bir web sayfası ya da web sayfasının bir parçası olarak tariflenmektedir.

Anlamsal Web, RDF üzerine inşa edilmiştir. RDF, verilerin sunumunu sağlamak için üst-verileri ifade etmeye yarayan URI'ler kullanmaktadır. RDF tarafından biçimlendirilen ve ilişkilendirilen bu veri kümeleri ontolojiler yolu ile saklanmaktadır. Çünkü ontolojiler birçok ilgi alanındaki sınıfları ve web dokümanlarında kullanılan özellikleri anlamsal olarak tanımlar. Bu nedenle Anlamsal Web uygulamalarında Kaynaklar Şekil 3.2 de gösterildiği diğer kaynaklara bağlaçlar kullanılarak ilişkilendirilmektedir.

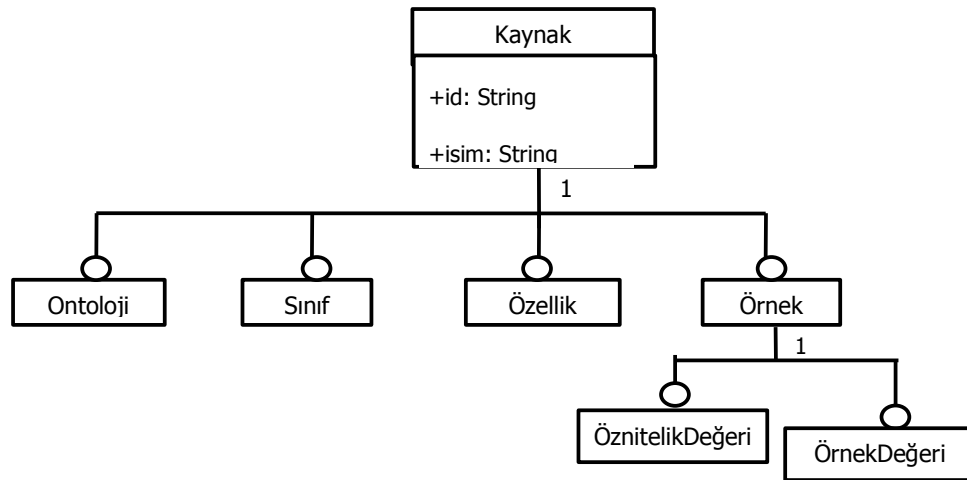


Şekil 3.2 Kaynak Gösterimi.

Web ontolojileri, farklı uygulama sistemleri arasında paylaşılabilen ortak alan bilgisi kullanmaktadır. (Örneğin, *şarap ontolojisi*, *turizm ontolojisi*.) Ontolojiler genellikle OWL ontoloji dili kullanılarak yaratılmaktadırlar (McGuinness et. al., 2004). OWL dili, RDFS'i temel aldığı için OWL üst-şeması, RDFS üst-şemasından genişletilerek oluşturulmuştur. UML, yazılım

mühendisliğinde kabul görmüş ve yoğun olarak kullanılan bir dildir. Bundan dolayı ODM belirtiminin içinde, zaten bir OMG standardı olan UML diliyle de bir bağlantı kurulmuştur. RDFS ve OWL için UML profilleri, UML dilinin ve araçlarının ontoloji modelleme için kullanımını sağlar. ODM'nin içerdiği üst-şemalar arasında eşleştirmeler de tanımlanmıştır (Brockmans, 2004).

MOF ile modellenen en basit ODM kavramları Şekil 3.3 de gösterilmektedir. Ontolojiler, sınıflar, sınıflara ait özellikler, sınıfa ait örnekler ve bu örneklerin kendi aralarındaki ilişkilerini ilgili temel elemanlar ile açıklanmaktadır.



Şekil 3.3 ODM için Kaynak (OMG ODM, 2008).

Şekil üzerinde yer alan Kaynak varlığı, ODM içinde tanımlanan kök sınıf ve RDF için en temel kavramı, *ontoloji*, varlıkların hiyerarşik bir düzen içinde tanımlandığı uzayı, *sınıf*, varlıkların tanımlanmasında kullanılan temel yapıyı, *özellik*, sınıflar için tanımlanan karakteristikleri ve sınıflar arası ilişkileri gösteren bağlantıları, *örnek*, bir sınıfa ait değerler bütünü ve özel türde kaynakların aldığı değerleri *öznitelikDeğeri* ya da *örnekDeğeri* olarak içermektedir.

Yukarıda yer alan kavramları kullanarak oluşturulan Tercih Yönetim Çerçevesi için Kaynak elemanı Bölüm 5 de açıklanmaktadır.

3.3 Değerlendirme

Bu bölümde *kavramsal bir model* olan Tercih üst-şemasının oluşturulması için gerekli alan yazın incelemesi sunulmuştur. Bu nedenle tez çalışmasında model temelli yazılım gerçekleştirme yaklaşımı benimsenmiştir. Model kavramı olguların sistem yaklaşımıyla incelenmesi ve tasarlanmasında kullanılmaktadır. Hiçbir model tanımı, tanımı gereği hedef sistemin tüm özelliklerini içermemektedir. Ancak sistemin o anda incelenen özelliklerini ve sistem davranışlarının en belirgin özelliklerini mutlak olarak içermektedir. Tercih Yönetim Çerçevesi içinde bu nedenle göz ardı edilen durumlar söz konusudur.

Kavramsal modeller için oluşturulan üst-şemalar karmaşık problemlere çözüm bulmak amacıyla kullanılmaktadır. Bu nedenle mimariler üzerinde yer alan farklı modelleme dilleri arasında geçişleri tanımlamak üst-şemaların kullanımı ile kolaylaşmakta ve farklı seviyelerde soyutlama gerektiren problem alanları için farklı dillerin tanımlanmasını sağlayan bir altyapı ile sağlanmaktadır. Bu nedenle Tercih üst-şemasının farklı diller ile tanımlanabileceği unutulmamalıdır. Çalışmada alan yazında belirtilen model temelli yazımlardan biri olan EXPRESS veri modelleme dili kullanılmaktadır. Bu dilin sağladığı avantajlar arasında;

- standartlaşmış uluslararası bir dil ile modelin ifade edilmesi ve bu dili bilen herkes tarafından diyagramların anlaşılması,
- kodlamayı kolaylaştırması,
- hataları en aza indirmesi, çıkan hataları düzeltmenin daha kolay olması,
- tekrar kullanılabilir bileşenlerin artması,
- program kararlılığının artması gelmektedir.

Oluşturulan modelin gerçekleştirimi için ise Ontoloji Temelli Veritabanlarından biri olan OntoDB mimarisi seçilmiştir. Bir sonraki bölümde genel olarak OTV mimarileri için bir alan yazın sunulmakta ve OntoDB mimari açıklanmaktadır.

4 ONTOLOJİ TEMELLİ VERİTABANLARI

Ontoloji temelli bilgi sistemleri Anlamsal Web ve Ontoloji Tabanlı Veritabanları (OTV) (Dehainsala et al., 2007), (Pierra et al., 2005) için bir araştırma konusudur. OTV ler bilgiyi ontolojik olarak ifade edebilmeyi sağlama yönünde geleneksel veritabanı mimarilerine ontolojik yaklaşımların eklenmesi ile oluşturulmuştur. Belirtilen mimariler ontoloji ve ontolojiler ile ifade edilen bilgiyi veritabanlarının sağladığı fonksiyonel özellikleri kullanarak işlemektedir. Bu sayede veri saklama yetenekleri, yüksek sorgulama performansları ve işlem gerçekleştirime hacimleri ile avantajlı duruma gelmektedirler. Günümüzde yaygın kullanılan OTV mimarileri arasında Sesame (Broekstra et al., 2002), RDFSuite (Alexaki et al., 2001), Jena (B.McBride, 2001), (Carroll et al., 2004), OntoDB (Pierra et al., 2005), (Dehainsala et al., 2007b), OntoMS (Park et al., 2007) and KAON (Bozsak et al., 2002) yer almaktadır.

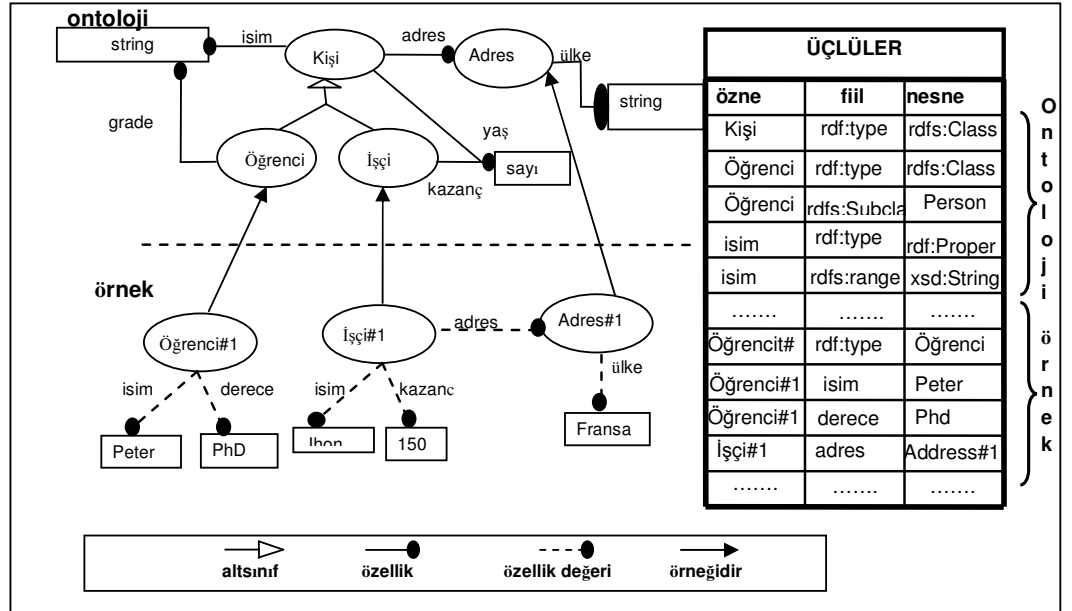
Bu bölümde öncelikle OTV mimarileri açıklanmakta, daha sonra bir OTV mimarisi olan OntoDB mimarisi ve bu mimari üzerinde kullanılan OntoQL dili (Jean et. all., 2007) yetenekleri anlatılmakta, son olarak da tez çalışması için neden OntoDB mimarisinin seçildiği değerlendirilmektedir.

4.1 OTV Mimarileri

OTV mimarileri ile ontolojik verinin ve sistem kaynaklarının etkin bir biçimde yönetilmesi amaçlanmaktadır. Son on yıl içinde farklı araştırmalarda farklı OTV mimarileri ortaya konulmuştur. Bu çeşitlilik OTV mimarileri üzerinde yer alan şema dokümanlarını sayısı ile doğru orantılıdır. Aşağıda 3 grup altında belirtilen OTV mimarileri incelenmektedir.

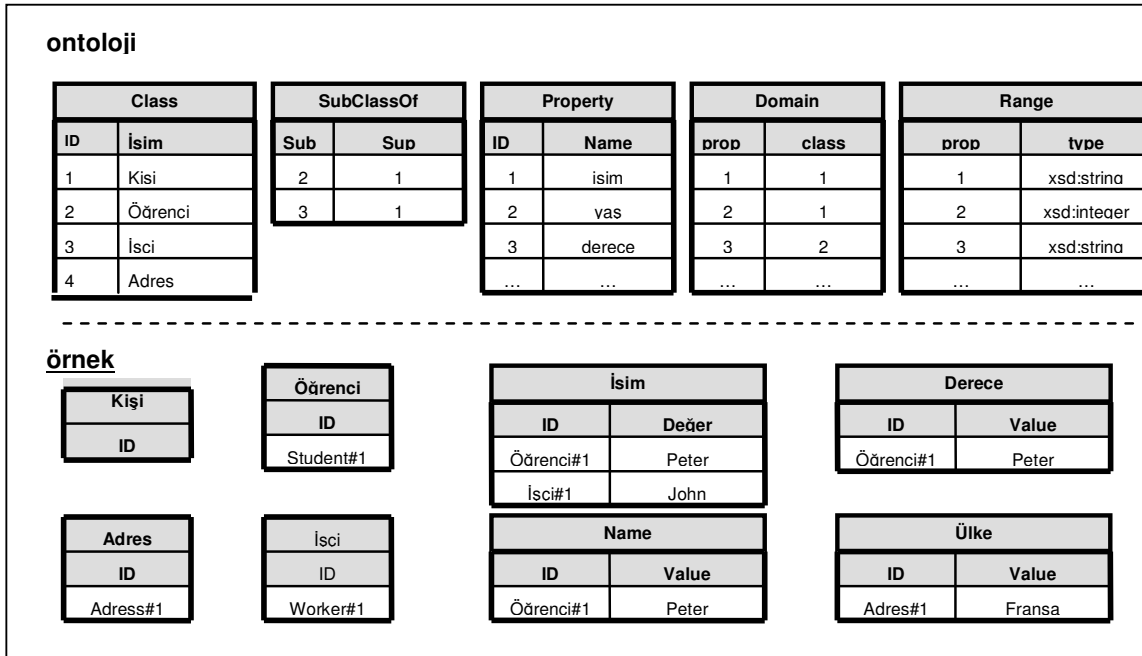
Birinci Tip Mimariler: Verinin tek bir şema altında, üçlüler şeklinde saklandığı mimarilerdir. Şekil 4.1 de gösterilen mimari “dikey tablo” olarak adlandırılan şema dokümanını ile örneklemektedir. Mimari üzerinde tek şema kullanımı, sorgulama süreçlerine yönelik olarak performans sorunlarına neden olmaktadır. Ancak performans artırıcı araştırmalardan kümeleme tekniklerin ilgili sorunu aşmak için kullanılmaktadır. Mimari üzerinde kullanılan ontoloji modeli *içsel* olarak geliştirilmiştir ve *statiktir*. Bu nedenle farklı ontoloji modellerinin birlikte kullanılmasına izin vermemektedir. Aşağıda yer alan şekilde bu tip mimarinin örneklenebilmesi için oluşturulan bir şema dokümanı sunulmaktadır.

Şeklin sol bölümünde *kişi* ontolojisi sınıf diyagramı ile birlikte sunulmakta, sağ bölümünde ise üçlüler şeklinde nasıl saklandığı gösterilmektedir.



Şekil 4.1 1.Tip OTV Mimarisi.

İkinci Tip Mimariler: Birinci tip mimarilere yeni bir şema dokümanı eklenmesi ile oluşturulmaktadır. Bu sayede mimari üzerinde veri ve verinin tariflendiği ontoloji ayrı ayrı saklanabilmektedir (Alexaki et al., 2001), (Broekstra et al., 2002). Bu tip mimariler üzerinde yapılan sorgulamalara daha hızlı cevap dönmesi birinci tip mimarilere göre daha başarılı bir yaklaşıma sahip olduğunu göstermektedir. Ancak bu tip mimariler farklı ontoloji modelleri tarafından desteklenmemektedir. Sadece oluşturulan ontolojinin yapılandırıldığı modele (örnek; RDFS, Owl, PLib) uygun olan veriyi saklayabilmektedir Bu nedenle esnek olmayan, statik bir yapı ile temsil edilmektedirler. Şekil 4.2 de ilgili mimari örneği Şekil 4.1 de sunulan *kişi* ontolojisine ait tabloların bu mimari üzerinde nasıl gösterildiğini belirtmek için sunulmaktadır. Buna göre ontolojide tanımlanan her sınıf için oluşturulmuş tablolar ile belirtilen sınıflara ait özellikler ise oluşturulan diğer tablolar üzerinde kolonlara atanan değerler ile saklanmaktadır.



Şekil 4.2 2.Tip OTV Mimarisi.

Üçüncü Tip Mimariler: İkinci tip mimarilere yeni bir şema dokümanı eklenmesi ile genişletilen mimarilerdir. Belirtilen şema “üst-şema” adını almaktadır. Hedef üst-şema dokümanı ile farklı ontoloji modellerini destekleyen esnek bir mimari sunmaktır. Farklı ontoloji modellerine (OWL, DAML + OIL, PLIB, vs.) uyumlu veriye erişim bu tip mimariler için performans ve sorgulama yeteneği için avantaj sağlamaktadır. Şekil 4.3 de belirtilen yapı OntoDB mimarisi geçerli olan üst-şema yapısını örneklemektedir.

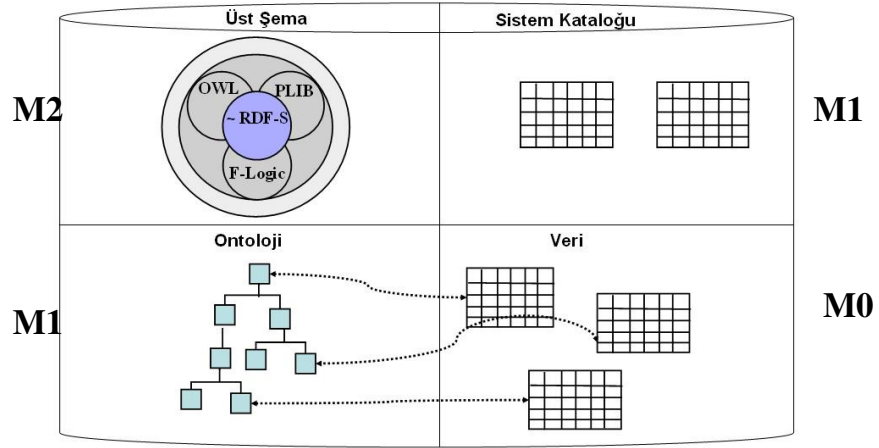
Entity			Attribute				Type	
ID	Name	SuperEntit	ID	Name	Domain	Range	ID	Name
1	Resource	1	1	name	1	1	1	String
2	Class	1	2	domain	3	2	2	Entity#2
3	Property	1						

Şekil 4.3 3.Tip OTV Mimarisi.

Tez çalışması üçüncü tip mimarilere örnek teşkil eden OntoDB mimarisi üzerinde gerçekleştirilmektedir. Bir sonraki bölümde bu mimariye ilişkin özellikler açıklanmaktadır.

4.2 3. Tip Ontoloji Temelli Veritabanı: OntoDB

OntoDB 3. tip bir Ontoloji Temelli Veritabanı mimarisidir. Farklı ontoloji modellerine ait ontolojilerin örnek dokümanlarının entegrasyonunu otomatik olarak yapma ve yönetme yeteneğine sahiptir. *İçerik* ve *ontoloji* olmak üzere iki kısımdan oluşmaktadır.



Şekil 4.4 OntoDB Mimarisi (Pierra et. all, 2005).

İçerik Bölümü OntoDB mimarisinin sağ tarafı ile temsil edilen bölümdür. Şekil 4.4 de gösterilen mimari üzerinde tutulan verilerin, veritabanı yönetim tablolarının ve ilgili ontolojilere ait örnek değerlerin saklandığı bölüm olarak da tariflenmektedir. Bu bölümde yer alan,

- **Sistem Kataloğu** klasik veritabanı mimarisine ait veri yönetim tabloları için gerekli olan sistem tablolarından oluşmaktadır. OntoDB mimarisi içinde yer alan diğer üç bölümün yönetimi bu bölüm üzerinden yapılmaktadır.
- **Veri Bölümü** mimari üzerinde yer alan tüm verilerin saklandığı bölüm olarak tariflenmektedir. ODM de yer alan MO seviyesine karşılık gelmektedir. Örnek değerleri içermektedir.

Ontoloji Bölümü: Şekil 4.4 ün sol tarafı ile temsil edilmektedir. Ontolojilerin yaratımı için gerekli sınıf ve öznitelik yapılarını içermekte, “üst-şema” ve “ontoloji” bölümlerden oluşmaktadır. Bu bölüm her hangi bir ontoloji modeline (RDF, OWL ve PLIB (ISO 13584-25)) ait ontolojileri oluşturabilmek için gerekli yapı taşları tanımlanmaktadır. OntoDB mimarisi PLIB ontoloji modeli ile yaratılmıştır bu modelin geliştirilmesinde ise EXPRESS dili yetenekleri kullanılmaktadır.

Mimari üzerinde sol tarafta yer alan Ontoloji bölümü iki kısımdan oluşmaktadır.

- **Üst-şema bölümü** bir ontoloji modeli tanımlayabilmek için gereken ontolojik varlıkları üst-şema seviyesinde saklamaktadır, bu nedenle M2 seviyesini temsil etmektedir.
- **Ontolojik Bilgi Bölümü** ilgili ontoloji modeline uygun alan ontolojilerin saklandığı bölümdür. M1 seviyesi olarak temsil edilmektedir.

OntoDB mimarisi üzerinde tutulan ontolojik veri *dikey yaklaşım* ile tariflenmektedir. Dikey yaklaşım içinde her ontolojik sınıf için bir tablonun yaratılması gerekmektedir. Tablo üzerinde var olan kolonlar bir sınıf tanımının sahip olduğu öznitelikleri içermektedir. Bir özneliğin yaratılması için ise öncelikle bir anahtar değerin (*id değeri*) üretilmesi gerekmektedir. İlgili öznitelik *nesne_özelligi* olarak tabloda “*isim, tarif, domain, range*” öznitelikleri ile birlikte yerini almaktadır. *Property_to_property* tablosu üzerinde ise sınıflar arasında ki ilişkiler tutulmaktadır. İçerik bölümü içinde yer alan örnek değerlerin oluşturulması ve örneğin saklanabilmesi için ise yine ilgili tablolar oluşturulmaktadır.

OntoDB mimarisi OntoQL tanımlama dili ile oluşturulmuştur. Bu dile ait yetenekler ise aşağıda sunulmaktadır.

4.3 OntoQL Dili

OTV ler de mantıksal/kavramsal modele ilişkin örnekler ontolojik veri olarak saklanmaktadır. Bu durum bir yanı ile kavramsal/mantıksal modele ilişkin verinin tariflenmesinde ontolojiye ilişkin kavram tanımının bir alt set olarak oluşturulmasını sağlamaktadır. Ancak diğer yanı ile tıpkı klasik veritabanlarında

olduğu gibi kavramsal model içinde ki tüm kavramların mantık model ile tariflenmesi gösterilmektedir. OntoDB mimarisi üzerinde geçerli olan ve kullanılan OntoQL tanımlama dili bu nedenle

- ontoloji üzerinde değişiklik yapmayı,
- ontoloji üzerinden kavramsal model oluşturmayı,
- yeni operatörler tanımlayabilmeyi,
- içerik değişikliği yapmayı,
- sorgulama yeteneklerine, içerik sorgulama, ontoloji sorgulama, ontolojiden içeriği sorgulama gibi çeşitlilik getirmeyi,
- formal olarak tutarlı anlamsallığı sağlama yönünde gerekli yetenekleri içerecek özellikleri kapsamaktadır.

Bu kapsamda OntoDB mimarisi üzerinde tanımlanan ve kullanılan OntoQL dinin yetenekleri arasında;

- **Standardları gözetmesi**, formal olarak ifade edilen bir ontoloji modeli anlamsal olarak mimari üzerinde tanımlanması için gerekli yapıları sunması (*Örn: Create Entity, Create Class, Insert Entity vs. (bkz Ek2)*)
- **Format değişimini desteklemesi**, ontolojiler için farklı ontoloji modellerinin kullanılmasını desteklemek yönünde üst-şema oluşturulması için gerekli yetenekleri kapsamakta ve bu sayede farklı ontoloji modellerini desteklemesi,
- **Veri değiştirilmesi**, kavramsal ontoloji modelleri ile oluşturulan ontolojik kavramlar ve örnekleri için ekleme, güncelleme, silime özellikleri yerine getirilebilmesi,
- **Dil desteği**, OTV için dil temelli isimlendirme ve kavramları farklı diller ile açıklama getirme özelliklerine sahip olması,

- **Veri sorgulaması**, etkin olarak hem ontolojiler hem de ontolojik verilerin sorgulamasını gerçekleştirilmesi, sorgulamaların içerik, ontoloji ve ontolojiden içerik ya da içerikten ontoloji yönünde yapılandırılması,
- **Ontoloji eşleme yeteneği**, farklı ontolojilerin entgre edilebilmesi için eşleme fonksiyonuna sahip olması gelmektedir.

Yukarıda sunulan yetenekler ile tez çalışması içinde geliştirilen Tercih Yönetim Çerçevesinin OntoDB mimarisi üzerinde ki gerçekleştirimi OntoQL dili ile sağlanmaktadır. Aşağıda bu bölüme ilişkin genel bir değerlendirme yer almakta, Bölüm 6 da ise OntoDB mimarisi üzerinde OntoQL dilinin kullanılarak geliştirilen kavramsal model açıklanacaktır.

4.4 Değerlendirme

İlişkisel veri modeli pek çok gerçek dünya veritabanı yönetimi için bir standart olarak kullanılmaktadır. Ancak yapılan alan yazın araştırmasında da gözlemlendiği gibi mantıksal modele sahip ilişkisel veritabanlarının yetersiz kaldığı noktalar için nesneye yönelik veritabanı tasarımı kullanılmaktadır. Bu sayede son derece karmaşık bilginin hızlı bir şekilde yönetilmesi mümkün kılınmaktadır. Bilindiği gibi veritabanı kendi içinde verinin anlamı hakkında hiç bir şey saklayamaz. Bu durum veritabanı uygulamalarında sınırlama oluşturmaktadır. Oysa anlam bilimsel veri modelleri ile veri yapısının geliştirilmesi sağlanmakta, anlaşılabilirlik, mirası destekleme, çıkarsama gibi avantajlar kullanılmaktadır.

OntoDB mimarisi anlamsal modeli içeren ontoloji temelli veritabanlarına bir örnek olarak sunulmaktadır. Tez çalışmasına dahil edilen bu mimari çekirdeğini oluşturan bir üst şema yapısına sahiptir. Bu sayede ontoloji modellerinin oluşturulmasında kullanılan üst-şema tanımlayıcılarına ve ontolojik bilginin sorgulanmasında kullanılan ontoloji sorgulama diline (OntoQL) sahiptir. Böylece girilen/tanılan üst modeller ışığında oluşturan ontolojik verilerin kullanılan sorgulama dilleri ile sorgulanabilmektedir.

OntoDB mimarisinin genişletilmesi sahip olduğu kavramsal modele ilişkin için üst-şema dokümanının genişletilmesi ile sağlanmaktadır. Yapılan

geniřletmenin geerlilięi iin OntoQL dili ile oluřturulacak sorgulamalar kullanılmaktadır, bylece mantıksal modele iliřkin sorgulamalar ile kavramsal modele iliřkin sorgulamalar birlikte deęerlendirilebilmektedir.

OntoDB mimarisinin Blm 3 de anlatılan OMG ye ait modelleme altyapısına sunulan model seviyeleri cinsinden ifade edilebiliyor olması, ODM de kullanılan temel kavramları ieren bir alt yapı sunması tez alıřması iinde oluřturulan Tercih Ynetim erevesi yaklařımının aıklanmasına katkı saęlamaktadır.

Bir sonraki blmde tez alıřması nerisi olarak sunulan Tercih Ynetim erevesi elemanları aıklanmaktadır. Blm 5 de sunulan kavramsal modelin gerekleřtirimi ise Blmde 6 da OntoDB mimarisi zerinde yapılmaktadır.

5 TERCİH YÖNETİM ÇERÇEVESİ

Alan bilgisi üzerinde yer alan her örnek değer, kullanıcıların seçimi ile tercihler olarak temsil edilen varlıklara dönüşmektedir. Bu nedenle tez çalışması boyunca kişisel tercihlerin alan bilgisi üzerinden elde edildiği kabulü yapılmakta, bu kabulün gerçekleştirimi için ise Bölüm 2 de incelenen Kaynak tanımı Tercih Yönetim Çerçevesi içinde kullanılmaktadır. Bu tanım sayesinde alan bilgisi üzerinde yer alan tüm varlıkları kapsayacak genel bir yapının modellenmesi hedeflenmektedir.

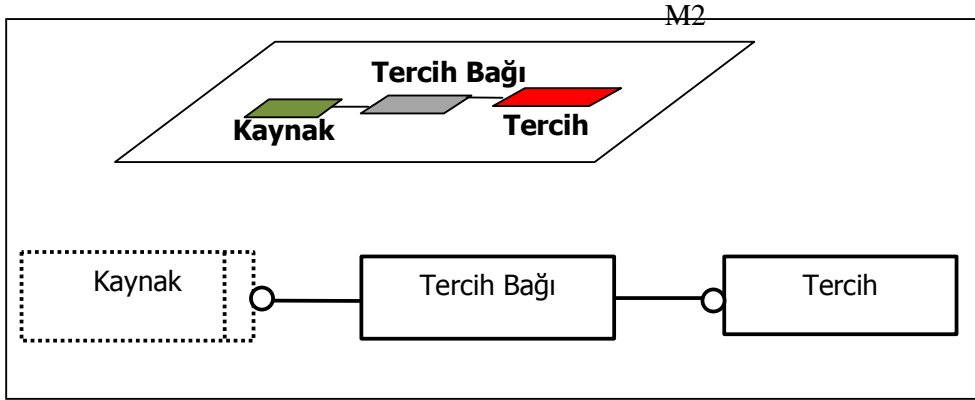
Kişiselleştirilmiş bilgiye ulaşabilmek adına, alan bilgisinden bağımsız oluşturulacak tercih tanımlarına ihtiyaç duyulmaktadır. Yapılan alan yazın çalışmasında (bkz. Bölüm 2) Veritabanı ve Anlamsal Web araştırmaları için *nitel* ve *nicel* yaklaşımlar ile ifade edilen sınıflandırmalara ve bu sınıflandırmalar altında yer alan birçok farklı tercih tipi tanımına ulaşılmaktadır. Ancak farklı tercih tiplerini bir araya getirebilmek için her iki yaklaşımı kapsayan genel bir çözüm modeline literatürde rastlanılmamaktadır. Bu sebeple geliştirilen Tercih Yönetim Çerçevesinin her iki yaklaşımı destekleyecek şekilde bir Tercih üst-şeması içermesine, belirtilen şema içinde *nicel yaklaşımlar* ile tariflenen tercih tipleri (aralık, oransal, boolean, sayısal) için gerekli tanım fonksiyonlarının kullanılmasına, *nitel yaklaşımlar* ile tariflenen tercih tipleri için ise yan tercih kavramlarını belirtmekte kullanılacak gerekli bir tanımın üst-şemaya dahil edilmesine karar verilmiştir. Bu sayede, Tercih üst-şeması içinde kullanılan tercih tarifleri ile alan bilgisinin sorgulanmasına yönelik bir çözüm önerisinin de çalışmaya eklenebileceği düşünülmektedir.

Tez çalışması kapsamında kişiselleştirilmiş bilginin oluşturulabilmesi için gerekli alt yapının Tercih Yönetim Çerçevesi elemanları ile açıklanmıştır. Bu nedenle Tercih Yönetim Çerçevesi;

- Anlamsal Web araştırma alanını desteklemektedir. Anlamsal Web araştırmalarında alan bilgisi ontolojiler ile temsil edilmektedir. Bu nedenle ontolojik bilgiye yönelik bir modelin oluşturulması için Bölüm 3 de anlatılan kavramsal model oluşturma yaklaşımı kullanılmaktadır.

- farklı tercih tiplerinin bir araya getirildiği bir model sunmaktadır. Bölüm 2 de yer alan, alan yazın içinde örneklenen araştırma alanları içinde tanımlanan tercih tipleri için genel, genişletilebilir ve modüler bir model oluşturma fikrine yönelik olarak bir Tercih üst-şeması tanımlanmaktadır.
- alan bilgisinden ayrıık olarak oluşturulan Tercih üst-şeması ile alan bilgisinin nasıl ilişkilendirildiği gösterilmektedir.

Yukarıda belirtilen üç ana maddeyi temel alan Tercih Yönetim Çerçevesi Şekil 5.1 de sunulmaktadır. Kaynak, Tercih ve Tercih Bağlantısı olmak üzere üç kısımda incelenen çerçeve ve M2 seviyesinde tanımlanmaktadır.



Şekil 5.1 Tercih Yönetim Çerçevesi.

Kaynak, kullanıcıların tercihlerini ifade edebilecekleri bir altyapı olarak kabul edilmektedir. Ontoloji Tanımlama Dili (ODM, 2006) ile açıklanan temel ontolojik yapı taşlarına ait tanımları içermektedir (bkz. Bölüm 3).

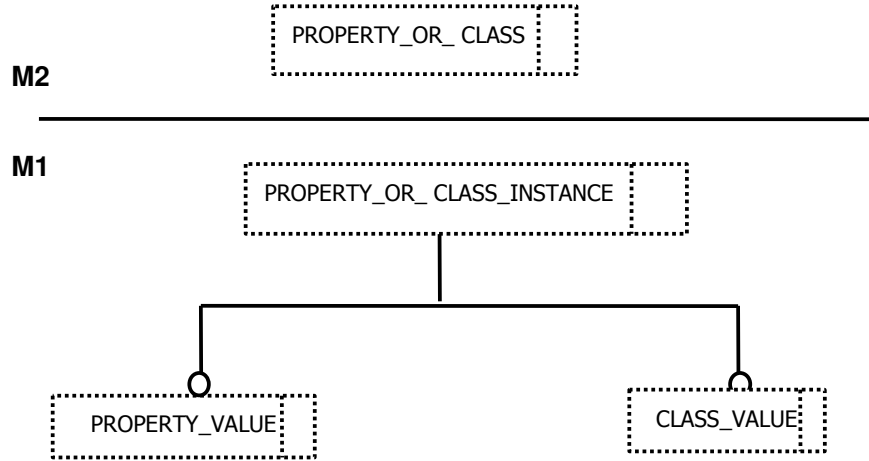
Tercih, farklı tercih tiplerini bir araya getirmek için tanımlanan ve Tercih üst-şemasını temsil eden üst varlık olarak kabul edilmiştir. Farklı kategorilerde ele alınan tercih tiplerini içermektedir (bkz. Bölüm 2).

Tercih Bağı, Tercih üst-şeması ile Kaynak içinde yer alan örnek değerlerin birbirleriyle ilişkilendirilebilmesi için yaratılan varlıktır.

Aşağıda öncelikle oluşturulan Tercih Yönetim Çerçevesi elemanları açıklanmakta, daha sonra Kısım 5.4 de modele ilişkin genel bir değerlendirme sunulmaktadır.

5.1 Tercih Yönetim Çerçevesi ve Kaynak Tanımı

Kişisel tercih bilgisinin oluşturulabilmesi için alan bilgisi, Bölüm 2 de açıklanan Kaynak varlığı ile temsil edilmektedir. Kaynak varlığı bir ontolojik bilgi olarak ele alınmakta, *öznitelik* ve *sınıf* varlıklarına ilişkin örnek değerleri içerecek şekilde tanımlanmaktadır. Şekil 5.2 de gösterilen bu tanım Tercih Yönetim Çerçevesi içinde M2 seviyesinde incelenmektedir. Aşağıda Kaynak varlığı için oluşturulan model bilgisi ve örnekleri yer almaktadır.



Şekil 5.2 Tercih Yönetim Çerçevesi için Kaynak Tanımı.

Model. EXPRESS modelleme dili kullanılarak oluşturulan Kaynak tanımı MDA seviyeleri cinsinden Şekil 5.2 de sunulmaktadır.

- PROPERTY_OR_CLASS adıyla tanımlanan varlık, ODM üzerinde yer alan ontoloji elemanlarını üst-şema (M2) seviyesinde temsil etmektedir.
- PROPERTY_OR_CLASS varlığından türeyen ve aralarında <<örneğidir>> ilişkisi olan PROPERTY_OR_CLASS_INSTANCE varlığı ise ontolojiler içinde yer alan tüm örnekleri ve öznitelik

değerlerini kapsamaktadır. M1 seviyesinde seçimli bir tip olarak tanımlanmaktadır. İlgili seçim `PROPERTY_VALUE` ya da `CLASS_VALUE` tipleri arasında yapılmaktadır.

- `CLASS_VALUE` ilgili alan bilgisi içinde tanımlanan sınıflara ait örnek `EXPRESS` değerlerini,
- `PROPERTY_VALUE` her bir sınıf tanımına ait özniteliklere atanan değerleri tutmak için tanımlanmaktadır.

Aşağıda `class_value` ve `property_value` örnek değerleri `EXPRESS` modelleme dili ile oluşturulan örnek gösterimleri ile sunulmaktadır.

```
#1=CLASS_VALUE ('HotelMercury');

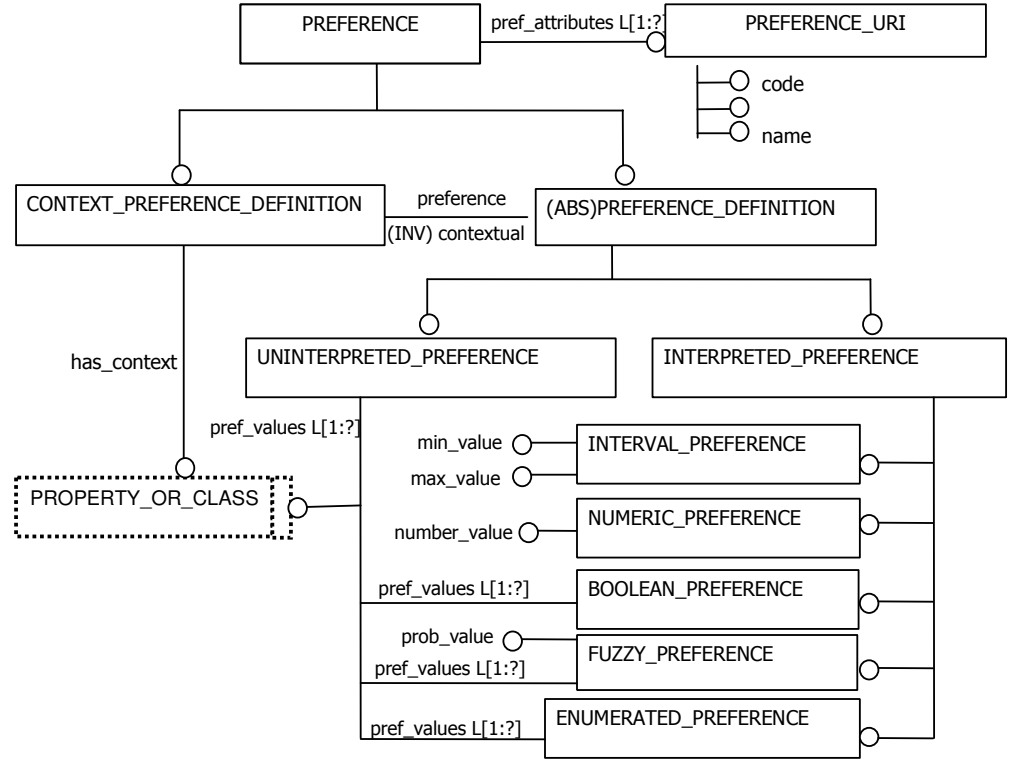
#2=PROPERTY_VALUE ('starRate', 5);
```

Bu kısımda açıklanan Kaynak elemanı kullanıcı tanımlı bir tip olarak oluşturulmakta ve Tercih üst-şeması içinde kullanılmaktadır.

5.2 Tercih Üst-Şeması

Tez çalışması içinde ortaya konan Tercih Yönetim Çerçevesi için Tercih üst-şeması, farklı tercih tiplerinin temsili için oluşturulan hiyerarşik bir yapıyı sunmaktadır. Tez çalışmasında ortaya konan yaklaşım, tercihlerin tanımlarını oluşturulan tercih-üst şemasından, değerlerini alan bilgisinden aldıklarını gösterme yönündedir. Bu nedenle Bölüm 5.1 de açıklanan ve kullanıcı tanımlı bir tip olarak yaratılan Kaynak tanımı Tercih Üst-şeması içinde tercih değerlerini gösterecek şekilde kullanılmaktadır.

Aşağıda yer alan Şekil 5.3 oluşturulan Tercih Üst-şemasını sunmaktadır.



Şekil 5.3 Tercih Üst-Şeması.

EXPRESS dili ile oluşturulan şema içinde yer alan *Preference* varlığı alan yazında (bkz: Bölüm 2) sunulan *metinsel*, *boolean*, *sayısal*, *aralık* ve *bağlam temelli* tercih tiplerini bir arada toplayan bir üst-varlıktır. *Preference_URI* varlığı ile ilişkilendirilmiştir.

Preference_URI varlığı *nitel* yaklaşımları temsil eden örneklerin tariflenmesinde kullanılmaktadır. Tanımı gereği tercihlerin genel ifadeler şeklinde alan bilgisinden bağımsız olarak (örn: *ucuz*, *yakın*, *lüks*) sunulabileceğini göstermektedir. Alan bilgisi üzerinde yer alan örnek değerler ile bir araya getirildiğinde bir tercih niteleyicisi (örn: *ucuz otel*, *yakın otel*, *lüks otel*) tanımını oluşturmaktadırlar. Bu nedenle oluşturulan Tercih üst-şeması içinde yer alan tüm tercih tipleri için kullanılması gerektiği kabulü ile *Preference* üst-varlığına direk olarak Şekil 5.3 de gösterildiği gibi ilişkilendirilmiştir.

Preference varlığı *nicel* yaklaşımlar ile ifade edilen genel tercih tanımlarını (*Preference_Definition*) ve tanımlandığı bağlama göre gerekli değer aralığı ile ifade edilecek şekilde bir tanım içeren bağlam temelli

tercih tanımlarını (Context_Preference_Definition) içermektedir. Genel olarak tercih tanımları, (Preference_Definition) yorumlanabilir (Interpreted_Preference) ve yorumlanamayan (UnInterpreted_Preference) tercihler olarak ikiye ayrılmaktadır. *Yorumlanabilir tercihler*, alan yazında sunulan ve farklı araştırmalarda ortaya konan, kendilerine ait tanım fonksiyonu içeren tercih tipleri için oluşturulmuştur. Tercih üst-şeması üzerinde beş farklı tip ile temsil edilmektedir. *Yorumlanamayan tercihler* ise, yorumlanabilir tercihlerin dışında kalan tercihler olarak düşünülmelidir. Herhangi bir değerlendirme fonksiyonu ile gösterilemeyen ancak bir set halinde bir arada tutulabilen ayrık tercihlerden oluşmaktadır. Bu tür tercihler ile bir tercih yığını oluşturacak durumların örneklenmesi hedeflenmektedir.

Tercih üst-şeması içinde yer alan son tercih tipi ise incelendikleri bağlama göre farklı değer aralıkları ile ifade edilebilen *bağlam temelli tercihlerdir* (Context_Preference_Definition). Bu kategoriye ilişkin tercih tipi tanımlaması için farklı bir çalışma alanının literatürde yer aldığı bilinmektedir. Oluşturulan üst-şemaya katkı sağlayıcı olabilmesi için ele alınmakta, Tercih Yönetim Çerçevesi içinde ise ayrı bir çalışma olarak incelenmesi gerektiği vurgulanmaktadır.

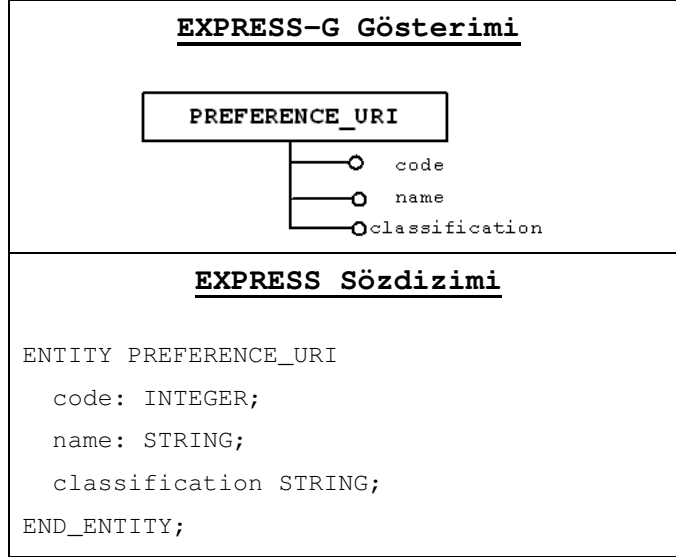
Yukarıda kısaca açıklanan Tercih üst-şeması, aşağıda oluşturulan EXPRESS kodları ve EXPRESS-G gösterimleri ile birlikte sunulmaktadır.

5.2.1 Tercih Niteleyici: Preference_URI Tanımı

Tercih Yönetim Çerçevesi içinde *nitel* olarak tariflenen bir tercihin *nicel* olarak ifade edilmesi hedeflenmektedir. Bu nedenle Tercih üst-şeması içinde *tercih niteleyicisi* kavramı öne çıkartılmaktadır. *Tercih niteleyicisi*, bir tercihin anlamsal olarak ifade edilebilmesi için gereken yan kavramları tariflenmektedir ve üst-şema içinde yer alan tüm tercih tipleri için kullanılmaktadır. Belirtilen yan kavramlar (örn: *ucuz*, *yakın*, *lüks*) ancak alan bilgisi ile birlikte kullanıldığında kullanıcıların tercih kısıtlarını sunmaktadır (örn: *ucuz otel*, *yakın otel*, *lux otel*). Günlük yaşantıda kişisel tercihlerin belirtilmesinde yaygın olarak kullanılan bu kavramlar için oluşturulan Preference_URI tanımı bir tercih tipi tanımının genişletilebilir olmasını sağlamakta, ilgili tanıma

verilebilecek örneklerin çeşitliliğini arttırmaktadır. Aşağıda *tercih niteleyicisi* tanımının Tercih üst-şeması içinde nasıl yaratıldığı gösterilmektedir.

Çizelge 5.1 Preference_URI



Model. Çizelge 5.1 de yer alan Preference_URI varlığına ait EXPRESS tanımı, “*code, name, clasifcation*’ olmak üzere üç özniteliği içermektedir.

- *code özniteliği* ilgili tercih tarifine ait bilginin tutulduğu *http* adresi (bir URI bağlantısı) olarak kabul edilmektedir. Bu kabul doğrultusunda tüm tercih niteleyicilerinin gerçekte bir ontolojik bilgi üzerinden elde edildiği vurgulanmakta, bir kaynak varlığı temsil ettiği gösterilmektedir. Aşağıda yer alan Preference_URI örneklerinde ilgili öznitelik bir URI adresi olarak değil, bir id değeri olarak temsil edilmektedir.
- *name öznitelği*, tercihin anlamsal olarak karşılık bulduğu ifadeyi bir nitelik ismi olarak tariflenmektedir.
- *classification öznitelği*, tercih niteleyicinin içinde yer aldığı kategoriye sunmak için kullanılan genel bir ismi göstermektedir.

Bu sayede kullanıcıların tercihleri tariflenmekte kullandıkları tercih niteleyicilerin, tanımlamalarının yapıldığı adres bilgileri ve kategorileri ile birlikte

gösterilmeleri sağlanabilmektedir. (örnek: (193.148.12, “yakın”, “mesafe”), (193.158.2, “luks”, “kalite”).

Temsil edilen her kategori (*classification*) içinde ise birden çok tercih niteleyeci tarifinin olabileceği göz ardı edilmemektedir (örnek: *mesafe kategorisi için; yakın, çok yakın, uzak, çok uzak gibi...*). Bu durumda her bir tercih niteleyici tarifinin ilişkili olduğu bir tercih tanımı ile değerlendirmesi gerektiği fikri kabul edilmekte, ilişkilendirildiği tercih tipi tanımına uygun bir fonksiyon ile gerekli değerleri alabileceği vurgulanmaktadır.

5.2.2 Yorumlanabilir Tercihler: Interpreted_Preferences Tanımı

Yorumlanabilir tercih tipleri Bölüm 2 de sunulan çalışmaları bir araya getirebilmek için birçok farklı tercih tipinden oluşmaktadır. Bu kategoriye giren tüm tercih tipleri için bir değerlendirme (örn: kümeleme, boolean, olasılık) ya da yorumlama fonksiyonu ($ucuz(x)$ için $fıyat(x) \leq 20$) bulunmaktadır. Bu sayede tercihler sahip oldukları veri tipleri ile ilişkili olarak tanımlanabilmekte ve belirlenen sıralama ilişkisi ($<, >$) ile ifade edilebilmektedir. Aşağıda bu kategoriye ait tanımlanan beş farklı tercih tipi literatürde yer alan örnekleri ile birlikte açıklanmaktadır.

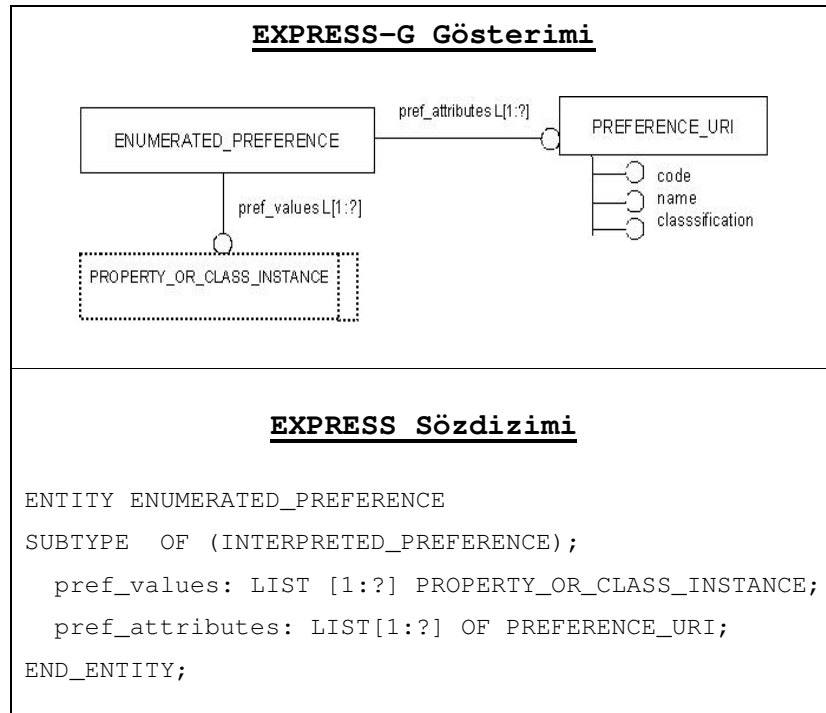
5.2.2.1 Listelenmiş Terciler: Enumerated_Preference Tanımı

Listelenmiş tercihler, matematik ve teorik bilgisayar bilimlerinde *bir set içinde olgu popülasyonunun gösterilmesi* olarak tariflenmektedir. Tercih üst-şeması içinde Enumerated_Preference tipi ile ontolojik olguların bir küme olarak nasıl ilişkilendirildiği açıklanmaktadır.

Model. Çizelge 5.2 de tanımlanan Enumerated_Preference varlığı,

- *pref_values* özniteliği içinde CLASS_VALUE ve PROPERTY_VALUE örneklerini bir set olarak,
- *pref_attributes* özniteliği içinde ise Preference_URI tercih niteleyicisine ait id leri tutmaktadır.

Çizelge 5.2 Enumerated_Preference



Tanımı gereği liste halinde sınıf örneklerini ya da bir sınıfın sahip olduğu öznitelik örneklerinin tercih olarak gösterilmesi hedeflemektedir.

Bu tercih tipi tanımı ile kullanıcının lux IBIS Otellerini ya da Akdeniz bölgesinde ki ucuz otelleri tercih etmesi durumu örneklendirilmektedir.

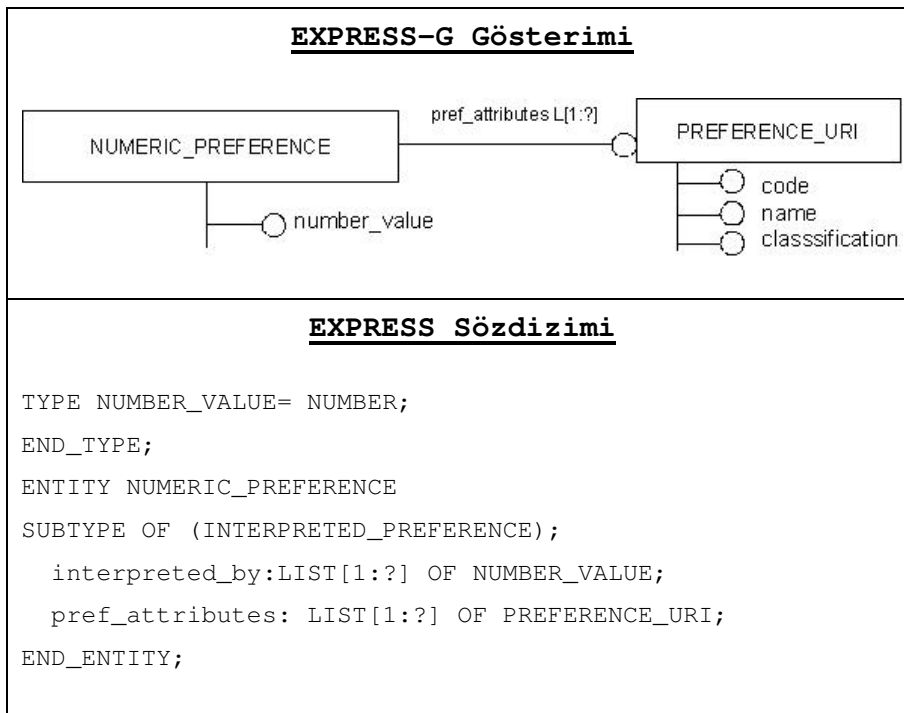
5.2.2.2 Sayısal Tercihler: Numeric Preference Tanımı

Fischer ve ekibi araştırmalarında sayısal veri tipleri ile ifade edilen tercihlere ilişkin durumları örneklemektedirler (Fischer et al., 2006). Oluşturdukları tanım gereği sayısal ifadeleri en büyükten en küçüğe (ya da tersi) doğru sıralayan bir skorlama fonksiyonu içinde belirtmekte ve oluşturulan skorların sayısal tercihler olarak nasıl kullanıldığını açıklamaktadırlar. Benzer bir yaklaşım Tercih Yönetim Çerçevesi içinde de yer almaktadır. Sayısal veri tipi ile tariflenecek tercihler için kullanılan bu tercih tipi, bir skor fonksiyonuna ihtiyaç duymadan sayısal ifadelerin tercih niteleyiciler ile birlikte bir tercih olarak belirtilmesini sağlamaktadır.

Model. Çizelge 5.3 de tanımlanan Numeric_Preference,

- *interpreted_by* özneliği ile NUMBER_VALUE olarak tanımlanan sayısal değer alabilecek veri tipini,
- *pref_attributes* özneliği ile Preference_URI tercih niteleyicisine ait id değerleri saklanmaktadır.

Çizelge 5.3 Numeric_Preference



Bu sayede alan bilgisi üzerinde sayısal öznelik örnek değerleri için tercihler oluşturulmaktadır.

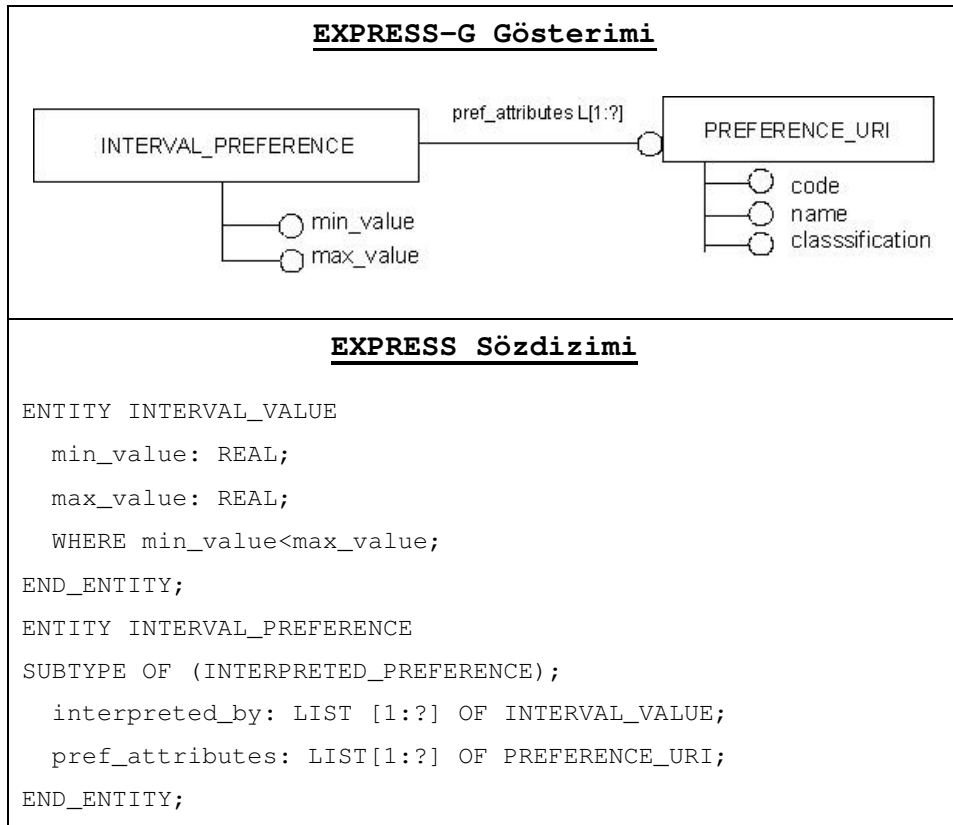
2 yıldızlı bir oteli kullanıcının ucuz otel olarak tercih etmesi durumu sayısal tercih tipine verilebilecek bir örnektir.

5.2.2.3 Aralık Tercihleri: Interval Preference Tanımı

Öztürk çalışmasında aralık değerleri ile tarif edilebilen tercihlerin örneklendiği durumları incelemekte, tercihlerin alt ve üst kısıt değerlerine bağlı olarak ele alınması gerektiği vurgulanmaktadır (Oztürk and Tsoukiàs, 2005). Bu nedenle Aralık Tercihleri Tercih Yönetim Çerçevesi içinde benzer bir yaklaşım ile ele alınmakta, alan bilgisi üzerinde ki bir örnek değer ile ilişkilendirilen tercih niteleyicilerin yorumlanmasında kullanılmaktadır.

Model. Çizelge 5.4 de verilen EXPRESS kodunda gösterildiği gibi öncelikle INTERVAL_VALUE veri tipi yaratılmakta, bu tip içinde *min_value* ve *max_value* öznitelikleri en küçük ve en büyük tamsayı değerlerini temsil edebilmek adına oluşturulmaktadır. Böylece Aralık Tercih tipine ait,

Çizelge 5.4 Interval_Preference



- *interpreted_by* özniteliği içinde *min_value* ve *max_value* özniteliklerini ikililer olarak tutan *INTERVAL_VALUE* tipine ait id değeri,
- *pref_attributes* özniteliği içinde ise örneklenebilecek *Preference_URI* tercih niteleyicilerine ait id değerleri tutmaktadır.

Bu tercih tipi için kullanıcının pahalı otel tercihinde bulunması durumu, ücret özniteliğine ait bir örnek değerin ([100-200] aralığında olan oteller için) pahalı olarak belirtilmesi ile örneklendirilebilmektedir.

5.2.2.4 Olasılık Tercihleri: Fuzzy Preference Tanımı

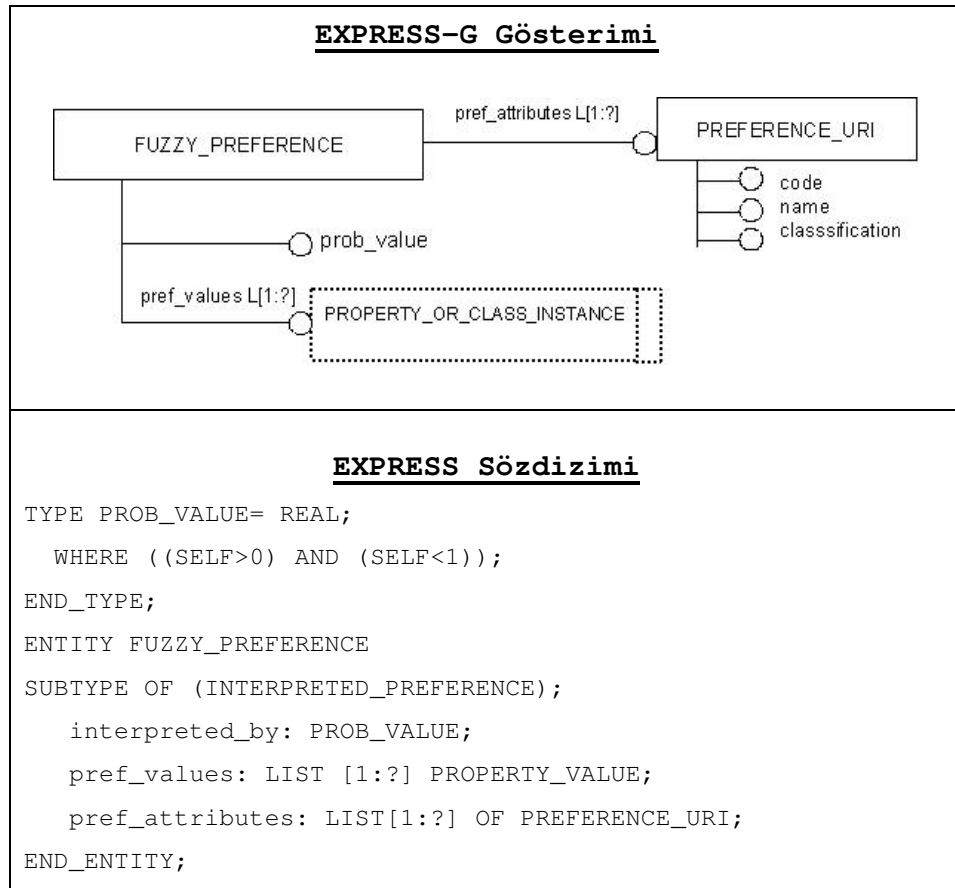
Tercihlerin belirsiz olduğu durumlarda belirli bir olasılık değerleri ile ifade edilmesi gerektiği Dubois'in çalışmasında ele alınmaktadır (Dubois and Prade., 1998). Bu tür tercihlerin Fuzzy model tanımlamaları ile birlikte sunulmaları gerektiği vurgulanmaktadır, 0 ile 1 arasında değişen bir olasılık değerleri içermeleri gerektiği farklı araştırmalarda sunulmaktadır (Pini *et al.*, 2005). Tercih Yönetim Çerçevesi içinde ise benzer bir yaklaşım olasılık tercih tipi tanımı içinde ele alınmaktadır. Ancak literatürde bulunan örnek modelleri kapsayan tam bir çözüm çalışmada sunulmaktadır. Bu nedenle Tercih Yönetim Çerçevesine dışsal bir model olarak olasılık modelinin dahil edilmesi gerektiği fikri tez çalışması içinde hedeflenen gelecek çalışma olarak verilmektedir.

Model. Olasılık Tercihleri, Çizelge 5.5 de gösterildiği gibi bir olasılık değeri ile temsil edilmektedir. Bu nedenle *prob_value* özniteliği tanımlanmakta, ilgili özniteliğin 0 ile 1 değerleri arasında bir değere sahip olması gerektiği kısıtı verilmektedir. Oluşturulan tanımda,

- *interpreted_by* özniteliği, ile *prob_value* tipine atanan değerler,

- *pref_value* özneliği ile ise kaynak tanımından gelen PROPERTY_VALUE özneliğine ait değerler,
- *pref_attributes* özneliği ise Preference_URI tercih niteleyicilerin id leri yer almaktadır.

Çizelge 5.5 Fuzzy_Preference



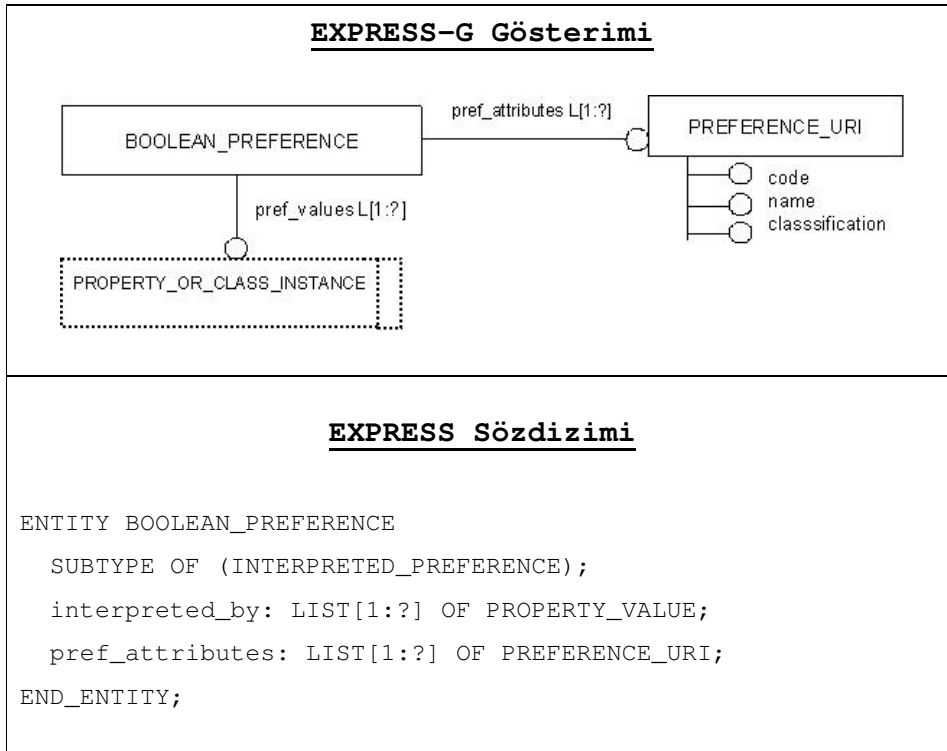
Bu sayede belirlenen olasılık değerleri ile seçilen örnek değerlere yönelik ihtimal değerleri verilebilmektedir.

Kullanıcının lux otel tercihi için 0.50 ihtimalle golf sahası, 0,50 SPA özelliklerini istemesi oluşturulan tercih tipine verilen bir örnek durumdur.

5.2.2.5 İkili Tercihler: Boolean Preference Tanımı

İkili tercihler, PROPERTY_OR_CLASS_INSTANCE varlığı ile temsil edilen örnek değerler için tercih edilen ya da edilmeyen durumların ifadesinde kullanılmaktadır. Siberski'nin araştırmasında belirtilen tercih tipinin tanımı oluşturulmaktadır. İlgili çalışmada ikili tercihler Boolean veri tipi ile ifade edilen öznitelik tanımları için oluşturulmaktadır. Ancak bu tanımlama sınıf örneklerine ait yapılacak bir ikili tercih tanımını içermemektedir. Bu durumda da bir sınıf örneği, ikili tercih olarak tercih edilen ya da edilmeyen bir durum için kullanılmamaktadır [Siberski et al., 2006]. Tercih Yönetim Çerçevesi içinde ise Boolean_Preference tanımı bu eksikliği giderecek şekilde tanımlanmaktadır. İlgili tercih tipi PROPERTY_OR_CLASS_INSTANCE varlığından değer alacak şekilde oluşturulmaktadır.

Çizelge 5.6 Boolean_Preference



Model. Çizelge 5.6 da belirtilen tercih tipi tanımı içinde yer alan özniteliklerden,

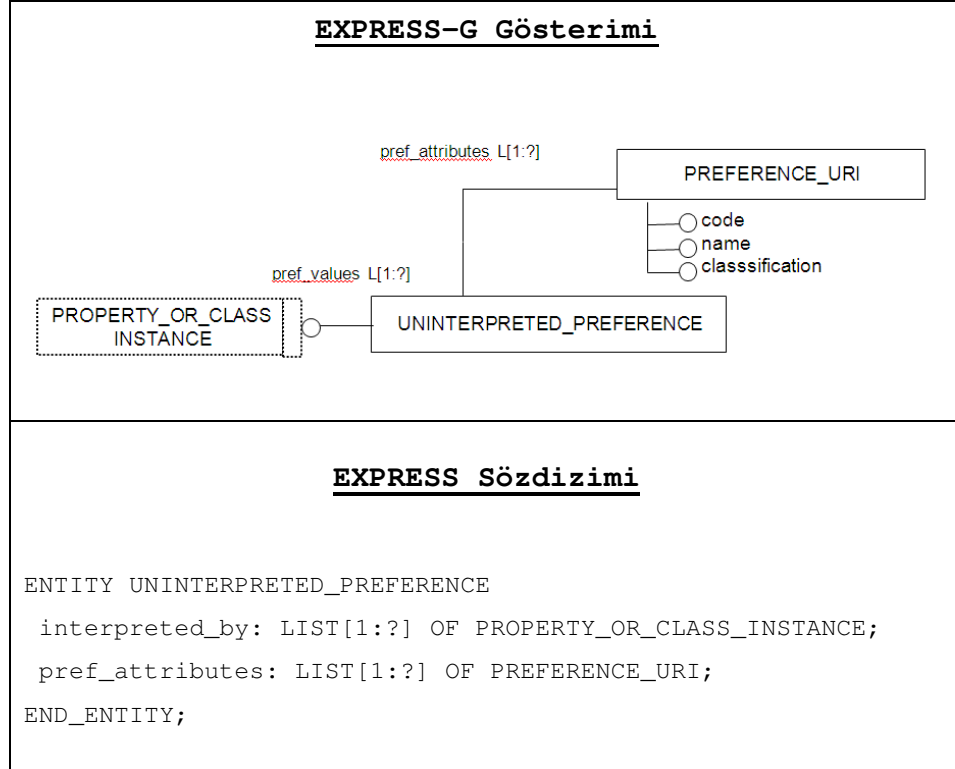
- *interpreted_by* özneliği PROPERTY_OR_CLASS_INSTANCE tanımdan gelen PROPERTY_VALUE öznelik değerleri, ya da CLASS_VALUE değerleri,
- *pref_attributes* özneliği ise Preference_URI tercih niteleyicilerine ait id leri saklanmaktadır.

Belirtilen tercih tipi ile bir kullanıcı lüks otel tercihini casinosu, spası, golf sahası olan bir Hilton Hoteli olsun şeklinde sunmaktadır.

5.2.2.6 Yorumlanamayan Tercihler

Yorumlanamayan tercihler tanımı, herhangi bir yorumlama sağlayıcı prosedüre sahip olmaksızın bir araya getirilmiş ontolojik örnekleri bir set olarak göstermektedir.

Çizelge 5.7 UnInterpreted_Preference



Model. Çizelge 5.7de gösterildiği şekilde,

- *interpreted_by* özneliği ile `PROPERTY_OR_CLASS_INSTANCE` tanımdan gelen değerler,
- *pref_attributes* özneliği ile ise `Preference_URI` tercih niteleyicilerine ait id ler saklanmaktadır.

Bu tercih tipi ile kullanıcı alan bilgisi içinde yer alan örnek değerler, lüks tercih tarifi altında BMW, pizza, jakuzi, Milano, ipod gibi ayrık değerlerden oluşan bir liste içinde belirtebilmektedir.

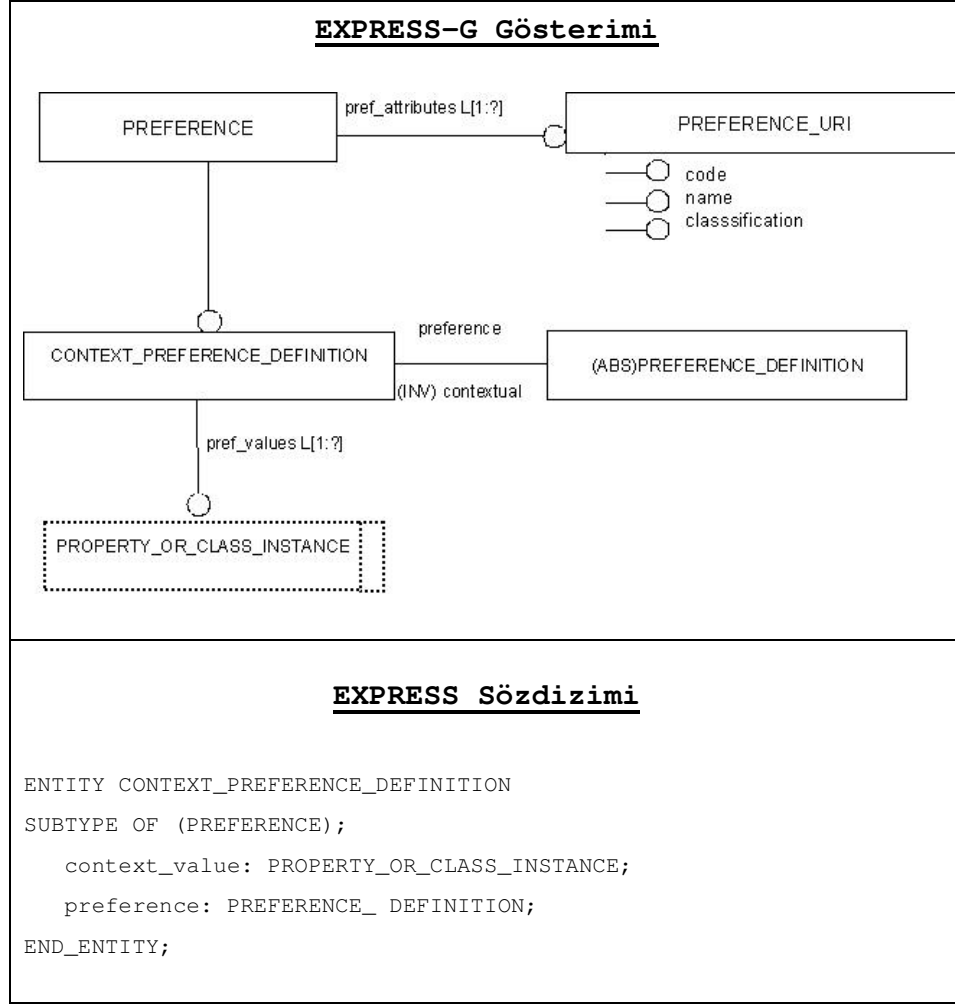
5.2.3 Bağlam Temelli Tercihler: Context-Based-Preference Tanımı

Bağlam, belli bir dilsel ifadeyi içine alan herhangi başka bir ifade olarak açıklanmaktadır. Tercih üst-şeması içinde örneklenmesi gereken yeni bir kategoriye temsil etmektedir. Bu sayede tercihlerin farklı durumlar karşısında farklı değerler alabileceği örneklendirilmektedir.

Model. Çizelge 5.8 de oluşturulan EXPRESS kodu içinde öncelikle literatürde yer alan farklı Bağlam Modelleri incelenmiş (Hwang et. All., 2005), (Sandkuhl et. All., 2009), ortak öznelikler kullanılarak belirtilen tercih tipi oluşturulmuştur. Ancak çakışmaların oluşacağı söz konusu durumlar için bağlam temelli tercih tipinin ayrı bir model olarak ele alınması gerektiği ve Tercih üst-şeması içinde yer alan tüm tercih tipleri için tek tek incelenmesi gerektiği belirlenmiştir. Aşağıda sunulan tanım gereği oluşturulan,

- *context_value* özneliği `Property_or_Class_Instance` kaynak tanımından gelen olguları,
- *pref_attributes* özneliği ile ise `Preference_URI` tercih niteleyicilerine ait id leri,
- *preference* özneliği ise `Preference_Definition` ile tariflenen tercih tiplerinden birini içermek üzere oluşturulmuştur.

Çizelge 5.8 Context_Based_Preference



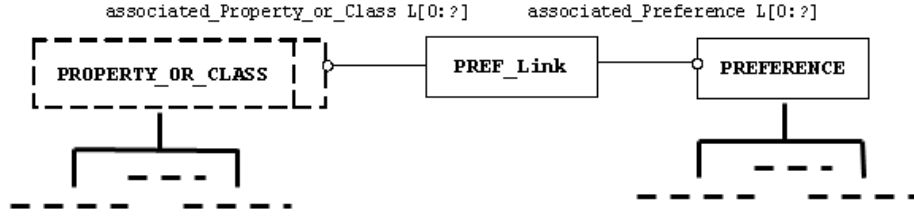
Böylece seçilecek her hangi bir tercih tipi için oluşturulacak tercih örneğinin tanımlandığı bağlama göre farklı değer alacağı gösterilebilmektedir.

Örneğin; kullanıcının ucuz ve pahalı otel tercihinde bulunmasının seçilen yer bilgisine göre farklı yorumlanacak olması durumu. Alan bilgisinde yer alan Paris ve İzmir class_value örnek değerlerine, aralık tercih tipi tanımı ile ulaşılması durumunda ilgili aralık değerlerinin farklı tercih niteleyicileri ile yorumlanması söz konusudur. Ya da aynı tercih niteleyicilere farklı değer aralıklarının atanması ile yine tercihlerin yorumlanması farklı olacaktır. Böylece aynı min_value ve max_value değerleri Paris de bulunan bir oteli için ucuz, İzmir bulunan bir otel için ise pahalı olabilmekte ya da ucuz

olarak tariflenen bir otel Paris ve İzmir için ayrı değer aralık değerlerini sahip olmaktadır (Paris için (70-100 euro) İzmir için (40-70 euro) gibi).

5.3 Tercih Bağı

Tercih Yönetim Çerçevesinin son elamanı Şekil 5.1 de gösterildiği gibi Tercih Bağıdır. Belirtilen varlık Kısım 5.1 ve 5.2 de açıklanan *Property_or_Class* kaynağını üst şema seviyesinde Tercih varlığı ile ilişkilendirebilmek adına yaratılmaktadır. Şekil 5.4 de gösterildiği gibi Kaynak tanımı olarak belirtilen ontolojik örneklerin Tercih tipleri ile üst-şema seviyesinde ilişkilendirilmesi için oluşturulmaktadır.



Şekil 5.4 Tercih Bağı

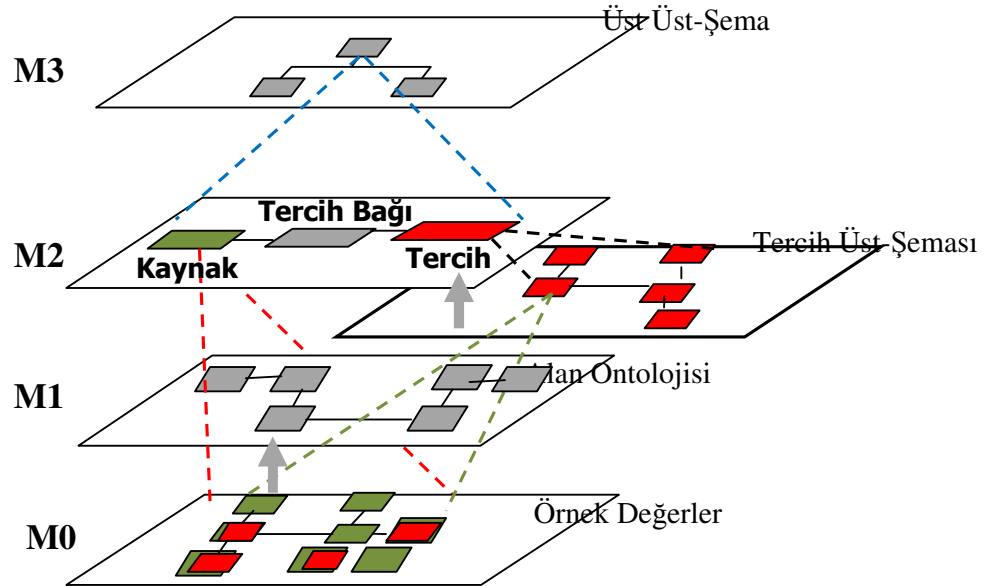
Burada hedef Tercih üst-şemasında tariflenen tüm tercih tiplerinin alan bilgisi üzerinde yer alan örnek değerler ile eşleştirilmesidir. Kurulan bağ, alan bilgisinden bağımsız olarak ortaya konan tercih tiplerine ait tanımları, tercih niteleyiciler ile birlikte alan bilgisi üzerinde yer alan örnek değerlere atamaktadır. Oluşturulan Tercih Bağı örneği Bölüm 7 de yer alan uygulama içinde gösterilmektedir.

5.4 Değerlendirme

Tercihler karar verme odaklı süreçler içinde değerlendirilmektedir. Günlük yaşantıda belirtilen süreçler için öncelikle tercihlerin ifade edilebiliyor olması gerekmektedir. Bu durum incelendiğinde tercihlerin alan bilgisi üzerinde kullanılan örnek değerler ve bu değerlere atanan tercih niteleyici isimler ile tariflendiği gözlenmektedir. Ancak tezde öne çıkarılan bu yaklaşım bir başka araştırmada yer almamaktadır. Bu nedenle çalışma içinde kullanıcı tercihlerinin

nasıl bir kavramsal model ile ifade edilebileceği konusu araştırılmış, incelenen *nicel* ve *nitel* yaklaşımların çalışmaya dahil edilmesi gerektiği görüşüne varılmıştır. Ancak bu sayede *nitel* yaklaşımlara *nicel* sonuçlar ile ulaşılabileceği belirlenmiştir.

Bir kavramsal modelin geliştirilmesi için farklı görsel özelliklere sahip modelleme dilleri kullanılmaktadır. Bu nedenle Bölüm 3 de farklı bakış açıları ile bir kavramsal modelin nasıl tariflenebileceği açıklanmıştır. Bu bölümde EXPRESS veri modelleme dili ile oluşturulan Tercih Yönetim Çerçevesi EK-3 de UML modelleme dili ile de sunulmuştur. UML gibi kabul görmüş MOF tabanlı bir standarda uygun olarak Tercih üst-şemasının geliştirilmesi, oluşturulan kavramsal modelin UML Profillerin sağladığı yetenekler ile birlikte değerlendirilebileceğini de göstermektedir. UML Profilleri kullanılarak farklı üst şemaların genişletildiği alan yazın çalışmasında örneklenmektedir. Aynı bakış açısı Tercih Yönetim Çerçevesi için de geçerli olmakta, bu sayede tez çalışmasında oluşturulan Tercih-üst şemasının herhangi bir ontoloji modelini tariflemek için kullanılan ontoloji üst-şemasını genişletilebileceği fikrine gelinmektedir. Bu fikrin kavramsal model üzerindeki gerçekleştirimi ise Şekil 5.5 de sunulmaktadır.



Şekil 5.5 Tercih Yönetim Çerçevesi ve MDA seviyesi

Şekil 5.5, M3 seviyesinde bir ontoloji modeline ait tanımların yer aldığı üst üst-şema dokümanını, M2 seviyesinde ise ontoloji üst-şemasını ve Tercih üst-şemasına ait tanım bilgilerini bir arada göstermektedir.

Tercih Yönetim Çerçevesi içinde kullanıcı tercihleri, alan bilgisi üzerinde tanımlandığı haliyle genel kabul gören ODM ve ODM de yer alan değer varlıklar ve üst-şema (M2) seviyesinde soyut bir model oluşturacak şekilde Kaynak tanımı altında sunulmaktadır. Kaynak tanımı, Tercih Yönetim Çerçevesi içinde ontolojik varlıkların genel geçer bir ontoloji modelinin parçası olarak <class> ve <property> varlıkları ile temsil edilen yapıya dönüştürülmekte, bu yapı ile de herhangi bir ontoloji geliştirme diline taşınabileceği fikrine dayandırılmaktadır.

Şekil üzerinde M1 seviyesinde bulunan alan ontolojileri ve Tercih ontolojisine ait tüm örnek değerler M0 seviyesinde yer almaktadır. Bu doğrultuda tercihlerin M1 seviyesinde ki Tercih üst-şemasından türediği fikri oluşmaktadır. Ancak bu durum karşısında, genel olarak tariflenen ontoloji modeli (Kaynak varlığı) ile tercih tipleri tanımları için yaratılan Tercih üst-varlığının M2 seviyesinde ilişkilendirilmesi gerekmektedir. Sonuçta, oluşturulan Tercih üst-şemasının ontoloji modelini genişletebilmek adına M2 seviyesinde, ontolojilere ait örnek değerleri model üzerinde tarifleyebilmek adına ise M1 seviyesinde kullandığı gözlenmektedir. Bu durum görsel olarak Şekil 5.5 de sunulmakta, oluşturulan modelin bir ara katman olarak M2 ve M1 seviyeleri arasında yer alması gerektiği fikrine varılmaktadır.

Tercih Yönetim Çerçevesi içinde kullanıcı seçimleri olarak tariflenen tercihlerin değerlerini alan bilgisinden, tanımlarını oluşturulan Tercih üst-şemasından aldıkları kabulü Bölüm 5.1 de belirtilmiştir. Bu kabul ile alan bilgisi içinde yer alan tercih değerlerinin alan bilgisinin dinamik ve değişken yapısından olumlu anlamda etkilendiği ancak üst-şema yaklaşımı ile oluşturulan tercih tanımlarından etkilenmediğidir.

Tercih Yönetim Çerçevesinin *yan modeller* ile desteklenmesi gerektiği yine bu bölümde vurgulanmaktadır. Özellikle Olasık ve Bağlam temelli tercih tanımları için alan yazında kabul görmüş yan modellerin çalışmaya dahil edilmesi gerekmektedir. Ayrıca, tercih niteleyecisi olarak tanımlanan varlığın da yeni bir yan model olarak oluşturulması ve Tercih üst-şemasının

geniřletilmesinde kullanılması geleceęe yönelik bir alıřma olarak deęerlendirilebilir.

Son olarak, oluřturulan Tercih st-řemasının alan bilgisinden baęımsız, genel, modler ve esnek bir yapı sunulmasına tez alıřmasında zen gsterilmiřtir. Bu nedenle,

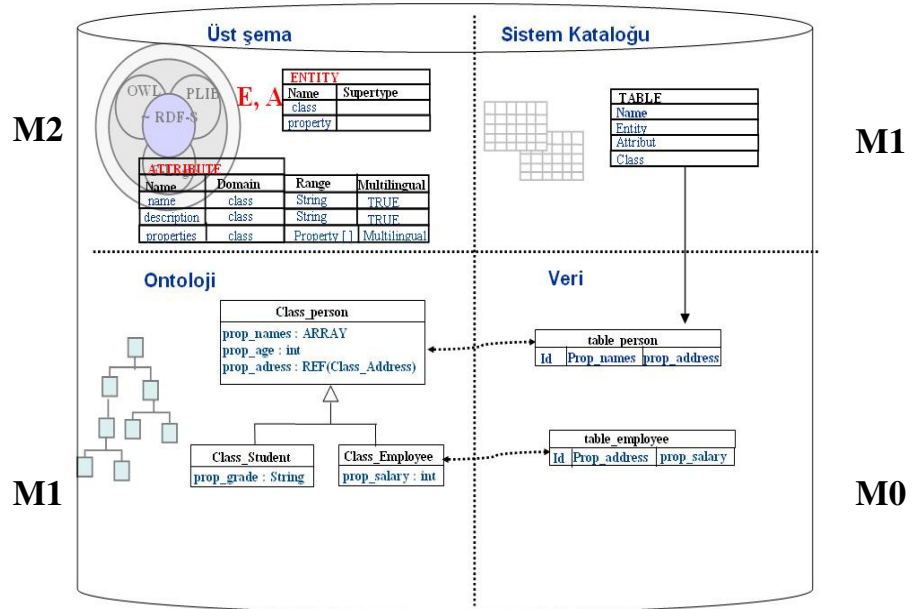
- *Genel* bir model olma zellięi iin farklı tercih tiplerinin ortak bir atı altında toplanması,
- *Modlerlik* zellięi iin farklı ontoloji modellerini destekleyebilecek řekilde alan bilgisinden ayrı olarak tretilmesi,
- *Esneklik* zellięi iin ise Tercih st-řemasının hiyerarřiki bir yapıda sunulması ve yeni bir tercih tipi tanımının řema dokmanına eklenmesi ile saęlanabilmektedir.

Bir sonraki blmde ortaya konan bu yaklařımın seilen ontoloji tabanlı bir mimariye (OntoDB) nasıl aktarıldıęı aıklanmaktadır.

6 ONTODB VE TERCİH MODELİ ENTEGRASYONU

Bir önce ki bölümde kullanıcı tercihlerini sunmak yönünde anlatılan Tercih Yönetim Çerçevesi, bu bölümde OntoDB mimarisine entegre edilerek açıklanmaktadır. Bölüm 4 de anlatıldığı gibi OntoDB mimarisi, verinin Veritabanı tabloları üzerinde ve ontolojik bilgi olarak ontolojiler üzerinde saklanması için gerekli yapıyı bir arada sunmaktadır. Görsel olarak Şekil 6.1 de gösterildiği gibi dört kısımdan oluşan mimari, farklı MDA seviyeleri ile açıklanmaktadır.

MDA seviyeleri (bkz. Bölüm 3), üst-üst-şema (M3), üst-şema (M2), şema (M1) ve örnek değerleri içeren M0 seviyelerinden oluşmaktadır. Tez çalışması ve OntoDB mimarisi için önem kazanan M2, M1 ve M0 seviyeleri bu bölümde Tercih Yönetim Çerçevesinin oluşturulması ve mimariye entegrasyonu sırasında yine kullanılmaktadır.



Şekil 6.1 OntoDB Mimarisi (Pierra et.al., 2005).

Bu nedenle şekilde yer alan M0 seviyesi, OntoDB üzerinde saklanan bilginin yer aldığı *veri* kısmını, M1 seviyesi, veritabanı yönetim tabloları için *sistem kataloğunu*, alan ontolojileri için *ontoloji kısmını*, ve M2 seviyesi ise ontoloji modellerinin saklandığı *üst-şema* kısmını göstermektedir. Mimarinin en

önemli bölümü olarak açıklanan M2 seviyesi üzerinde farklı ontoloji modellerinin desteklenmesi için Entity ve Attribute tanım tabloları bulunmaktadır. Entity tablosu içinde tutulan `<class>` ve `<property>` varlıkları bir ontoloji modeli için gerekli tanım bilgilerini, Attribute tablosu ise Entity tablosu üzerinde saklanan sınıflara ait genel öznitelik tanımlarını içermektedir. Belirtilen genel öznitelik tanımlarının sınıf tanım bilgileri için kullanılıyor olması, bu tablonun genişletilmesi ile mimarinin yeteneklerinin üst-şema seviyesinde artırılması ve ontoloji modeline yeni bir özelliğin kaydedilmesi anlamını taşımaktadır.

Bölüm 3 de açıklandığı gibi OntoQL dili bir tanımlama ve sorgulama dili olarak mimari üzerinde yer almaktadır. Bu nedenle mimari üzerinde gerekli tanımlamaların, tablo güncellemelerinin, üst-şema seviyesinde saklanan tabloların genişletilmesinin gerçekleştirimi ve gerekli sorgulamaların yapılması OntoQL komutları ile sağlanmaktadır (bkz. Ek-3). Aşağıda öncelikle bir önceki bölümde açıklanan Terih üst-şemasının yeni bir kavramsal model olarak mimari üzerinde oluşturulması ve tercih temelli sorgulama gerçekleştirim için Tercih üst-şemasının kullanımı açıklanacaktır.

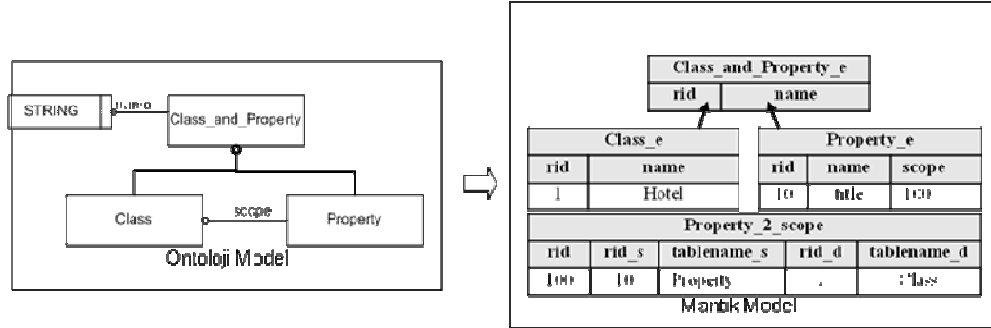
6.1 Tercihlerin OntoDB Üzerinde Modellenmesi

Bölüm 5 de tariflendiği gibi alan bilgisi üzerinde yer alan her örnek değer, kullanıcıların seçimi ile tercihler olarak temsil edilen varlıklara dönüşmektedir. Bu nedenle tez çalışması boyunca kişisel tercihlerin alan bilgisi üzerinden elde edildiği kabulüne uygun olarak aşağıda öncelikle alan ontolojilerinin OntoDB mimarisi içinde nasıl oluşturulduğu açıklanmaktadır.

6.1.1 OntoDB Üzerinde Ontoloji Gösterimi

OntoDB mimarisi üzerinde genel olarak bir ontoloji modeli Şekil 6.2 de sol tarafta gösterildiği gibi bir üst varlık olan `Class_and_Property` varlığı ve sahip olduğu alt varlıkları ile (`Class` ve `Property`) temsil edilmektedir. Bu varlıklar bir ontoloji içinde genel kabul görmüş `Class` ve `Property` tanımlarına karşılık gelen varlıklar olarak tanımlanmaktadırlar (`<name>` özniteliğine sahiptirler). Şeklin sağ tarafında ontoloji modeline

karşılık gelen mantık modelini anlatan tablolar ve tablolar arasında belirtilen ilişkiler gösterilmektedir.



Şekil 6.2 OntoDB için Ontoloji ve Mantık Model Gösterimi.

İlgili modelde name özniteliği ve rid ile gösterilen tanımlayıcılar kullanılmaktadır. Rid, her oluşan örnek değere atanan içsel bir anahtardır. Mantık model içinde yer alan class_e ve property_e tabloları Class_and_Property tablosundan türemektedir. Property_2_scope tablosu class ve property ler arasında ki ilişkiler tutulmakta, rid_d ve rid_s öznitelikleri ise anahtar değerleri saklayan kolonları temsil etmektedir.

Çizelge 6.1 de sunulan örnek ontoloji, OntoDB mimarisinde yer alan concept (Class_and_Property) varlığından türeyecek şekilde class varlığı altında yaratılmaktadır. Descriptor satırı, kullanılacak ikinci bir dil seçeneği için gerekli tarifleyiciyi göstermektedir. Property cümlecisi içinde ise oluşturulan class a ilişkin öznitelik tanımları aldıkları değer tarifleri ile sunulmaktadır. CREATE EXTENT OF satırı, sınıf tanımına (örn: otel sınıfı) ait tüm öznitelikleri bir arada göstermekte ve yeni bir sınıfın var olan sınıf tanımını genişleterek oluşturulmasını sağlamaktadır.

Çizelge 6.1 Otel Ontolojisi

<pre> CREATE #Class Hotel(DESCRIPTOR (#name[fr]='hotel'), #PROPERTY(name STRING, starRate INTEGER, city STRING, price INTEGER); END_ENTITY; </pre>
<pre> CREATE EXTENT OF HOTEL(name,starRate,city); </pre>
<pre> INSERT INTO Hotel(name, starRate,city,price) VALUES('HotelFormule1', 'Paris',2,30); INSERT INTO Hotel(name, starRate,city) VALUES('HotelIBIS', 'Paris',3,85); </pre>

Yaratılan ontolojiye ait çeşitli örnek değerler, kendilerine atanan oid değerleri ile birlikte Çizelge 6.2 de verilmektedir.

Çizelge6.2 Otel ontolojisi Örnek Değerleri

oid	Name	City	starRate	price
23	HotelFormule1	Paris	2	30
28	HoetIBIS	Paris	3	85
29	HotelIBIS	Poitiers	3	65

Bir sonra ki bölümde alan ontolojisinden bağımsız olarak Bölüm 5 de oluşturulan Tercih üst-şemasının OntoDB mimarisi için OntoQL dili ile nasıl yaratıldığı anlatılmaktadır.

6.2 OntoDB Mimarisinin Tercih Modeli ile Genişletilmesi

OntoDB mimarisinin Tercih üst-şeması ile genişletilebilmesi için OntoQL dili içinde tanımlı CREATE komutu kullanılmaktadır (bkz EK-2). Bölüm 5 de anlatılan Tercih üst-şeması kısmı mimari içinde yer alan üst-şema seviyesi içinde oluşturulmaktadır. Tercih üst-şemasının kök varlığı PREFERENCE varlığıdır.

```
CREATE ENTITY #PREFERENCE (
    oid INT,
    ...
);
```

Aşağıda ilk olarak bu varlık ile ilişkilendirilen PREFERENCE_URI varlığı, daha sonra ise modelin sahip olduğu diğer alt varlıkları tanımlanmaktadır.

6.2.1 Preference_URI

Tercih üst-şeması içinde yer alan tercih varlığı için tercih niteleyecisine duyulan gereksinim Bölüm 5 de açıklanmaktadır. Aşağıda bu doğrultuda OntoQL dili ile oluşturulan PREFERENCE_URI varlığı bir örnek ile birlikte oluşturulmaktadır.

Çizelge 6.3 PREFERENCE_URI

```
CREATE ENTITY #Preference_URI (
    #code INT;
    #name STRING,
    #classification STRING);

INSERT INTO #Preference_URI (
    #code, #name, #classification)
VALUES (100, 'cheap', 'price');
INSERT INTO #Preference_URI (
    #code, #name, #classification)
VALUES (50, 'low level', 'quality');
```

Çizelge 6.3 de `CREATE ENTITY` komutu ile yaratılan tercih niteleyici varlığı ve `INSERT INTO` komutu ile atanan örnek değerleri sunulmaktadır. Oluşturulan örnek değerler Çizelge 6.4 gösterildiği gibi kendilerine ait `oid` değeri ile birlikte saklanmaktadır.

Çizelge 6.4 PREFERENCE_URI Değerleri

oid	Code	Name	classification
1315	101	cheap	cost
1316	100	expensive	cost
1319	40	Lux	starRate

Böylece tercih tipleri ile ilişkilendirilecek olan PREFERENCE_URI varlığı için `oid` değeri kullanılmaktadır. İleri ki kısımlarda `oid` değeri örnekleri için URI adresleri de kullanılmaktadır. Bu sayede okunabilirliğin artırılması ve anlamsal bir yapının örneklendirilmesi hedeflenmektedir.

6.2.2 Numeric_Preference

Bölüm 5 Kısım 2 de açıklanan Numeric_Preference tercih tipi model üzerinde Interpreted_Preference varlığı altında, *number_value* özniteliği ile tanımlanmaktadır. Bu tercih tipine ilişkin OntoQL kodu Çizelge 6.4 de sunulmaktadır.

Çizelge 6.5 Numeric_Preference

```
CREATE ENTITY #Numeric_Preference(
  UNDER #Interpreted_Preference(
    #number_value INT,
    #URI #REF(Preference_URI))
);

INSERT INTO #Numeric_Preference(
  #number_value, #URI)
VALUES (2, 'http://...');
```

Oluşturulan örnekte *ucuz* tercih niteleyicisi 2 sayısı ile ilişkilendirilmektedir.

6.2.3 Boolean_Preference

Bölüm 5 Kısım 2 de açıklanan Boolean_Preference tercih tipi yine Interpreted_Preference varlığı altında Çizelge 6.4 de ki gibi tanımlanmaktadır.

Çizelge 6.6 Boolean_Preference

```
CREATE ENTITY #Boolean_Preference(
  UNDER #Interpreted_Preference(
    #properties (Boolean ARRAY),
    #URI #REF(Preference_URI))
);

INSERT INTO #Boolean_Preference(
  #properties, #URI)
VALUES (ARRAY[('casino', yes),
  ('golf', yes)], 'http://...');
```

Sahip olduğu URI özniteliği ile ilgili tercih niteleyicisini, properties özniteliği ile ise Boolean bir değişken olarak tanımlanmış öznitelikleri bir dizi ile temsil edecek şekilde tutmaktadır.

Verilen örnek seçilen öznitelikler ile uygun bir tercih niteleyicisinin nasıl eşleştirildiğini göstermektedir.

6.2.4 Interval_Preference

Bölüm 5 Kısım 2 de açıklanan Interval_Preference tercih tipi Interpreted_Preference varlığı altında min_value, max_value öznitelikleri ve URI tercih niteleyecisi ile Çizelge 6.5 de ki gibi tanımlanmaktadır. Oluşturulan tercih tipi içine atanan değerler için ise Insert komut satırı kullanılmaktadır.

Çizelge 6.7 Interval_Preference

```

CREATE ENTITY #Interval_Preference(
  UNDER #Interpreted_Preference(
    #min_value INT,
    #max_value INT,
    #URI #REF(Preference_URI))
);

INSERT INTO #Interval_Preference(
  #min_value, #max_value, #URI)
VALUES(90, 100), 'http://...';

```

Verilen örnek aralık tercih tipi için seçilen uygun aralık değerlerini, atanan tercih niteleyici ile bir arada sunulmaktadır.

6.2.5 Fuzzy_Preference

Bölüm 5 Kısım 2 de açıklanan Fuzzy_Preference varlığı Interpreted_Preference varlığı altında prob_value, properties ve URI öznitelikleri ile Çizelge 6.6 de ki gibi tanımlanmaktadır.

Çizelge 6.8 Fuzzy_Preference

```

CREATE ENTITY #Fuzzy_Preference(
  UNDER #Interpreted_Preference(
    #prob_value FLOAT,
    #properties REF (BOOLEAN ARRAY),
    #URI #REF(Preference_URI))
);

INSERT INTO #Fuzzy_Preference(
  #prob_value, #properties, #URI)
VALUES(0.95, ARRAY [ ('casino',yes),
  ('golf', yes), ('jakuzi',yes)],
'http://...');

```

Olasılık tercih tipi örneği için ilgili tercih niteleyici ve öznitelik değeri, atanan bir olasılık değeri ile buluşmaktadır.

6.2.6 Enumerated_Preference

Bölüm 5 Kısım 2 de açıklanan Enumerated_Preference tercih tipi Interpreted_Preference varlığı altında pref_value ve URI öznitelikleri ile Çizelge 6.6 de ki gibi tanımlanmaktadır.

Çizelge 6.9 Enumerated_Preference

```
CREATE ENTITY
#Enumerated_Preference(
  UNDER #Interpreted_Preference(
    #pref_value REF (STRING ARRAY),
    #URI #REF(Preference_URI))
);

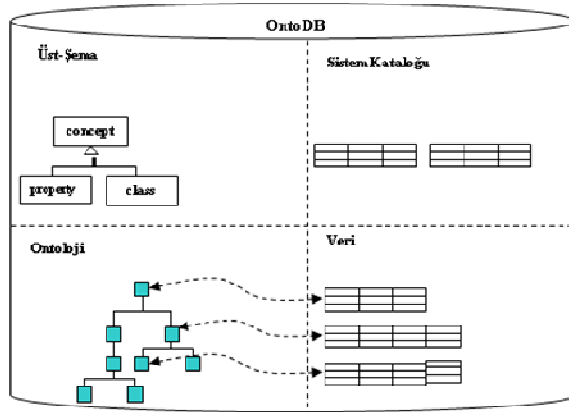
INSERT INTO #Enumerated_Preference(
  #pref_value, #URI)
VALUES (ARRAY[ 'HotelFormule1',
  'HotelIBIS'], 'http://...');
```

Listelenmiş tercihlere atanacak değerler için gerekli tercih niteleyecisi, sınıf ya da öznitelik örnekleri ile birlikte saklamaktadır.

Bu adımlar sonrasında OntoDB mimarisi üzerinde yaratılan ve örnek değerleri ile saklanan tercih tipleri için Kaynak ve Tercih üst-şeması entegrasyonu gerçekleştiriminin yapılması gerekmektedir. Bir sonra ki kısımda OntoDB mimarisi üzerinde gerekli işlem adımları anlatılmaktadır.

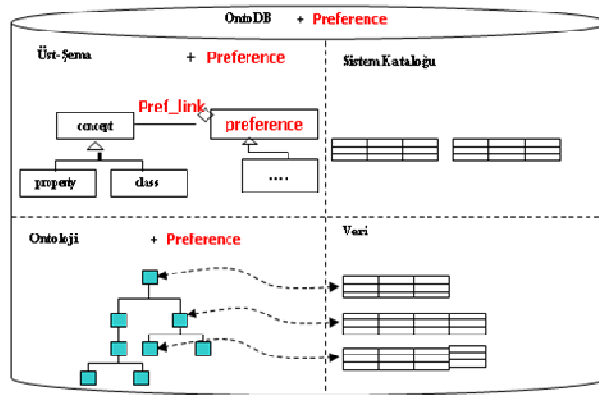
6.3 OntoDB Mimarisi İçin Tercih - Ontoloji Bağı

Mimari üzerinde alan ontolojisi ve Tercih üst-şeması oluşturulduktan sonra Tercih-Ontoloji bağının kurulması gerekmektedir. Bu sayede alan bilgisi olarak belirtilen ontolojik bilgi üzerinden tercihlerin elde edilebileceği gösterilebilecektir. Aşağıda sunulan Şekil 6.3 de OntoDB mimarisi üst-şema ve şema kısımları ile açıklanmaktadır.



Şekil 6.3 Genişletme öncesi OntoDB

Tercih üst-şemasının gerçekleştirimi sonrası mimari üzerinde yer alan yapısal genişleme Şekil 6.3 ü genişletilecek şekilde Şekil 6.4 de sunulmuştur.



Şekil 6.4 Tercih Üst-Şeması ile Genişletme sonrası OntoDB Mimarisi.

Böylece kavramsal bir model olarak oluşturulan Tercih üst-şemasının mimari üzerinde yer alan ontoloji modeline entegre edildiği gösterilmekte, üst-şemanın tarifinde kullanılan concept kavramına Pref_Link özelliğinin eklendiği belirtilmektedir.

Pref_Link OntoDB mimarisi üzerinde yer alan üst-şema kısmına eklenecek yeni bir özelliktir. ALTER komutu (OntoQL dili komutu) kullanılarak

gerçekleştirimi sağlanmaktadır. Açıklayıcı sözdizimi EK-2 de sunulmuştur. Belirtilen sözdizimine uygun komut satırı ile Pref_Link özelliğinin oluşturulması aşağıda gösterilmektedir.

```
ALTER ENTITY #CONCEPT ADD ATTRIBUTE #PREF_LINK
REF (#PREFERENCE) ARRAY;
```

Bu sayede Pref_Link özelliği ile alan ontolojiler için yeni bir üst özellik tanımlama alanı üst-şema seviyesinde oluşturulmuştur.

UPDATE komutu ise Pref_Link özelliğine bir değer kümesi set edebilmek için kullanılmaktadır.

```
UPDATE #CLASS SET #PREF_LINK ARRAY [ 'cheap' ,
'standard' , 'full board' ] where name='HOTEL' ;
```

Alan ontolojisi (otel ontolojisi) ile Tercih üst-şemasının entegre edilmesine ilişkin kullanılan komut satırı yukarıda gösterilmektedir. Bu komut ile class varlığından türeyen her hangi bir sınıfa (örn: Hotel) Pref_Link özelliğinin eklenmesi ile Pref_URI tercih niteleyecisine ait değerler (id) verilmektedir.

Ancak, yukarıda örneklenen UPDATE komutu içinde id değerleri yerine okunabilirliği arttırmak için PREFERENCE_URI tercih niteleyicisinin name öznitelikleri ile belirtilen tercih niteleyici isimleri kullanılmıştır.

Bir sonraki kısımda ise tercih üst-şeması ve ontoloji üst-şeması entegrasyonu sonrası OntoDB üzerinde gerçekleştirilecek olan tercih temelli sorgulama anlatılmaktadır.

6.4 Tercihlerin OntoDB Üzerinde Sorgulanması

Tıpkı SQL ve SPARQL sorgulama dillerinde olduğu gibi tercih temelli sorgulama gerçekleştirimi OntoQL dili içinde önem kazanmaktadır. Bu doğrultuda tez çalışması boyunca yapılması öngörülen tercih temelli sorgulama gerçekleştirimi alan yazın bölümünde incelenen sorgulama çeşitlerinde olduğu gibi OntoQL dili için ele alınmakta, yeni bir sorgu operatörü tanımı oluşturulması yönünde çalışılmaktadır. Tercih Yönetim Çerçevesi adı altında geliştirilen kavramsal modelin kullanıldığı bir yaklaşım ile tercihlerin anlamsal seviyede nasıl sorgulanabileceği, OntoDB mimarisine entegre edilen Tercih üst-şeması ve Pref_Link özniteliğini kullanan PREFERRING operatörünün nasıl oluşturulduğu, OntoQL sorgu motoru ile birlikte aşağıda açıklanmaktadır.

6.4.1 Preferring Operatorü Sözdizimi

Bu bölümde OntoQL diline ait tercih tabanlı sorgulama gerçekleştirim adımları anlatılmaktadır. Bölüm 2 de yer alan ve alan yazın taramasında belirtilen tercih temelli sorgulama gerçekleştirim adımlarına benzer bir yaklaşım ile OntoQL dili için gerekli söz dizimleri açıklanmakta daha sonra bu söz diziminin tercihlerin sorgulanabilmesi yönünde nasıl genişletildiği anlatılmaktadır.

OntoQL dili Bölüm 4 ve Ek-2 açıklandığı gibi SQL yetenekleri kullanılarak oluşturulmuştur. OntoDB mimarisi üzerinde geliştirilen bir sorgu motoru üzerinde çalışmaktadır. Bu dil üzerinde kısıt değerleri SQL dilinde olduğu gibi <hard constraint> olarak tariflenmekte ve where, having, group by komutları kullanımı ile sorgulanmaktadır. Aşağıda basit bir OntoQL cümleciğinin içinde yer alan komutların söz dizimleri gösterilmektedir.

```
<query specification> ::= <select clause>
<from clause> [ <where clause> ]
[ <group by clause> ] [ <having clause> ] [ <order by clause> ]
[ <namespace clause> ] [ <language clause> ]
```

Ancak kullanıcı tercihlerinin sorgulanması yönünde yetersiz kalan OntoQL sorgulama dili için tez çalışmasında sunulan çözüm, SPARQL dilinde olduğu gibi tercihlerin birer `<soft constraint>` olarak ele alınması yönündedir. Bu nedenle PREFERRING komutunun geliştirilmesi, bu komutun Tercih Yönetim Çerçevesi ile entegre edilmesi, gerekli kısıtların (`<soft constraintler>`) tercih üst-şemasında yer alan tanımlamaları cinsinden ifade edilmesi ve kullanıcıların sorgularında bu kısıtları kullanmaları hedeflenmektedir. Bu sayede *nitel* olarak tariflenen tercihlerin *nicel* yaklaşımlarda olduğu gibi sorgulanabilecekleri gösterilmektedir. Aşağıda bu bakış açısıyla oluşturulan söz dizimi sunulmaktadır.

```

<query specification> ::= <select clause>
<from clause> [ <where clause> ]
[ <group by clause> ] [ <having clause> ]
[ <order by clause> ] [preferring clause]
[ <namespace clause> ] [ <language clause> ]
<preferring clause> ::= <boolean preferenceIdentifier expression>
<boolean preferenceIdentifier expression> ::= <boolean term>
| <boolean value expression> OR <boolean term>
<boolean term> ::= <boolean factor>
| <boolean term> AND <boolean factor>
<boolean factor> ::= [ NOT ]preferenceIdentifier
<preferenceIdentifier> ::= <identifier>

```

Genişletilen OntoQL sorgulama söz dizimi içinde tanımlanan PREFERRING operatörü bir `<preferenceidentifier>` değerini kısıt değeri olarak almaktadır. Bu değer oluşturulan kavramsal model üzerinde Preference_URI ile tariflenen tercih niteleyicisine karşılık gelmektedir. Bu sayede kullanıcıların günlük yaşantıda tercihlerini açıklamakta kullandıkları `<soft constraint>` tercih niteleyici yan kavramları tercih üst-şeması içinde yer alan uygun tercih tipleri ile buluşmakta, alan ontolojisi içinde tanımlanan ilgili örnek değerleri cinsinden sorgulanabilmektedir. Böylece tercih üst-şemasında yer alan tercih niteleyicilerin birer `<preferenceidentifier>` olarak sorgu motorunun içinde kullanılabilmesi sağlanmaktadır.

Bölüm 5.2 de açıklanan ontoloji tercih bağı kısmında hareketle tez çalışmasında hedeflenen tercih ontoloji entegrasyonu, tercih temelli sorgulama gerçekleştirimi içinde kullanılmaktadır. Tercih niteleyeciler üzerinden tariflenen tercih tanımları bu sayede alan ontolojisi üzerinde yer alan ontolojik örnekler olarak kişisel tercihlerin sorgulanması sırasında kullanılmaktadır.

Bir sonra ki kısımda söz dizimsel olarak ifade edilen tercih operatörünün örnek sorgular içinde nasıl kullanıldığı açıklanmaktadır.

6.4.2 Sorgu Yorumlama

Yukarıda açıklandığı şekli ile PREFERRED operatörü kullanıcı tercihlerini birer kısıt değeri olarak almaktadır. SPARQL sorgu dilinde olduğu gibi <softconstraint> olarak belirlenen tercihler PREFERRED operatörü içinde Tercih üst-şeması ile entegre edilmekte, üst-şemada tanımlanan tercih tipleri ile birlikte örtülü olarak kullanılmaktadır. Aşağıda yer alan sorgu cümleciği yapısı ilgili operatör kullanımını açıklamaktadır.

```
SELECT 'selection'
FROM 'tableReference'
PREFERRED 'preferenceIdentifier'
```

Oluşturulan sorgunun ilişkili tablo üzerinden bir tercih niteleyici isim ile yapılabileceği gösterilmekte, bu sayede kullanıcının tercih tarifi (*ucuz*, *yakın*, *lüks*) sorgulanabilmektedir. Belirtilen tercihler üst-şema üzerinde yer alan yorumlanabilir tercihler ile ilişkilendirilmekte, böylece tanımlandıkları fonksiyonlara atanan uygun değerler ile cevap bulmaktadır. Sonuç olarak kullanıcılara alan bilgisi üzerinde yer alan örnek değerlerin, kişisel tercihleri cinsinden döndürülmesi sağlanmaktadır (*ucuz otel*, *yakın otel*, *lüks otel*).

Tercih operatörü söz dizimi içinde boolean operatör kullanımına yer verilmektedir. Bu sayede birden çok tercihin bir arada sorgulanabilmesinin sağlanabilmesi yönünde ilk adım atılmıştır.

```
SELECT 'selection'
FROM 'tableReference'
PREFERRING 'preferenceIdentifer' AND 'preferenceIdentifer'
```

```
SELECT 'selection'
FROM 'tableReference'
PREFERRING 'preferenceIdentifer' OR 'preferenceIdentifer'
```

```
SELECT 'selection'
FROM 'tableReference'
PREFERRING NOT 'preferenceIdentifer' AND 'preferenceIdentifi
```

Ancak alan yazında anlatıldığı gibi tercihlerin önceliklendirilmesi ya da ağırlık değerleri ile gösterilerek çoklu tercihlerin sorgulanması, SQL dilinde yer alan *Pareto* ve *Priority* tercih sorgulayıcılarının OntoQL dili içine entegre edilmesi ile sağlanabilmektedir. Tez çalışması içinde bu kısım gerçekleştirilememiştir. Ayrıca tercihlerin basit ve karmaşık tercihler olarak ele alındığı durumlar Bölüm 2 de sunulmaktadır. Basit tercihler CASCADE komutu kullanılarak da bir araya getirilebilmektedir. Tespit edilen bu durum karmaşık tercihlerin farklı tercih niteleyiciler ile birlikte sorgulanabileceğini göstermektedir. Ancak bu bölümde tercih temelli sorgulama gerçekleştirimi sırasında CASCADE komutu kullanımı ele alınmamakta, bu kapsamında çözüm önerisi sunulmamaktadır

6.5 Değerlendirme

Bu bölümde kavramsal model olarak geliştirilen Tercih üst-şemasının OntoDB mimarisine entegre edilmesi anlatılmaktadır. OntoDB mimarisi içinde yer alan üst-şema kısmı Anlamsal Web çalışmalarına örnek teşkil eden yapıyı sunmaktadır. Bu yapı içinde ontoloji tanımlamak için gereken ontolojik üst-model bulunmaktadır. Belirtilen üst-model üzerinde tanımlanan concept varlığı, Tercih Yönetim Çerçevesi içinde yer alan Kaynak tanımı ile eşleşmektedir. Bu sayede kullanılacak ontoloji dili ne olursa olsun ontoloji tanımlamak için kullanılması gereken temel class ve property varlıklarına üst şema seviyesinde ulaşılmaktadır. Aynı yapı Bölüm 5 de açıklanan Tercih üst-şemasının tanımlanacağı kısım olarak da görev yapmaktadır. Bu seviyede mimari üzerinde yeni bir modelin oluşturulması, var olan yapının genişletilmesi OntoQL dili yetenekleri ile sağlanmaktadır, çünkü

OntoQL dili aynı zamanda mimari içinde ontoloji tanımlama ve sorgulama dili olarak kullanılmaktadır.

OntoDB mimarisi içinde yer alan üst-şema kısmının genişletilmesi sonrası modelin geçerliliğinin kontrolü, tercih temelli sorgulama gerçekleştirimi ile sağlanmaktadır. Bu nedenle Bölüm 2 de yer alan alan yazın çalışmasında SQL ve SPARQL dillerinde oluşturulan yaklaşımın bir benzeri üzerinde çalışılmış, tercihlerin `<softcondition>` olarak tanımlanan kısıt değerleri ile ifade edildiği fikri kullanılmıştır. Ancak diğer çalışmalardan farklı olarak tek bir tercih tipi tanımı için değil, oluşturulan Tercih üst-şemasında yer alan tüm tercih tipleri için sorgulama gerçekleştirimi yapılabileceği gösterilmiştir. Bu nedenle öncelikle Tercih-üst şeması içinde yer alan Tercih üst-varlığı ile ontoloji modelini temsil eden `concept` varlıkları arasında `Pref_Link` bağı oluşturulmuş ve bu bağı bir özellik olarak OntoDB üzerinde yer alan özellik tablosuna eklenmesi sağlanmıştır. Bundan sonra ki adımda ise `<softcondition>` olarak tariflenen kullanıcı tercihlerinin bir `<preferenceidentifier>` olarak sorgu cümleleri içinde kullanılması sağlanmıştır. Bu sayede kullanıcıların bir tercih niteleyici ile tercihlerini tariflemeye alan bilgisinden bağımsız yan kavramaları kullanmaları durumu tercih temelli sorgulama gerçekleştirimi içinde örneklendirilebilmiştir. Ancak literatürde örnekleri bulunan karmaşık tercih tanımlarının oluşturulması ve örneklendirilmesine yönelik olarak bir çalışma tez çalışması kapsamında gerçekleştirilmemiştir. Oluşturulan tercih operatörü ile yapılabilecek sorgulamalarda birden çok tercih niteleyicisinin bir araya getirilebileceği durumlar, *skyline* ve *top(k)* gibi tercih temelli sorgulama gerçekleştiriminde kullanılan önemli algoritmalar çalışmaya dahil edilmemektedir. Bölüm 2 de açıklanan *Pareto* ve *Priority* algoritmalarının çalışmaya sağlayacağı katkılar bilinmektedir. Bu sayede birden çok tercih niteleyicisinin önem sırası ve önceliği belirtilerek sorgulanabilecek olması çoklu ve karmaşık tercihlerin sorgulanmasında avantaj sağlayacaktır. Göz ardı edilen bu durum gelecek çalışma bölümünde belirtilmekte, OntoQL dili için tercih temelli sorgulama gerçekleştiriminin iyileştirilmesine yönelik yeni bir hedef olarak gösterilmektedir. Bu bölümde mimari üzerinde ortaya konan model ve tercih temelli sorgulama gerçekleştirimi için oluşturulan `PREFERRING` operatörünün kullanım örnekleri bir sonraki bölümde oluşturulan bir senaryo içinde tariflenen durumlar ile ele alınmaktadır.

7 UYGULAMA GERÇEKLEŞTİRİMİ

Bu bölümde Tercih Yönetim Çerçevesi için geliştirilen iki uygulama sunulmaktadır. Uygulamalar,

- kişisel tercihlerin Tercih Yönetim Çerçevesi elemanları kullanılarak alan bilgisi üzerinden elde edilmesini,
- OntoDB mimarisi üzerine entegre edilen Tercih Yönetim Çerçevesini ve tercih temelli sorgulama gerçekleştirimi için tanımlanan *tercih operatörünün* kullanımını açıklamaktadır.

Çalışma boyunca Turizm alanı üzerinde örneklenen kişisel tercihler bu bölümde yine benzer bir yaklaşım ile aşağıda yer alan senaryoda sunulduğu gibi kullanılmaktadır.

Örnek Senaryo:

Enis bir akademisyendir, ailesi ile geçireceği yaz tatili için o yıl Fransa’ da katıldığı konferans sırasında konakladığı oteli seçmek istemektedir. Ancak kişisel tercihleri otel fiyatı için ‘ucuz’, oda konforu için ‘standart’ odadır. Tercihlerini sorgulamak üzere kullanıcısı olduğu online tatil rezervasyon sistemine bağlanır. Kişisel tercihlerini bulma yönünde sorgulama yapmaya başlar.

Senaryo içinde tariflenen sistem, anlaşmalı olduğu otellerin yönetimini yapan bir portal olarak düşünülmektedir. Portal üzerinde sunulan otellerin genel hizmet ve kalite standartları 1 den 6 ya kadar kullanılan yıldız sayıları ile temsil edilmektedir. Bu nedenle uygulama için geliştirilen otel ontolojisi turizm alanına yönelik gerekli kavramlar uygun değerleri ile saklanmaktadır.

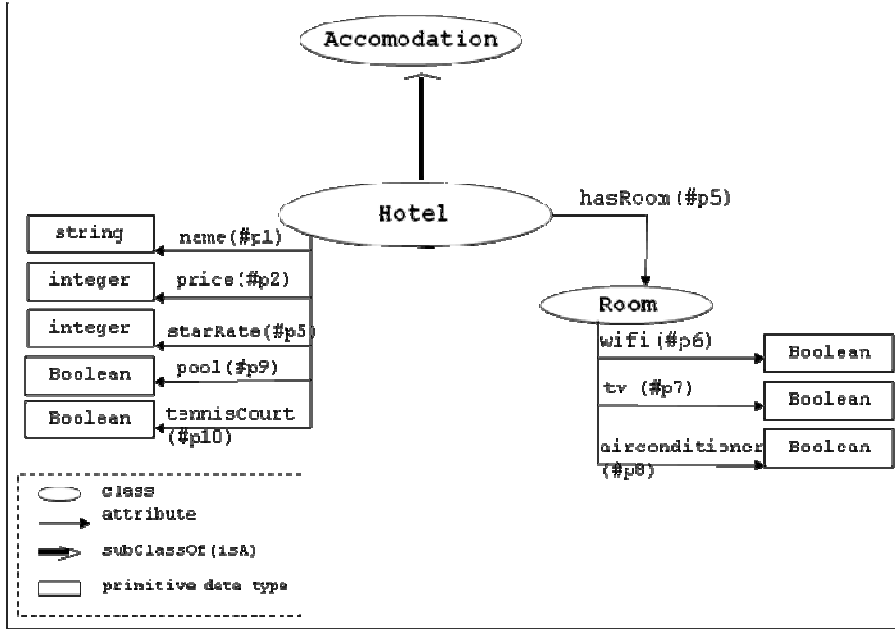
7.1 Örnek-1: Kişisel Tercihlerin Alan Ontolojisi Üzerinden Elde Edilmesi

Bu uygulamada kullanıcıların seçimlerine cevap verebilecek örnek değerlerin bulunması yönünde tercihlerin oluşturulmasını anlatmaktadır. Bu nedenle önce tercih uzayı olarak kabul edilen alan ontolojisinin yaratılması, daha sonra senaryo içinde örneklenen tercih niteleyicilerin (*'ucuz'*, *'standart'* gibi) tercih üst-şeması içinde yer alan tercih tipleri ile birlikte tanımlanması ve son olarak uygun tercih tipleri ile alan ontolojisi örnekleri arasında kurulması gereken bağın oluşturulması gerekmektedir. Hedef herhangi bir sayısal değer kullanılmadan belirtilen tercih niteleyicilere ontolojik bilgiler cinsinden ulaşılmasıdır. Bu nedenle aşağıda öncelikle senaryo gereği Turizm alanı için gerekli tanımları içeren otel ontoloji oluşturulmaktadır. Belirtilen ontolojinin geliştirilmesinde Bölüm 3 de açıklanan ve Ek-1 de sunulan EXPRESS dili kullanılmaktadır.

7.1.1 Alan Ontolojisi Geliştirme

Turizm alanı için oluşturulacak Otel ontolojisi konaklama ve oda bilgileri (tv, wifi, ...) ile bir otelin sunduğu servisleri (golf, yüzme havuzu, casino, ...) tanımlama yönünde gerekli bilgileri içermektedir. Şekil 7.1 de gösterildiği gibi örnek bir ontoloji sınıf tanımları, sınıf tanımına ait öznitelik özellikleri ve farklı sınıfları birbirleriyle ilişkilendiren nesne özelliklerinden (#p) oluşmaktadır. Belirtilen tüm özellikler (*öznitelik* ve *nesne*) ve sınıf tanımları için örnek değerler oluşturulan ontoloji üzerinde yaratılmaktadır. Bu değerler gerekli tercih tariflerinin ifadesinde kullanılan değerler olarak ileride kullanılacaktır.

Bölüm-5 Kısım 5.1.2 de açıklanan Kaynak tanımı, Tercih Yönetim Çerçevesinin ilk elemanı olarak sunulmaktadır. Bu nedenle Otel ontolojisi içinde sunulan tüm temel öznitelik değerleri, Kaynak tanımına ait *property_value* ve *class_value* değerlerine atanacak şekilde oluşturulmaktadır.



Şekil 7.1 Otel Ontolojisi.

Class_value ve property_value değerlerine ilişkin örnekler;

```
#50 =CLASS_VALUE ('HotelFormule1');
#p6=PROPERTY_VALUE ('wifi', yes/no);
```

olarak Çizelge 7.1 de Express modelleme dili kullanılarak oluşturulmuştur. Farklı bir gösterim ile Şekil 7.2 de sunulan otel ontolojisine ait örnekler, tercih üst-şeması ve alan ontolojisi entegrasyonu için kurulan tercih bağı gösteriminde yine kullanılacaktır.

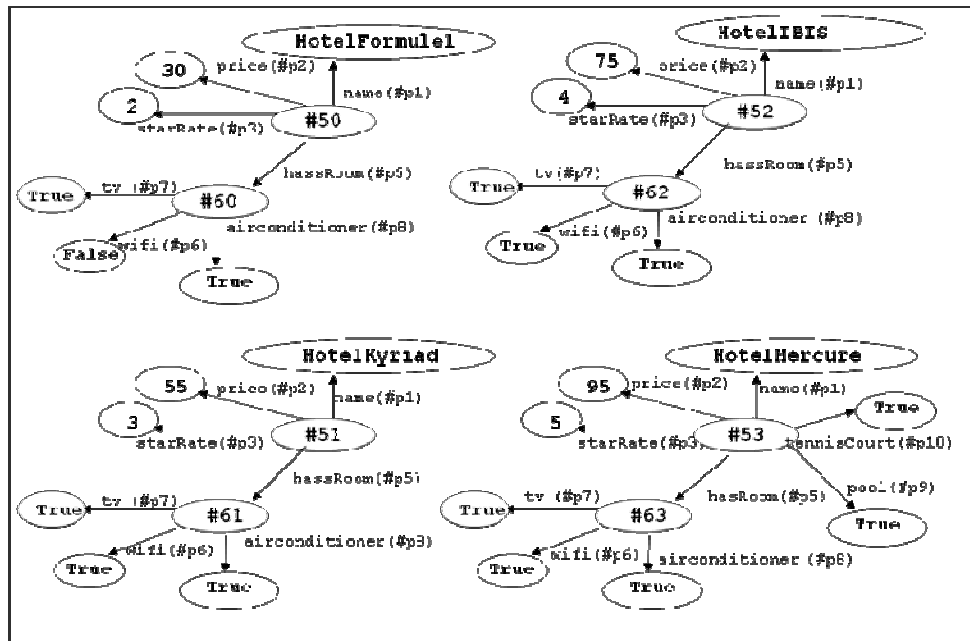
Çizelge 7.1 Ontolojik Örnekler.

```

#id=name(tv,wifi,tenniscourt,pool,airconditioner,
starRate,price)

#50=HotelFormule1(('tv',yes),('wifi',no),('tennis
court',no),('airconditioner',yes),('pool',no),('s
tarRate',2),('price',30))
;
#51=HotelKyriad(('tv',yes),('wifi',yes),
('tenniscourt',no),('airconditioner',yes),('pool'
,no),('starRate',3),('price',55))
;
#52=HotelIBIS(('tv',yes),('wifi',no),
('tenniscourt',no),('airconditioner',yes),('pool'
,no),('starRate',4),('price',75))
;
#53=HotelMercure(('tv',yes),('wifi',yes),
('tenniscourt',yes),('airconditioner',yes),('pool'
',yes),('starRate',5),('price',95));
#54=HotelIBIS(('tv',yes),('wifi',yes),
('tenniscourt',no),('airconditioner',yes),('pool'
,yes),('starRate',3),('price',55))
;
#55=HotelIBIS(('tv',yes),('wifi',yes),
('tenniscourt',no),('airconditioner',yes),('pool'
,no),('starRate',3),('price',45))
;

```



Şekil 7.2 Otel Ontolojisi Örnekleri.

Bu nedenle bir sonraki bölümde kavramsal bir model olarak sunulan Tercih üst-şeması uygun örnek değerleri ile nasıl oluşturulduğu açıklanmaktadır.

7.1.2 Tercih Üst-Şeması Örnekleri

Bölüm 5 de açıklanan Tercih Yönetim Çerçevesinin oluşturulmasında kullanılan varlıklar için yukarıda sunulan senaryoya uygun olarak oluşturulan otel ontolojisi örnek değerleri kullanılmaktadır.

Tercih üst-şeması içinde yer alan tercih tipleri, örnek değerleri ile sunulmaktadır. Bölüm 5 Kısım 5.2 de açıklanan tercih niteleyiciler kişisel tercihlerin ifadesinde kullanılmakta, tercih üst şeması içinde yer alan her tercih tipi için ele alınmaktadır. Aşağıda öncelikle senaryo gereği ihtiyaç duyulan *fiyat* ve *kalite* kategorileri için belirlenen tercih niteleyici örnekleri verilmektedir.

Tercih Niteleyici Örnekleri: Senaryoda tariflenen *ucuz (cheap)* ve *standart (standard)* tercih niteleyicilerinin yaratılması için seçilen *fiyat (price)*, *yıldız (starRate)* ve *kalite (quality)* kategorileri için farklı örnek tercih niteleyicileri Çizelge 7.2 de ilgili EXPRESS kod parçacığı ile birlikte sunulmaktadır. Üç farklı kategori için oluşturulan,

- #1, #2 #3 ve #4 numaralı örnekler; 100, 101, 102, 103 kod numaraları ile tariflenen *ücret (cost)* kategorisi altında belirtilebilecek tercihleri *çok ucuz (very cheap)*, *ucuz (cheap)*, *pahalı (expensive)* ve *çok pahalı (very expensive)* tercih niteleyici isimleri ile,
- #7, #8, #9, #10, #11 numaralı örnekler; 50, 51, 52, 53 ve 54 kod numaralı tercih niteleyicileri *starRate* kategorisi altında *düşük (low)*, *standart (standard)*, *orta (middle)*, *lüks (lux)* ve *çok lüks (very lux)* olarak belirlenen tercih niteleyici isimleri ile,
- #27, #28, #29, #30, #31 numaralı örnekler; 150, 151, 152, 153 ve 154 kod numaralı tercih niteleyicileri *kalite (quality)* kategorisi altında *düşük (low)*, *standart (standard)*, *orta (middle)*, *lüks (lux)*

ve *çok lüks* (very lux) olarak belirlenen tercih niteleyici isimleri ile belirtmektedir.

Çizelge 7.2 Preference_URI Örnekleri

<u>EXPRESS Sözdizimi</u>
<pre> ENTITY PREFERENCE_URI code: INTEGER; name: STRING; classification STRING; END_ENTITY; </pre>
<u>Örnek</u>
<pre> #1=Preference_URI(100,'very cheap','cost'); #2=Preference_URI(101,'cheap','cost'); #3=Preference_URI(102,'expensive','cost'); #4= Preference_URI(103,'very expensive','cost'); #7=Preference_URI(50,'low','starRating'); #8=Preference_URI(51,'standard','starRating'); #9=Preference_URI(52,'middle','starRating'); #10=Preference_URI(53,'lux','starRating'); #11=Preference_URI(54,'very lux','starRating'); #27=Preference_URI(150,'low','quality'); #28=Preference_URI(151,'standard','quality'); #29=Preference_URI(152,'middle','quality'); #30=Preference_URI(153,'lux','quality'); #31=Preference_URI(153,'very lux','quality'); </pre>

Yukarıda oluşturulan tercih niteleyicilerin farklı tercih tipi tanımları ile nasıl kullanıldıkları ise aşağıda gösterilmektedir.

Listelenmiş Tercihler: Bölüm 5 de açıklandığı gibi bir popülasyonunun gösterilmesi için oluşturulan tercih tipi olarak açıklanmaktadır. Çizelge 7.3 de EXPRESS sözdizimi için örnekleri yer almaktadır.

Çizelge 7.3 Enumerated_Preference Örnekleri

<u>EXPRESS Sözdizimi</u> <pre> ENTITY ENUMERATED_PREFERENCE SUBTYPE OF (INTERPRETED_PREFERENCE); pref_values: LIST [1:?] PROPERTY_OR_CLASS_INSTANCE; pref_attributes: LIST[1:?] OF PREFERENCE_URI; END_ENTITY; </pre>
<u>Örnek Değerler</u> <pre> #28=Preference_URI (151,'standard','quality'); #200=Enumerated_Preference ([('HotelIBIS'), CLASS_VALUE ('HotelIBIS')], (#28)); </pre>

‘Standard IBIS otellerini’ tarifleyen #200 numaralı örnek, Çizelge 7.1 de yer alan *standard* tercih niteleyicisinin PROPERTY_OR_CLASS_INSTANCE özniteliğine atanması ile oluşturulmuştur. Liste içinde otel ontolojisine ait uygun class_value örnek değerleri ile buluşturulması hedeflenmektedir.

Sayısal Tercihler: Bölüm 5 de açıklanan tercih tipi için gerekli EXPRESS söz dizimi ve örneği Çizelge 7.4 de verilmektedir.

Çizelge 7.4 Numeric_Preference Örnekleri

<u>EXPRESS Sözdizimi</u>
<pre> TYPE NUMBER_VALUE= NUMBER; END_TYPE; ENTITY NUMERIC_PREFERENCE SUBTYPE OF (INTERPRETED_PREFERENCE); interpreted_by:LIST[1:?] OF NUMBER_VALUE; pref_attributes: LIST[1:?] OF PREFERENCE_URI; END_ENTITY; </pre>
<u>Örnek Değerler</u>
<pre> #8=Preference_URI (51,'standard','starRating'); #9=Preference_URI (52,'middle','starRating'); #10=Preference_URI (53,'lux','starRating'); #77= ('starRate', 3); #78= ('starRate', 4); #79= ('starRate', 5); #80= ('starRate', 6); #207=Numeric_Preference (#77,#8); #208=Numeric_Preference (#78,#9); #209=Numeric_Preference ([#79,#80],#10); </pre>

‘Lüks otelleri’ tarifleyen #209 numaralı örnek; Çizelge 7.1 de gösterilen Otel ontolojisi içinde yer alan örnek otellerde yer alan *yıldız* (starRate) öznitelik örnek değerlerini (5 ve 6) *lüks*, (4) *orta seviyeli*, (3) *standard* tercih niteleyicileri ile buluşturmaktadır.

Aralık Tercihleri: Alt ve üst kısıt değerlerine bağlı olarak bir aralık değeri uygun ile tariflenen aralık tercihleri Çizelge 7.5 de gösterildiği gibi ilgili EXPRESS sözdizimi için örneklendirilmektedir.

‘Ucuz otelleri’ tarifleyen #251 numaralı örnek; uygun aralık değerlerini [45 60] *ücret* (cost) kategorisi ile sunulan tercih niteleyici ile ilişkilendirmektedir.

Çizelge 7.5 Interval_Preference Örnekleri

<u>EXPRESS Sözdizimi</u> <pre> ENTITY INTERVAL_VALUE min_value: REAL; max_value: REAL; WHERE min_value<max_value; END_ENTITY; ENTITY INTERVAL_PREFERENCE SUBTYPE OF (INTERPRETED_PREFERENCE); interpreted_by: LIST [1:?] OF INTERVAL_VALUE; pref_attributes: LIST[1:?] OF PREFERENCE_URI; END_ENTITY; </pre>
<u>Örnek Değerler</u> <pre> #1=Preference_URI(100,'very cheap','cost'); #2=Preference_URI(101,'cheap','cost'); #3=Preference_URI(102,'expensive','cost'); #4= Preference_URI(103,'very expensive','cost'); #11= Interval_Value(20, 45); #12= Interval_Value(45, 60); #13= Interval_Value(60, 90); #17= Interval_Value(90, 100); #250= Interval_Preference (#11,#1); #251= Interval_Preference (#12,#2); #252= Interval_Preference (#13,#3); #253= Interval_Preference (#17,#4); </pre>

Diğer aralık değerleri (#11, #13 ve #17) yine benzer bir yaklaşım ile (*pahalı, çok pahalı ..*) tercih niteleyiciler ile buluşmaktadır.

Olasılık Tercihleri: Bölüm 5 de açıklanan ve tercihlerin belirli olasılık değerleri ile ifade edildiği durumları örnekleyen tercihlerdir. Çizelge 7.6 da yer aldığı şekilde EXPRESS sözdizimi ile sunulmaktadır.

Çizelge 7.6 Fuzzy_Preference Örnekleri

<u>EXPRESS Sözdizimi</u> <pre> TYPE PROB_VALUE= REAL; WHERE ((SELF>0) AND (SELF<1)); END_TYPE; ENTITY FUZZY_PREFERENCE SUBTYPE OF (INTERPRETED_PREFERENCE); interpreted_by: PROB_VALUE; pref_values: LIST [1:?] PROPERTY_VALUE; pref_attributes: LIST[1:?] OF PREFERENCE_URI; END_ENTITY; </pre>
<u>Örnek Değerler</u> <pre> #27=Preference_URI(150,'low','quality'); #28=Preference_URI(151,'standard','quality'); #30=Preference_URI(153,'lux','quality'); #80=('probvalue', 0.80) #67= ('tenniscourt', yes); #68= ('pool', yes); #77= ('starRate', 2); #80= ('starRate', 5); #300= Fuzzy_Preference (#80,[#67,#68],#28); </pre>

‘Standard otelleri’ tarifleyen #300 numaralı örnek; otel ontolojisi içinde örneklenen özniteliklerden (‘tenniscourt’, ‘pool’) öznitelik değerlerine sahip olan otelleri, 0.50 olasılık oranı ile *kalite (quality)* kategorisine göre *standard (lux)* tercih niteleyicisi ile buluşturmaktadır.

İkili Tercihler: İkili tercihler, ontolojik örnekleri aldıkları öznitelik değerlerine göre tercih edilen ya da edilmeyen olarak Bölüm 5 de açıklanmaktadır. Çizelge 7.7 de ise oluşturulan EXPRESS söz dizimi ile gösterilmektedir.

‘Pahalı otel tarifi için oluşturulan #350 numaralı örnek otel ontolojisi içinde verilen öznitelikler değerlerinden tenis kortu ve havuzu olan otelleri pahalı (expensive) tercih niteleyicisi ile buluşturmaktadır.

Çizelge 7.7 Boolean_Preference Örnekleri

<u>EXPRESS Sözdizimi</u>
<pre> ENTITY BOOLEAN_PREFERENCE SUBTYPE OF (INTERPRETED_PREFERENCE); interpreted_by: LIST[1:?] OF PROPERTY_VALUE; pref_attributes: LIST[1:?] OF PREFERENCE_URI; END_ENTITY; #2=Preference_URI(101,'cheap','cost'); #3=Preference_URI(102,'expensive','cost'); #67= ('tenniscourt', yes); #68= ('pool', yes); #350= Boolean_Preference ([#67,#68],#3); </pre>

Yukarıda oluşturulan tercih tiplerinin alan ontolojisi örnek değerleri ile ilişkilendirilmesi için gereken tercih bağı aşağıda açıklanmaktadır.

7.1.3 Tercih Bağı Oluşturulması

Bölüm 5 Kısım 5.3 de açıklanan tercih bağı Tercih Yönetim Çerçevesinin son elemanıdır. Otel ontolojisi üzerinde yer alan örnek değerlerin tercih tipleri ile buluşturulmasında anlamsal köprü olarak görev almaktadır. Senaryoda belirtilen *ucuz* ve *standart* tercihler için oluşturulan bağı Çizelge 7.8 da gösterilmektedir.

Çizelge 7.8 Tercih Bağları Örnekleri

```
#500=PREF_Link(#50,[#76,#250]);
#501=PREF_Link(#51,[#77,#251]);
#502=PREF_Link(#52,[#78,#252]);
#503=PREF_Link(#53,[#79,#253]);
```

Bu sayede Tercih Yönetim Çerçevesi içinde ki Kaynak elemanı örnek değerleri (otel ontolojisi için ((#50, #51, #52, #53) numaralı örnekler), tercih üst-şeması içinde tanımlı tercih tipleri ile ilişkilendirilebilmektedir. Böylece öznitelik ya da örnek değeri için bir kaynağın farklı tercih niteleyiciler ile tariflenebildiği gösterilmektedir. *Yıldız (starRate)* özniteliği için 3 değerine, ücret özniteliği değeri için 55 değerine sahip *Otel Kyriad*, kurulan tercih bağı (PREF_Link #501) ile Numeric_Preference ve Interval_Preference tercih tipleriye sahip olmaktadır. Bu sayede Numeric_Preference tercih tipleri içinde yer alan *standard* ve Interval Tercih tipinde yer alan *ucuz* tercih niteleyicileri *Otel Kyriad* a bağlanmaktadır.

7.1.4 Değerlendirme

Yukarıda yer alan ilk örnek uygulamada kavramsal bir model olarak tariflenen Tercih Yönetim Çerçevesi elemanları anlatılmaktadır. Oluşturulan kavramsal modele ilişkin bir değerlendirme aşağıda yapılmaktadır. Bu değerlendirme verilen örneklerle yönelik sağlanan bulguları ve modele yönelik tespit edilen olumlu ve olumsuz yanları bir arada sunmaktadır.

Tercih Yönetim Çerçevesinin ilk elemanı Kaynak varlığıdır. Kaynak varlığı, ODM üzerinde tariflendiği haliyle sınıf ve özellik örnek değerlerini kapsamaktadır. Ancak ODM üzerinde tanımlı iki farklı özelliğin bulunduğu unutulmamalıdır. Bunlar veri öznitelik özelliği ve nesne özelliği olarak belirtilmektedir. Tercih Yönetim Çerçevesi içinde tercihlerin *property_value* ve *class_value* değeri ile tarifleniyor olması nesne özelliğinin göz ardı edildiği izlenimini yaratmaktadır. Ancak ODM

içinde tanımlandığı şekliyle bir nesne özelliği iki sınıf arasında tanımlanan ilişki sonrasında oluşmaktadır. Ancak bu tip tercihlerin oluşturulmasında ulaşılabilecek tercihin yine bir `class_value` örnek değeri olması nedeni ile çalışma içinde göz ardı edilmediği, örtülü olarak ele alındığı bilinmektedir. Senaryoda yer alan `<hasRoom>` nesne özelliği üzerinden açıklanabilecek bu durum, `accomodation` ve `room` sınıfları arasında kurulan ilişki için yine `room` sınıfına ait `class_value` örneklerine ulaşıldığını göstermektedir .

Oluşturulan Tercih Yönetim Çerçevesi elemanları örnek değerleri birçok çelişkili durumu beraberinde getirmektedir. Bir kısmı çelişki, tercihlerin bir kavramsal model ile ifade edilebilmesi sırasında karşılaşılan zorluktan, diğer bir kısım çelişki ise oluşturulan örneklerin doğasından kaynaklanmaktadır. Tespit edilen en belirgin durum tercih niteleyicilerin ilgili tercih tipi ile birlikte kullanılması sırasında oluşmaktadır. Oluşturulan kavramsal modele yönelik verilen örneklemelerde asıl karar vericinin yine bir insan olduğu gözlenmiştir. Yukarıda yer alan senaryodan da anlaşılacağı gibi portal üzerinde yer alan örnek değerlere uygun tercih tiplerinin atanması işlemini bir sistem yöneticisinin yapması gerektiğidir. Bu durumda da farklı örnek değerlerin farklı tercih niteleyiciler ile buluşabileceği, olası çakışmaların yaşanabileceği unutulmamalıdır.

Tercih niteleyicilerin kişisel tercihlerin ifade edilebilmesinde ki önemi tez çalışması boyunca öne çıkarılmaktadır. Bu nedenle tercih niteleyiciler URI ler olarak düşünülmüş, tanımlarını farklı adreslerden alabilecekleri, sahip oldukları kategorilerine göre farklı tercih niteleyici isimleri ile tarflenebilecekleri vurgulanmıştır. Bu sayede Anlamsal Web'in doğasına uygun dağıtık mimariler örneklenebilmekte, oluşturulan kategoriler içinde yer alan tercih niteleyici isimlerinin çeşitliliğinin genişletilebilmesi sağlanmaktadır.

Oluşturulan çerçevenin ikinci elemanı Tercih üst-şemasıdır. Tercih üst-şeması alan yazında tarflenen tercih tiplerini bir araya getirmek üzere yaratılmıştır. Ancak bulunan tercih tiplerinin anlamsal bir yapı içine oturtabilmesi sırasında bir takım zorluklar ile karşılaşmıştır. Oluşturulan

hierarchy yapının farklı kategorileri içerecek şekilde genel tarifleri içermesi gerekmektedir. Bu nedenle tercihler yorumlanabilir tercihler ve yorumlanamayan tercihler olmak üzere ikiye ayrılmıştır. Ancak yorumlanamayan tercihlerin tanımlanması sırasında çeşitli çelişkiler gözlenmiştir. Bunlar kullanılan alan ontolojilerinden ve örneklenen durumlardan oluşmaktadır. Alan ontolojileri üzerinde yer alan örnek değerler kimi zaman bir sınıf kimi zaman bir öznitelik olarak tanımlanmaktadır. Tercih edilecek varlığın hangi kategoride yer aldığı hangi hierarchy için oluşturulduğu öncelikle ilgili ontolojinin pars edilmesi ile tespit edilmelidir. Tez çalışması sırasında bu durum tespit edilmiş ancak bir uygulama ile gerçekleştirilmemiştir.

Yorumlanabilir tercih tipleri altında ise alan yazında belirtilen örneklerde olduğu gibi, farklı fonksiyonlar ile tanımlanması olası durumlar bir araya getirilmiştir. En kolay tarflenebilen tercih tipi olarak belirlenen aralık tercih tipi aralık değerlerine atanan tercih niteleyiciler ile sağlanmaktadır. Sayısal ve Boolean tercih tipleri için öncelikle kullanılacak alan ontolojisinin pars edilmesi, ilgili ontoloji üzerinde yer alan öznitelik tiplerine uygun olarak öznitelik değerlerinin gruplandırılması ve sonrasında örneklerin oluşturulması gerekmektedir. Senaryo gereği turizm alanı için oluşturulan otel ontolojisi içinde yıldız sayısı sayısal tercih tipi için başarılı bir örnektir. Elektronik ticaret uygulamalarında ise genellikle kullanıcı memnuniyetini belirten değerler sayısal tercih tipi ile temsil edilen duruma bir örnek teşkil etmekte diler. Boolean tercih tipi kullanıcıların genellikle tercih ettikleri öznitelik değerlerini bir arada sunmaktadır. Ancak tercih edilmeyen varlıklardan tercih edilen varlıklara ulaşılabilceği unutulmamalıdır, bazı durumlarda baskın olan ve tercih edilmeyen özelliklerin bir arada sunulması gerekmektedir. Olasılık tercih tipi, tez çalışması içinde örneklenmesi gerekli bir diğer durumun temsili için yaratılmıştır. Ancak olasılık değerinin kime ve neye göre nasıl belirlenebileceği üzerinde düşünülmesi gereken bir durumdur. Verilen örnekte ise bu değeri sistem yöneticisinin atadığı düşünölmektedir.

Çalışma boyunca tercih üst-şeması üzerinde yer alan farklı tercih tipleri arasında çakışmaların oluştuğu durumlar gözlenmiştir. En dikkat çekici çakışma listelenmiş tercihler ile yorumlanamayan tercihler arasında gözlenmiştir. Bölüm 5 de yer alan söz dizimlerine bakıldığında her iki tercih tipinin de benzer

şekilde tanımlandığı gözlenmektedir. Ancak farklı durumların örneklenmesi sonucu aldıkları değerler kümesinin değiştiği bilmekte, bu nedenle de üst-şema içinde ayrı ayrı ele alınmaktadır.

Son olarak model üzerinde kişisel tercihlerin değişken ve dinamik olması, alan bilgisinin değişken ve dinamik olması ile açıklanmaktadır. Bağlam bazlı olarak değişen tercihler, tercih-üst şeması içinde ele alınmaktadır. Ancak ortaya konan tanım, literatürde yer alan çalışmalarda olduğu gibi zaman ve kişiselleştirme bazlı olarak örneklendirilmemektedir. İlgili tercih tipi, tercih üst-şeması içinde yer alan yorumlanabilir tercih tipleri cinsinden tariflenmekte ve ilişkilendirildiği tercih tipine atanan değerlere göre uygun tercih niteleyecileri ile birlikte örneklendirilmektedir.

Oluşturulan model daha sonra ki kısımda açıklanacak olan örnek uygulama ile ontoDB üzerinde gerçekleştirilmektedir.

7.2 Örnek-2: OntoDB Üzerinde Tercih Temelli Sorgulama Gerçekleştirimi

Bu uygulamada, 4.Bölümde açıklanan OntoQL dili yetenekleri kullanılarak Bölüm 6 da anlatılan OntoDB mimarisinin Tercih üst-şeması ile genişletilmesi ve bu doğrultuda mimari üzerinde tercih temelli sorgulama gerçekleştirimi anlatılmaktadır.

7.2.1 Alan Ontolojisi Geliştirimi

Bölüm 7 Kısım 7.1.1 de açıklandığı gibi öncelikle OntoDB mimarisi üzerinde Tercih Yönetim Çerçevesinde yer alan Kaynak tanımı ilgili alan ontolojisi ile oluşturulmaktadır. Çizelge 7.9 de bu ontolojinin CREATE komutu ile oluşturulması gösterilmektedir.

Çizelge 7.9 OntoQL Dili ile Otel Ontoloji Yaratımı.

```
CREATE Class Hotel(
    #id int, #name String,
    #starRating int, #price int,
    #airCond boolean, #tv boolean,
    #wifi boolean, #pool boolean,
    #jakuzi boolean, #tennisCourt Boolean
);
```

Oluşturulan ontolojiye ilgili değerlerin atanması INSERT INTO komutu kullanılarak gerçekleştirilmektedir. Çizelge 7.10 da alan ontolojisi örnek değerleri (*HotelFormule1* ve *Hotel Kyriad*) gösterilmektedir.

Çizelge 7.10 Ontolojik Olguların Örneklenmesi

```
INSERT INTO Hotel (id, name, starRating, price,
    airCond, tv, wifi, pool, jakuzi, tennisCourt)
VALUES (51, 'HotelFormule1', 2, 30, 'yes',
    'yes', 'no', 'no', 'no', 'no');

INSERT INTO Hotel (id, name, starRating, price,
    airCond, tv, wifi, pool, jakuzi, tennisCourt)
VALUES (51, 'HotelKyriad ', 3, 55, 'yes', 'yes',
    'yes', 'no', 'no', 'no');
```

Bir sonra ki kısımda ise Tercih üst-şemasının OntoDB mimarisi üzerinde oluşturulması ve alan ontolojisi ile üst şemanın entegrasyonu açıklanmaktadır.

7.2.2 OntoDB Üzerinde Tercih Modeli Gösterimi

Örnek uygulama Kısım 7.1.2 de örneklenen sayısal ve aralık tercih tipleri için oluşturulmaktadır. Çizelge 7.11 ve 7.12 de ilgili tercih tiplerinin OntoQL dili ile yaratılması ve örneklendirilmesi gösterilmektedir.

Çizelge 7.11 Sayısal Tercih Örneklenmesi

```
CREATE #Numeric_Preference(
#number_value INT,
REF(Preference_URI)
);
INSERT INTO #Numeric_Preference(#number_value,#code,
#name, #classification)
VALUES(2, 50,'low','starRating');
```

Çizelge 7.12 Aralık Tercihi Örneklenmesi

```
CREATE Interval_Preference(
#min_value INT,
#max_value INT,
REF(Preference_URI)
);
INSERT INTO Interval_Preference(#min_value, #max_value,
#code, #name, #classification)
VALUES(45, 60, 100,'cheap','cost');
```

Bir sonra ki kısımda, OntoDB mimari üzerinde yer alan üst-şema kısmı ile mimari üzerinde M2 seviyesinde yer alan ontoloji modelinin entegrasyonu gösterilmektedir.

7.2.3 Tercih Modeli ve Alan Ontolojisi Entegrasyonu

OntoQL dili kullanılarak oluşturulan örnek ontoloji ve Tercih üst-şeması içinde yer alan ilgili tercih tiplerinin yaratılması sonrasında OntoDB mimarisi üzerinde yer alan üst-şema kısmı ALTER komutu kullanılarak genişletilmektedir. ALTER komutu Çizelge 7.13 de ve EK-2 de açıklandığı gibi bir veritabanı üzerinde yapılacak değişiklikler için kullanılmaktadır.

Çizelge 7.13 ALTER Komutu

```
ALTER ENTITY
#Property_Or_Class ADD
ATTRIBUTE #PREF_Link REF
(#Preference) ARRAY;
```

Çizelge 7.13 da Tercih üst-şeması ile OntoDB mimarinin üst-şema kısmının genişletilmesi için kullanılan komut satırı yer almaktadır. Belirtilen

genişletme mimari üzerinde yer alan `attribute` tablosuna yeni bir attribute olarak `Pref_Link` satırının eklenmesi ile sağlanmaktadır. Bu nedenle OntoDB de üst-şema seviyesinde yer alan `Property_Or_Class` varlığına bir eklenti yapılmaktadır.

Bir sonra ki adımda ise `Pref_Link` özniteliği içine ilgili değerlerin atanması için `UPDATE` komutu kullanılmaktadır. EK-3 de açıklandığı gibi `UPDATE` komutu veritabanı üzerinde istenilen bir alanın güncellemesini sağlamaktadır. Çizelge 7.14 da yer alan örnek uygulama da ise sınıf değeri üzerinde gerekli güncelleme nasıl yapıldığını sunulmaktadır. Bu güncelleme sonunda o sınıf içinde yer alan örnek değerlere ilgi tercih niteleyiciler ile ulaşılması sağlanmaktadır.

Çizelge 7.14 UPDATE Komutu

```
UPDATE Hotel set #Pref_link=ARRAY
['cheap','standard','full board']

WHERE name='Kyriad';
```

Belirtilen çizelgede okunabilirliğin artırabilmesi için gereken tanımlayıcılar yerine tercih niteleyicilere ait *isim özniteliği* kullanılmaktadır. Bu sayede tercihlerin örnek değerlere atanması gerçekleştirilmektedir. OntoDB üzerinde kurulan tercih bağı sayesinde artık alan bilgisi kullanıcı tercihleri ile sorgulanabilir bir şekilde saklanmaktadır. Bir sonra ki kısımda Tercih temelli sorgulama gerçekleştirim örnekleri sunulmaktadır.

7.2.4 Sorgulama Örnekleri

Bu kısımda Bölüm 6 da açıklanan `PREFERRING` operatörünün kullanımı tercih temelli sorgulama gerçekleştirimi ile birlikte açıklanmaktadır. Senaryoda örneklenen *standard* oda tercihi için öncelikle üst şema üzerinde tariflenen `Preference_URI` tercih niteleyicisi kullanılarak aşağıda ki sorgular yazılmaktadır;

```
SELECT name, starRating, pref
FROM Hotel
PREFERRING 'standard';
```



```
SELECT p.#name, pref.#number value
FROM #property p,
     unnest(p.#pref_link) as pref
where pref.#label='standard';
```

```
SELECT name, starRating, pref
FROM Hotel
WHERE starRating < 3;
```

name	starRating	pref
Hotel Kyriad	3	standard

Verilen örneklerde OntoDB mimarisi üzerinde yer alan örnek değerlere tercih üst-şeması üzerinde yer alan iki tercih tipinin atanması durumu gösterilmiştir.

```
SELECT name, price, pref
FROM Hotel
PREFERRING 'cheap';
```



```
SELECT p.#name,
pref.#min_value, pref.#max_value
FROM #property p,
     unnest(p.#pref_link) as pref
where pref.#label='cheap';
```

```
SELECT name, price, pref
FROM Hotel
WHERE price BETWEEN 45 and 60;
```

name	price	pref
Hotel Kyriad	55	cheap

Oluşturulan PREFERRING operatörü ve farklı tercih niteleyicilerin yan yana kullanılması AND operatörü ile sağlanabilmektedir. Bu sayede senaryoda örneklenen “ucuz ve standard oda” tercihi dönen cevap *Otel Kyriad* dır.

7.2.5 Değerlendirme

Bu bölümde sunulan ikinci uygulama OntoDB mimarisi üzerinde yer alan üst şema kısmının Tercih üst-şeması ile genişletilmesini açıklamaktadır.

Tez çalışmasının Anlamsal Web uygulamaları için geliştirildiği unutulmamalıdır. Ancak alan yazın incelemesi sırasında (bkz. Çizelge 2.4) belirlenen durum gereği çalışma Anlamsal Web çalışmalarını baz alan ve veritabanı araştırmaları ile arasında bir köprü görevi gören Ontoloji Tabanlı Veritabanları için oluşturulmaktadır. Bu nedenle ODM yapısı OntoDB mimarisi üzerinde incelenmekte, mimarinin üst-şema yapısının nasıl genişletildiği bir uygulama ile açıklanmaktadır.

Kavramsal model olarak oluşturulan Tercih Yönetim Çerçevesinin OntoDB mimarisine taşınması sırasında ise yine öncelikle Kaynak varlığı tanımlanmaktadır. Bu tanım farklı ontoloji dillerine hizmet verilebilecek (PLIB, OWL-F) bir yapının gerçekleştirimini sağlamaktadır. Böylece Tercih Yönetim Çerçevesinin farklı platformlara taşınması mümkün kılınmaktadır.

Tez çalışma boyunca oluşturulan çerçevenin OWL dili yetenekleri ile yapılandırılması için ayrıca çalışılmıştır. Süreçte OntoQL dili ile oluşturulan yapı öne çıkartılarak tez çalışması tamamlanmıştır. Ancak OntoQL dili içinde kullanılan tanımlamaların oluşturulacak scriptler ile OWL dili içinde kullanılan sınıf tanımlarına ve özelliklerine dönüştürülmesinin mümkün olduğu belirlenmiştir.

OntoDB mimarisi üzerinde PLIB ontoloji modeli kullanılmaktadır. Bu modelin oluşturulması sırasında EXPRESS veri modelleme dili öne çıkartılmaktadır. Ancak OntoDB mimarisi üzerinde mutlak kullanılması gereken modelleme dili olarak belirtilmemektedir. Bir prototip oluşturmada sağladığı avantajlardan faydalanabilmek adına model ve modele ilişkin örneklerin oluşturulmasında kullanılmaktadır.

OntoDB mimarisi üzerinde tercih bağının oluşturulması gerekmektedir. Bu nedenle OntoQL dili özelliklerine başvurulmuştur. Mimari üzerinde yapılan yapısal değişiklik ile yeni bir özellik üst-şemaya eklenmiş,

böylece tercih-üst şeması ile alan ontolojileri üzerinde yer alan örnek değerlerin ilişkilendirilmesi sağlanmıştır.

Bu bölümde sunulan ikinci önemli nokta OntoQL sorgulama dilini tercih temelli sorgulama gerçekleştirmiş için kullanabilmektir. Tercih üst-şeması ile entegre edilen sorguların bir kısmı çalışma boyunca gerçekleştirebilmiştir. OntoQL dili ile mimari üzerinde farklı seviyelerde sorgulamalar gerçekleştirilmektedir. İlk seviye sistem tabloları üzerinden kayıtlı verilere yönelik yapılan sorgulamaları, ikinci seviye ontolojiler üzerinden ontolojik verinin sorgulanmasını, üçüncü seviye ise ontoloji modeli üzerinden ontolojilerin sorgulanmasını sağlamaktadır. Her sorgulama seviyesi ile mimarinin sağladığı avantajların örneklendirildiği unutulmamalıdır.

Bu bölümde sunulan örnek uygulama, ontolojik bilginin kişisel tercihler doğrultusunda sorgulanabildiğini göstermektedir. Bu nedenle öncelikle *pref_link* özelliği mimari üzerinde tanımlanmaktadır. *Pref_link*, Tercih üst-şeması ile alan ontolojisi arasında kurulan bağlantıyı tanımlayan yeni bir kolon olarak düşünülmelidir. Bu sayede yapılan sorgulamalarda kullanılan tercih niteleyici ismi ile ontolojik verilere ulaşılabilir. Bu kısımda örneklenen iki sorgu tercih üst-şeması üzerinde bulunan iki farklı tercih tipi için oluşturulmuş ve bu sayede *tercih operatörünün* çalıştığı gösterilmiştir. *Tercih Operatorü* hem alan bilgisinin kişisel tercihler ile sorgulanabilmesi adına hem de kişisel sorgu oluşturulmasında tercih üst-şemasının kullanılabilmesi adına çalışmaya değer katmaktadır. Ancak tez çalışması boyunca yapılan alan yazın incelemesinde örneklenen Kießling ve Siberski 'nin çalışmalarında olduğu gibi *PREFERRING* operatörün yeteneklerinin artırılması gerekmektedir. Birden çok tercih tipini içeren karma sorgu örneklerine duyulan ihtiyaç alan yazın çalışmasında yer almaktadır. Ancak tez çalışması kapsamında farklı tercih tiplerinin bir araya getirilmesi ile oluşturulacak karma tercihlerin sorgulanması sağlanamamıştır. Kişisel sorguların oluşturulmasında incelenen tercih edilen varlıklara verilen önem ve öncelik veritabanı araştırmalarında ortaya konmaktadır. Pareto operatörü, *top(k)* sorguları ve skyline algoritmaları ayrı ayrı tercihleri önceliklendirme ve sıralandırabilmeyi hedef almaktadır. Tez çalışması içinde yer alan kişisel tercihlerin sorgulanması içinde bu bakış açısı ele alınmamıştır.

Bir sonraki bölümde, çalışma boyunca karşılaşılan zorluklar, ileri ki çalışmalara verebilecek katkılar birlikte değerlendirilmektedir.

8 SONUÇLAR

Günlük hayatta kullanıcı tercihlerinin tariflendiği alan bilgisinin dinamik, değişen, devingen yapısından etkilendiği gözlenmektedir. Bu nedenle tez çalışmasının benzer bir alt yapı üzerinde oluşturulmasına karar verilmiş, Anlamsal Web araştırma alanı için bir çözüm önerisi üzerinde çalışılmıştır. Belirtilen çalışma alanı ontolojileri kullanmaktadır. Ontolojiler dağıtık mimarilerde dinamik olarak değişen bilgiyi saklamaktadır. Literatürde Anlamsal Web araştırmaları ile paralellik gösteren ancak mimarisi gereği içinde veritabanı tablolarını barındıran Ontoloji Tabanlı Veritabanları (OTV) araştırmaları da bulunmaktadır. Ancak bu alanda henüz kişiselleştirme yaklaşımlarının sunulduğu, kullanıcı tercihlerine yönelik sorgulamaların örneklendiği çalışmalar yer almamaktadır. Tez çalışması kapsamında bir OTV mimarisi olan OntoDB nin kullanımı benimsenmiş, belirtilen mimari üzerinde kullanıcı tercihleri için bir kavramsal model oluşturulmuş, mimaride yer alan üst-şema yapısı genişletilmiş ve oluşturulan kavramsal model ile tercih tabanlı sorgulama gerçekleştirimi sağlanmıştır.

Model kavramı olguların sistem yaklaşımıyla incelenmesi ve tasarlanmasında kullanılmaktadır. İlişkisel veri modeli veritabanı yönetim sistemleri için bir standardı temsil etmektedir. Yapılan alan yazın araştırmasında mantıksal modelin sunulduğu ilişkisel veritabanı teknolojisinin karmaşık bilgi sistemlerinin oluşturulmasına yönelik olarak belirlenen hedefleri karşılamadığı gözlenmiştir. Oysa nesneye yönelik veritabanı tasarımı ile son derece karmaşık bilginin hızlı bir şekilde yönetilmesi mümkündür. Bilindiği gibi veritabanı kendi içinde verinin anlamı hakkında bilgi saklayamamaktadır. Bu da veritabanı uygulamalarında sınırlama oluşturmuştur. Oysa anlambilimsel veri modeli ile verinin yapısının geliştirilmesi sağlanmakta, uygulamalardaki kısıtlar ortadan kaldırılmaktadır.

Kavramsal modeller için oluşturulan üst-şemalar karmaşık problemlere çözüm bulmak amacıyla kullanılmaktadır. Bir kavramsal modelin geliştirmesi için farklı görsel özelliklere sahip diller kullanılmaktadır. Bu nedenle öncelikle Bölüm 3 de değişik bakış açıları ile bir kavramsal modelin nasıl tarifienebileceği açıklanmıştır. Hiçbir model tanımı, tanımı gereği hedef sistemin tüm özelliklerini içermemektedir. Ancak sistemin o anda incelenen özelliklerini ve sistem

davranışlarının en belirgin özelliklerini mutlak olarak içermektedir. Tercih Yönetim Çerçevesi içinde bu nedenle göz ardı edilen durumlar söz konusudur.

Tercihler karar verme odaklı süreçler içinde değerlendirilmektedir. Günlük yaşantıda belirtilen süreçler için öncelikle tercihlerin ifade edilmesi gerekmektedir. Bu durum incelendiğinde tercihlerin alan bilgisi üzerinde kullanılan örnek değerleri ve bu değerlere atanan tercih niteleyici isimleri içerdiği gözlenmiştir. Yapılan alan yazın çalışmasında tezde öne çıkarılan bu yaklaşımı örnekleyen bir araştırma bulunmamıştır. Tercih Yönetim Çerçevesi içinde, kullanıcı seçimleri olarak tariflenen tercihlerin değerlerini alan bilgisinden, tanımlarını oluşturulan Tercih üst-şemasından aldıkları kabulü Bölüm 5.1 de belirtilmiştir. Bu kabul ile alan bilgisine sıkı sıkıya bağlı olan tercihlere ait değerlerin, alan bilgisinin dinamik ve değişken yapısından olumlu anlamda etkilendiği ancak üst-şemadan gelen tanımlarının etkilenmediği gösterilmiştir. Oluşturulan kavramsal model için seçilen EXPRESS veri modelleme dili ile Tercih Yönetim Çerçevesi oluşturulmuş ve aynı çerçevenin UML modelleme dili ile de sunulabildiği belirtilmiştir. Bu sayede UML gibi kabul görmüş MOF tabanlı bir standarda uygun olarak Tercih üst-şemasının geliştirilebileceği, oluşturulan kavramsal modelin UML Profillerin sağladığı yetenekler ile birlikte değerlendirilebileceği fikri oluşmuştur. UML Profilleri kullanılarak farklı üst şemaların genişletildiği alan yazın çalışmasında örneklendirilmiştir. Aynı bakış açısı Tercih Yönetim Çerçevesi için de geçerli olmakta, bu sayede tez çalışmasında oluşturulan Tercih-üst şemasının herhangi bir ontoloji modelini tariflemek için kullanılan ontoloji üst-şemasını genişletilebileceği fikrine gelinmektedir.

Tercih Yönetim Çerçevesi içinde kullanıcı tercihlerinin ODM ve ODM de yer alan değer varlıklar ile üst-şema (M2) seviyesinde genel ve soyut bir model oluşturacak şekilde Kaynak tanımı altında sunulmuştur. Kaynak varlığı, ODM üzerinde tariflendiği haliyle sınıf ve özellik örnek değerlerini kapsamaktadır. Ancak ODM üzerinde tanımlı iki çeşit özellik bulunduğu unutulmamalıdır. Bunlar veri öznitelik özelliği ve nesne özelliğidir. Tercih Yönetim çerçevesi içinde tercihlerin `property_value` ve `class_value` değeri ile tarifleniyor olması nesne özelliğinin göz ardı edildiği izlenimini yaratabilir. Ancak ODM içinde tanımlandığı şekliyle bir nesne özelliği iki sınıf arasında tanımlanan ilişki sonrasında oluşmaktadır. Sınıflar (domain, range)

arasında oluşturulan bu ilişkinin de bir tercih olarak belirtilebileceği durumlar bulunmaktadır. Bu tip tercihlerin oluşturulmasında yine ulaşılacak varlık bir `class_value` örnek değeri olacaktır. Bu nedenle çalışma içinde göz ardı edilmiş gibi gözükse de bu durum aslında örtülü olarak ele alınmaktadır.

Kaynak tanımı içinde yer alan varlıkların genel geçer bir ontoloji modelinin parçası olarak, farklı ontoloji modelleri içinde `<class>` ve `<property>` varlıkları ile temsil edilen yapıya Tercih Yönetim Çerçevesi içinde dönüştürülmesi sağlandıktan sonra, bu yapının herhangi bir ontoloji modelini kullanan bir ontoloji geliştirme diline taşınabileceği fikri ortaya çıkmaktadır.

M1 seviyesinde alan ontolojileri ve M0 seviyesinde Tercih ontolojisine ait örnek değerler yer almaktadır. Bu doğrultuda tercihlerin, M1 seviyesinde ki Tercih üst-şemasından türediği fikri gösterilmektedir. Ancak bu durum karşısında genel olarak tariflenen bir ontoloji modelini tanımlayan Kaynak varlığı ve Tercih varlığı ile temsil edilen Tercih üst-şemasının M2 seviyesinde ilişkilendirilmesi gerekmektedir. Sonuç olarak, oluşturulan Tercih üst-şemasının ontoloji modelini genişletebilmek için M2 seviyesinde, ontolojilere ait örnek değerleri model üzerinde tarifleyebilmek için ise M1 seviyesinde kullandığı gözlenmektedir.

Oluşturulan çerçevenin ikinci elemanı Tercih üst-şemasıdır. Tercih üst-şeması alan yazında tariflenen tercih tiplerini bir araya getirmek üzere yaratılmıştır. Ancak bulunan tercih tiplerinin anlamsal bir yapı içine oturtabilmesi sırasında bir takım zorluklar ile karşılaşmıştır. Tercihler yorumlanabilir tercihler ve yorumlanamayan tercihler olmak üzere ikiye ayrılmıştır. Yorumlanamayan tercihlerin tanımlanması sırasında çeşitli çelişkiler gözlenmiştir. Bunlar kullanılan alan ontolojilerinden ve örneklenen durumlardan oluşmaktadır. Alan ontolojileri üzerinde yer alan örnek değerler kimi zaman bir sınıf kimi zaman bir öznitelik olarak tanımlanmaktadır. Tercih edilecek varlığın hangi kategoride yer aldığı hangi hiyerarşi için oluşturulduğu öncelikle ilgili ontolojinin pars edilmesi ile tespit edilmelidir.

Yorumlanabilir tercihler altında ise alan yazında belirtilen örneklerde olduğu gibi, farklı fonksiyonlar ile tanımlanması olası durumlar bir araya getirilmiştir. Boolean tercih tipi için kullanıcıların genellikle tercih ettikleri öznitelik değerleri bir arada sunulmaktadır. Ancak alan yazında incelenen araştırmalarda tercih edilmeyen özniteliklerin bir arada sunulduğu örnekler

bulunmaktadır. Bu sayede tercih edilmeyen varlıklardan tercih edilen varlıklara ulaşılabileceği düşünülmekte, bazı durumlarda baskın olan tercih edilmeyen özelliğin sunulması gerekebilmektedir. Olasılık tercih tipi tanımı ise tez çalışması içinde örneklenmesi gerekli bir diğer durumun temsili için yaratılmıştır. Ancak olasılık değerinin kime ve neye göre, nasıl belirlenebileceği üzerinde düşünülmesi gereken bir durumdur. Verilen örnekte sistem yöneticisinin atadığı bir değer olarak düşünülmektedir. Sayısal tercih tipi, sayısal bir değişken olarak tanımlanan öznitelik değeri için kullanılmaktadır ancak bazı araştırmalarda sayısal tercihlerin dışarıdan tanımlanan sayısal derecelendirme fonksiyonları için kullanıldığı bilinmektedir (örn: flim sitelerinde kullanılan yıldız değerleri). Tez çalışması kapsamında dışarıdan ontoloji örnekleri derecelendiren bir fonksiyon kullanımı ile ulaşılabilecek değerlerin sayısal tercihler olarak ele alındığı durumlar ele alınmamaktadır.

Çalışma boyunca Tercih üst-şeması üzerinde yer alan farklı tercih tipleri arasında çakışmaların ve çelişkilerin oluşabileceği durumlar da gözlenmiştir. Çakışmalar özellikle tercih niteleyicilerin ilgili tercih tipi ile kullanılması sırasında yaşanmıştır. Verilen örneklemelerde karar vericinin yine bir insan olduğu gözlenmiş, bu nedenle kaynak değeri olarak sunulan örnek değerlere uygun tercih tiplerinin atanması işleminin sistem yöneticisi tarafından yapılması gerektiği fikri ile bir takım çakışmalarının giderilebileceği belirlenmiştir.

Oluşturulan kavramsal modelin açıklanması sırasında yaşanan çelişkili durumların başında ise listelenmiş tercihler ile yorumlanamayan tercihler gelmektedir. Bölüm 5 de yer alan söz dizimlerine bakıldığında her iki tercih tipinin de benzer şekilde tanımlandığı gözlenmektedir. Ancak farklı durumların örneklenmesi sonucu aldıkları değerler kümesinin değiştiği bilmekte bu nedenle de üst-şema içinde ayrı ayrı ele alınması gerektiği fikri benimsenmektedir. Kişisel tercihlerin değişken ve dinamik olması alan bilgisinin değişken ve dinamik olması ile açıklanmaktadır. Zaman ve bağlam bazlı olarak değişen tercihler, tercih-üst ontolojisi içinde sadece bağlam bazlı olarak incelenmektedir. Bağlam tanımı değişken durumları örnekleyebilmek için oluşturulmuştur. Zaman tanımı literatürde yer alan çalışmalarda tercih ve kişiselleştirme bazlı olarak örneklendirilmektedir, ancak tez çalışması boyunca zamana bağlı durumların örneklendiği tercih tipleri kapsam dışında bırakılmıştır. Bağlam temelli tercih tipi tercih üst-şeması içinde ise yorumlanabilir tercih tipleri cinsinden tariflenmekte ve ilişkilendirildiği tercih tipine atanan değerlere göre

farklı deęerler alabileceęi durumlar, eřleřeceęi farklı tercih niteleyicileri ile birlikte sunulmaktadır.

OntoDB mimarisi içinde yer alan üst-řema kısmı Anlamsal Web alıřmalarına örnek teřkil eden yapıyı, ontoloji modeli ile uyumlu olan veritabanı kısmı ise nesnel veritabanları ile benzerlikler gsteren yapıyı sunmaktadır. Belirtilen mimaride üst-řema tanımlayıcıları ve sorgulayıcıları (üst-řema tanımlayıcı ve sorgulayıcı diller) OntoQL dili kullanılarak tanımlanmaktadır. Bylece girilen/tanılan üst modeller ıřıęında oluřturan veriler, tanıtılan sorgulama dilleri ile sorgulanabilmektedir. Bu yapı içinde bir ontoloji tanımlamak için gereken ontolojik üst-model bulunmaktadır. Belirtilen üst-model üzerinde tanımlanan concept varlıęı, Tercih Ynetim erevesi içinde yer alan Kaynak tanımı ile eřleřmektedir. Bu sayede kullanılacak ontoloji dili ne olursa olsun bir ontoloji tanımlamak için kullanılması gereken temel class ve property varlıklarına üst řema seviyesinde ulařılmaktadır. Aynı yapı Blm 5 de aıklanan Tercih üst-řemasının tanımlanacaęı kısım olarak da grev yapmaktadır. Bu seviyede mimari üzerinde yeni bir modelin oluřturulması, var olan yapının geniřletilmesi OntoQL dili yetenekleri ile saęlanmıřtır.

OntoDB mimarisi üzerinde yer alan ve farklı ontoloji modellerinin mimari üzerinde kullanılmasına imkan saęlayan üst řema kısmının Tercih üst-řeması ile geniřletilmesi gerekleřtirmisi sırasında ncelikle Kaynak tanımı ele alınmıř, bu yapıyı temel alan farklı ontoloji dillerine de hizmet verilebilecek (PLIB, OWL-F) bir yapının gerekleřtirmisi saęlanmıřtır. Bylece Tercih Ynetim erevesi elemanlarının farklı bir platforma tařınabileceęi gsterilmiřtir. alıřma boyunca oluřturulan erevenin OWL dili yetenekleri ile yapılandırılması için ayrıca alıřılmıř, ancak srete OntoQL dili ile oluřturulan yapı ne ıkartılmıřtır. OntoQL dili içinde kullanılan tanımlamaların oluřturulacak scriptler ile OWL dili içinde kullanılan sınıf tanımlarına ve zelliklerine dnřtrlmesi mmkn olduęu bilinmektedir.

OntoDB mimarisi üzerinde PLIB ontoloji modeli kullanılmaktadır, bu modelin oluřturulması sırasında EXPRESS veri modelleme dili kullanılmıřtır. EXPRESS dili OntoDB mimarisi üzerinde mutlak kullanılması gereken bir modelleme dili olarak tanımlanmamaktadır. Ancak bir prototip oluřturmada saęladıęı avantajlardan faydalanılmıřtır.

OntoDB mimari üzerine taşınan *Tercih üst-şemasının* mimarinin sahip olduğu üst-şema bölümünü genişletmesi için kavramsal model üzerinde sunulan *tercih bağının* oluşturulması gerekmektedir. Bu nedenle OntoQL dili özelliklerine başvurulmuştur. Mimari üzerinde yapılan yapısal değişiklik ile yeni bir özellik üst-şemaya eklenmiştir. Böylece *tercih-üst şeması* ile alan ontolojileri üzerinde yer alan örnek değerlerin ilişkilendirilmesi sağlanabilmektedir. Ancak bu ilişkilendirmenin yapılabilmesinde sistem yöneticisine ihtiyaç duyulmaktadır.

OntoQL dili mimari üzerinde yer alan dört bölüm arasında farklı seviyelerde sorgulamalar gerçekleştirilmektedir. Her sorgulama seviyesinin mimarinin sağladığı avantajları örneklediği unutulmamalıdır. Tez çalışması için bu kısımda sunulan örnek uygulama ile öne çıkartılan ontolojik bilginin kişisel tercihler ile sorgulanabilmesidir. *Pref_link* özelliği mimari üzerinde *tercih üst-şeması* ile alan ontolojisi arasında kurulan bağlantıyı tanımlayan bir kolon olarak düşünülmelidir. Bu sayede yapılan sorgulamalarda kullanılan *tercih* niteleyici ismi ile ontolojik verilere ulaşılabilceği gösterilmektedir.

Bölüm 6 da ele alınan kişiselleştirme bakış açısı için ise kişisel sorguların oluşturulması yaklaşımı kullanılmaktadır. Bu yaklaşım OntoDB mimarisi üzerinde *Tercih üst-şemasının* kullanılması ile sağlanmıştır. Bu nedenle *tercih üst-şeması* ile entegre edilecek bir *tercih operatörünün* yaratılması gerektiği çalışmada önem kazanmıştır. *Tercih Operatörü* hem alan bilgisinin kişisel tercihler ile sorgulanabilmesi adına hem de kişisel sorgu oluşturulmasında *tercih üst-şemasının* kullanılabilmesi adına çalışmaya değer katmaktadır. Ancak tez çalışması boyunca yapılan alanyazın incelemesinde örneklenen Kießling ve Siberski'nin çalışmalarında olduğu gibi *PREFERRING* operatörün yeteneklerinin artırılması gerekmektedir. Çalışmada kavramsal model içinde yer alan tüm *tercih tiplerinin* sorgulanması ayrı ayrı yapılandırılmıştır. Bu kısımda örneklenen iki sorgu, *tercih üst-şeması* üzerinde bulunan iki farklı *tercih tipi* için oluşturulmuş ve bu sayede *tercih operatörünün* çalıştığı gösterilmiştir. Ancak birden çok *tercih tipinin* içeren karma sorgu örneklerine değinilmemiştir. Kişisel sorguların oluşturulmasında önemle incelenmesi gereken, tercih edilen varlıklara verilen önem ve önceliktir. Yapılan veritabanı araştırmalarında ortaya konan *Pareto operatörü*, *top(k)* sorguları ve *skyline* algoritmaları ayrı ayrı tercihleri önceliklendirme ve sıralandırabilmeyi hedef almaktadır. Tez çalışması içinde yer

alan kişisel tercihlerin sorgulanması bölümüne bu bakış açısının taşınması gerekmektedir.

Yukarıda belirtilen değerlendirmelerin doğrultusunda tez çalışmasının gelecekte ki hangi çalışmalara nasıl yön verebileceği aşağıda sunulmaktadır.

Literatürde, tercihlerin devingen olarak alan ontolojilerinin değişimine paralel değiştiği durumları örnekleyen uygulamalar bulunmamaktadır. Ancak profil yaklaşımının kullanıldığı ve kullanıcı profilleri ile ontolojiler üzerinde sorgulama yapılması durumlarının örneklendiği çalışmalar yer almakta, aynı profile sahip kullanıcıların kendi içlerinde gruplanabileceğini belirten araştırmalar öne çıkmaktadır. Kişisel tercihlerin yönetilebilmesi için ise kullanıcı profilleri ile entegre edilmiş uygulamalar yer almamaktadır. Kişisel bakış açısının günümüz uygulamalarında yer bulabilmesi için öncelikle tercihlerin yönetilebilir, genel bir model ile ifade edilmiş olması gerekmektedir. Tez çalışmasında yönetilebilirliğin sağlanabilmesi için bir çerçeve oluşturulması yaklaşımı benimsenmiştir. Bu sayede alan bilgisinden bağımsız ifade edilen bir üst modelin gerekli tercih tanımlamalarını içereceği gösterilmiştir. İleri ki bir çalışmada oluşturulan çözümün kullanıcı profilleri ile entegre edilebilmesi düşünülebilir.

Tercihlerin yönetilebilirliği için kişisel bilgileri saklayan yapılara (örn: FOAF dosyaları) ihtiyaç duyulduğu bilinmektedir. Bu nedenle tez çalışması kapsamında oluşturulan Tercih Yönetim Çerçevesinin FOAF dokümanlarına entegre edilmesi gerektiği soncuna varılmıştır. Literatürde ‘kişisel içerik yönetimi gerçekleştirim uygulamaları’ (FVCPR, 2008), ‘sayısal kütüphaneler’ (Hicks, 2001) için geliştirilen yaklaşımlar bulunmaktadır. Bu çalışmalar arama sonuçlarını düzenleyebilmek adına ontoloji temelli çözümleri tercih tanımlamalarından bağımsız olarak sunmaktadır. Tez çalışmasında ortaya konan bakış açısına yönelik tercihlerin modellenmesi ve modele bağımlı bir tercih operatörü ile sorgulanabilmesi fikrine yer verilmemektedir. Bu fikre yönelik bir araştırma için tez çalışmasına kaldığı yerden devam edilebilmesi gerekmektedir.

Oluşturulan Tercih Yönetim Çerçevesinin FOAF ontolojileri ile entegre edilmesi, bu sayede FOAF ontolojilerinin sahip olduğu üst-şema tanımının genişletilmesi ve kullanıcı tercihlerinin ilgili tercih tipi tanımları içinde saklanması düşünülmektedir.

Günümüzde elektronik ticaret uygulamaları için öneri sistemleri öne çıkan araştırma alanlarındadır. Kişisel tercihleri bilen akıllı bir sistem için kullanıcıya tercihlerini sormadan öneri sunabilen uygulamalara henüz rastlanmamaktadır. Oluşturulan Tercih Yönetim Çerçevesinin bu araştırma alanı içinde katkı sağlayabileceği düşünülmektedir. Bu nedenle 'kümeleme' algoritmalarının kullanılması ve bu sayede kullanıcının tercihine en yakın kümeye dahil edilerek ilgili tercihiye yönelik yeni önerileri görebilmesi mümkün olacaktır.

Kişiselleştirme çalışmaları içinde önemli araştırma konularından bir diğeri de zorunlu yasaklamaların ve izinlerin belirlenmesi için bilginin filtrelenmesidir. Bu konu için öncelikle kişisel tercihlerin seçilmesi, kişisel bilgi içinde saklanması ve çakışan durumların belirlenmesi gerekmektedir. Bu sayede kullanıcılar için sistem üzerinde politikalar belirlenmeli ve kişisel tercihlerin oluşturulmasında ve engellenmesinde bu konu ele alınmalıdır.

Son olarak unutulmamalıdır ki kişisel tercihler bilgi sistemleri üzerinde gizlilik konusu ile birlikte incelenmektedir. Bu nedenle bilgisayar güvenliği çalışmalarında, saldırı tespit sistemlerinde, gizlilik konusu gelecekte yapılabilecek olası çalışmalardan biri olarak düşünülebilir.

ÖZGEÇMİŞ

1975 yılında İzmir’de doğan Dilek TAPUCU, 1998 yılında Ege Üniversitesi Kimya Mühendisliği Bölümünden lisans derecesi ile mezun olduktan sonra, yüksek lisans eğitimine 1999 yılında İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümünde başlamıştır. Saldırı tespit sistemlerini konu alan tez çalışması ile Yüksek lisans derecesini tamamlayarak, 2003 yılında Ege Üniversitesi Bilgisayar Mühendisliği Bölümüne doktora öğrencisi olarak kabul edilmiştir. 2005 yılında Fransa Hükümetinin verdiği İkili Doktora bursunu kazanarak doktora çalışmasına Poitiers Üniversitesi ENSMA/LISI laboratuvarında eş zamanlı olarak devam etmiştir. Prof. Dr. Yamine AIT AMEUR ve Yrd.Doç.Dr Murat Osman ÜNALIR’ın eş yöneticiliğinde sürdürdüğü doktora tezi kapsamında çeşitli bilimsel dergilerde yayınlanan makalesi, ulusal ve uluslar arası bilimsel konferanslarda sunulan bildirileri ve bir kitap bölümü bulunmaktadır. Mesleki ilgi alanları arasında Anlamsal Web, Bilgi Yönetim Sistemleri, Kişiselleştirme yaklaşımları, Tercih Modelleri ve Tercih temelli sorgulama konuları gelmektedir. Halen İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümünde araştırma görevlisi olarak çalışmaktadır.

KAYNAKLAR DİZİNİ

- Adomavicius G. and Tuzhilin A.,** 2005. *Toward the next generation of recommender systems: a survey of the state of the art and possible extensions*, IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749.
- Agrawal, A.,** 2003, “GReAT: A Metamodel Based Model Transformation Language”, ASE 2003, ISBN:1-58113-751-6 pp. 386 - 387.
- Agarwal N., Haque E., Liu H., and Parsons L.** 2005 Research Paper *Recommender Systems: A Subspace Clustering Approach*. In International Conference on Web-Age Information Management (WAIM), pp. 475–491.
- Agrawal, R. and Wimmers, E. L.,** 2000. *A framework for expressing and combining preferences*. In *SIGMOD Conference*, pages 297–306.
- Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., and Tolle, K.,** 2001. *The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases*. In Proceedings of the 2nd International Workshop on the Semantic Web, pp. 1–13.
- Atkinson, C. and Kühne, T.,** 2003, “Model-Driven Development: A Metamodeling Foundation,” IEEE Software, vol. 20, no. 5, pp. 36–41.
- Berners-Lee, T., Hendler, J. and Lassila, O.,** 2001, “The Semantic Web”, Scientific American, 284(5), pp.34–43.
- Bezivin J. and Gerard S.,** 2002, A Preliminary Identification of MDA Components, OOPSLA 2002 Workshop.
- Bezivin J., Hammoudi S., Lopes D. and Jouault F.,** 2004, Applying MDA Approach for Web Services Platform, Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conference (EDOC 2004).
- Bellatreche L., Giacometti A., Marcel P., Mouloudi H. and Laurent D.,** 2005. *A Personalization Framework for OLAP Queries*. In DOLAP’05, ACM International Workshop on Data Warehousing and OLAP, pp. 9-18.

- Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., & Poole, D.** 2004. *Preference-based constrained optimization with CP-nets*. Computational Intelligence, 20(2), pp.137-157. (Special issue on preferences).
- Borzsonyi, S., Kossmann, D., and Stocker, K.,** 2001. *The skyline operator*. In *ICDE*, pp. 421–430.
- Bozsak, E., Ehrig,M., Handschuh, S., Hotho, A.,Maedche, A.,Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R., and Zacharias, V.** 2002. *KAON - Towards a Large Scale Semantic Web*. In Proceedings of the 3rd International Conference on E-Commerce and Web Technologies (*EC-WEB'02*), pp. 304–313,
- Braun, D., Sivils, J., Shapiro, A., Versteegh, J.,** 2001, “Unified Modeling Language Tutorial”, Kennesaw State University CSIS 4650 - Spring 2001. [Web Adresi] http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm [Eriřim Tarihi: 15.10.2009]
- Brickley and Miller, 2004** <http://xmlns.com/foaf/0.1/>
- Brickley D. and Guha R.V.,** 2000 *Resource Description Framework (RDF) Schema Specification 1.0*. Candidate recommendation, World Wide Web Consortium, March 2000. See <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>
- Brelade S., Harman C.,** 2003, *Pratical Guide to Knowledge Management*, GBR: Thorogood, pp. 90-95..
- Brewster, C., O’Hara, K., Fuller, S., Wilks, Y., Franconi, E., Musen, M. A., Ellman, J., and Shum, S. B.,** 2004. *Knowledge representation with ontologies: The present and future*. *IEEE Intelligent Systems*, 19(1):72–81.
- Broekstra, J., Kampman, A., and van Harmelen, F.,** 2002. *Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema*.
- Brockmans, S., Volz, R., Eberhart, A. and Löffler, P.,** 2004, “Visual modeling of OWL DL ontologies using *UML*,” Proceedings of the 3rd International Semantic Web Conference, Hiroshima, pp. 108–213.
- Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., and Wilkinson, K.,** 2004. *Jena: Implementing the Semantic Web Recommendations*. In Proceedings of the 13th international World Wide Web conference track papers & posters (WWW’04), pp. 74–83

- Chomicki, J.**, 2003. *Preference formulas in relations queries*. ACM Transactions on Database Systems, 28: pp. 1–39.
- Davies J., Fensel D., and Harmelen F.**, 2002. *Towards the Semantic Web: Ontology-Driven Knowledge Management*, chapter 10: Ontology Middleware and Reasoning. Wiley & Sons, Europe. See also <http://www.ontotext.com/omm>.
- Dehainsala, H., Pierra, G., and Bellatreche, L.** 2007. *OntoDB: An Ontology-Based Database for Data Intensive Applications*. In Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07), volume 4443 of Lecture Notes in Computer Science, pp. 497–508.
- Dehainsala, H., Pierra, G., Bellatreche, L., and Aït-Ameur, Y.** 2007. *Conception de bases de données à partir d'ontologies de domaine : Application aux bases de données du domaine technique*. In Actes des 1ère Journées Francophones sur les Ontologies (JFO'07), pp. 215–230.
- Deng L., Chai X., Tan Q, Ng W., and Lee D.**, 2004, *Spying Out Real User Preferences for Metasearch Engine Personalization*, In Proc. of WebKDD 2004: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 22-25 .
- Deshpande, M., Karypis, G.**, 2001. *Selective Markov Models for Predicting Web-Page Accesses*. In Proceedings of 1st SIAM International Conference on Data Mining (SDM'2001).
- Dimopoulos, Y., Moraitis, P. Tsoukias, A.**, 2006. *Qualitative Preference Modelling in Constraint Satisfaction*. Decision Theory and Multi-Agent Planning, G. Della Riccia, D. Dubois, R. Kruse and H.J. Lenz (Eds.), pp. 15-31, Springer Verlag, NewYork.
- Diederich J. and Iofciu T.**, 2006. *Finding communities of practice from user profiles based on folksonomies*. In Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, ISSN 1613-0073, p. 288-297.
- Djuric, D., Gašević, D. and Devedžić, V.**, 2005, “Ontology modeling and MDA”, Journal of Object Technology, vol. 4, no. 1, pp. 109–128.

- Doerr M., Hunter, J., and Lagoze, C.,** 2003. *Towards a Core Ontology for Information Integration*, Journal of Digital Information, Volume: 4, Issue:1.
- Dori., D.,** 2004. *Visweb - the visual semantic web: unifying human and machine knowledge representations with object-process methodology*. VLDB, 13(2) pp. 120–147, 2004.
- Dubois, D., Prade, H., and Sabbadin, R.,** 1998. *Qualitative decision theory with sugeno integrals*. In in: Proc. 14th Conf. on Uncertainty in Artificial Intelligence, pp. 121–128.
- Dupret, G., Murdock, V., and Piwowski, B.** (2007). *Web search engine evaluation using clickthrough data and a user model*. In WWW2007 workshop Query Log Analysis: Social and Technological Challenges.
- Falkovych, K., Sabou, M. and Stuckenschmidt, H.,** 2003, “UML for the Semantic Web: Transformation-based approaches”, in Knowledge Transformation for the Semantic Web, eds. B. Omelayenko & M. Klein, Frontiers in Artificial Intelligence and Applications, vol. 95, IOS, Amsterdam, pp. 92–106.
- Fensel D., Lausen H., Polleres A., Bruijn J., Stollberg M., Roman D. and Domingue J.,** 2007, Enabling Semantic Web Services, The Web Service Modeling Ontology, ISBN-10 3-540-34519-1, Springer-Verlag Berlin Heidelberg, pp. 188.
- Fischer S., Kießling W. and Preisinger T.,** 2006. Preference based Quality Assessment and Presentation of Query Results. Studies in Fuzziness and Soft Computing 978-3-540-33288-6. Springer pp. 91-121.
- Fischer, G.,** 2001. *User modeling in human-computer interaction*. In User Modeling and User-Adapted Interaction, pp. 65–68.
- Fisburn, P.C.,** 1970. *Utility Theory for Decision Making*, John Wiley & Sons, Inc., New York.
- Fishburn, P.,** 1988. *Nonlinear preference and utility theory*. Johns Hopkins University Press, Baltimore.
- Gasevic D., Djuric D. and Devedzic V.,** 2006, Model Driven Architecture and Ontology Development, Springer-Verlag Berlin Heidelberg, p. 311.

- Georges Dupret, B. P., 2008.** *A user browsing model to predict search engine click data from past observations.* In SIGIR 2008. ACM.
- Gruber, T., 1993.** *A translation approach to portable ontology specifications..* Knowledge Acquisition, pp. 161–199.
- Gruber, T. and Olsen, G., 1994.** *An ontology for engineering mathematics.* In Fourth Int'l Conf. Principles of Knowledge Representation and Reasoning, pp 258–269.
- Han J. and Kamber M., 2006.** *Data Warehouse and OLAP Technology: An Overview, Data Mining: Concepts and Techniques*, chap 3, pp. 105-155.
- Hansson, S. O., 2001.** *Preference logic.* In D. Gabbay & F. Guenther (Eds.), *Handbook of Philosophical Logic*, Vol. 4. pp. 319-393.
- Happel H., Seedorf S., 2006.** *Applications of Ontologies in Software Engineering*, In Proc. of International Workshop on Semantic Web Enabled Software Engineering, 2006.
- Hayes P., 2003.** *RDF Model Theory. Working Draft*, World Wide Web Consortium, January 23, 2003. See <http://www.w3.org/TR/rdf-mt/20030123/>.
- Hey, J., 2004,** *The Data, Information, Knowledge, Wisdom Chain: The Metaphorical Link*, Instructors: Geoff Nunberg, Paul Duguid, p.18.
- Hristidis, V., Koudas, N., and Papakonstantinou, Y. 2001.** *PREFER: A system for the efficient execution of multiparametric ranked queries.* In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM, New York, 259–270.
- ISO 10303-11. (2004).** Industrial automation systems and integration—Product data representation and exchange. *Pt 11: Description Methods: The EXPRESS language reference manual.*
- ISO 13584-25. (2004).** Industrial automation systems and integration—Parts library—Part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content. *ISO, Geneve.*
- Jean, S., Ait-Ameur, Y., Pierra, G., 2006:** *Querying Ontology Based Database Using OntoQL* (an Ontology Query Language). In: Proceedings of Ontologies, Databases, and Applications of Semantics (ODBASE'06), pp. 704-721.

- Jean, S., Ait-Ameur, Y., Pierra, G.**, 2007.: An Object-Oriented Based Algebra for Ontologies and their Instances. In: Proceedings of the 11th East European Conference in Advances in Databases and Information Systems (ADBIS'07). LCNS 4690, Springer, pp. 141-156.
- Jørstad, I. & Thanh, D.v.** 2006, *Service Personalisation in Mobile HeterogeneousEnvironmet'*, the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, pp. 70 - 76.
- Jrad, Z. and Aufaure, M.-A.**, 2007. *Personalized interfaces for a semantic web portal: Tourism information search*. In *KES (3)*, pp. 695–702.
- Ju-Yeon Kim, J.-W. K. and Kim, C.-S.**, 2007 *Ontology-Based User Preference Modeling for Enhancing Interoperability in Personalized Services*, Universal Access in Human-Computer Interaction. Applications and Services, Springer, Lecture Notes in Computer Science, 4556/2007, isbn: 978-3-540-73282-2, pp. 903-912.
- Kießling, W.**, 2002. *Foundations of preferences in database systems*. In Mars, N. J. I., editor, *Knowledge and Data Engineering*, pp. 311–322.
- Kießling, W. and Kostler, G.**, 2000. *PreferenceSQL Design, implementation, experience*. Knowledge and Data Engineering, pp. 778 – 789.
- Koivnen, M.R.** and Miller, E. 2001, *Semantic Web Kick-off Seminar*, Finland.
- Koutrika, G. and Ioannidis, Y. E.**, 2004. *Personalization of queries in database systems*. In *ICDE*, pp. 597–608.
- Lacroix, M. and Lavency, P.** , 1987. *Preferences; putting more knowledge into queries*. In Stocker, P. M., Kent, W., and Hammersley, P., editors, VLDB'87, Proceedings of 13th International Conference on Very Large Data Bases, pp. 217–225.
- Lin, Z.-H. and Fu, L.-C.** 2007. *Multi-user Preference Model and Service Provision in a Smart Home Environment*. In Automation Science and Engineering, 2007. IEEE International Conference (CASE 2007), pp. 759–764.
- Masseglia F., Poncelet P., Cicchetti R.**, 1999. *WebTool: An Integrated Framework for DataMining*, in Proceedings of the 9th International Conference on Database and Expert Systems Applications (DEXA'99), Florence, Italy, pp. 892-901.

- Manola, F., Miller, E., McBride, B., 2004:** RDF primer. Technical report, W3C Recommendation (2004) [w3.org/TR/2004/REC-rdf-primer-20040210/](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/).
- McGuinness, D. L., Harmelen F. v., 2004:** OWL Web Ontology Language. Technical report, W3C Recommendation (2004) [w3.org/TR/2004/REC-owl-features-20040210/](http://www.w3.org/TR/2004/REC-owl-features-20040210/).
- Mike D. and Schreiber G., 2004.** *Owlweb ontology language Reference*. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/2004>.
- McBride B., 2001.** *Jena: Implementing the RDFModel and Syntax Specification*. Proceedings of the 2nd International Workshop on the Semantic Web.
- Mobasher B., Cooley R., Srivastava J., 2000.** *Automatic Personalization Based on Web Usage Mining*, Communications of the ACM, August 2000/Vol. 43, No. 8, pp. 142-151.
- Mobasher, B., Dai, H., Luo, T., Nakagawa, M., (2001).** *Effective Personalization Based on Association Rule Discovery from Web Usage Data*. In Proceedings of 3rd ACM Workshop on Web Information and Data Management.
- Nasraoui, O., Frigui, H., Joshi, A., Krishnapuram, R., (1999).** *Mining Web Access Logs Using a Fuzzy Relational Clustering Algorithm Based on a Robust Estimator*, In Proceedings of the Eighth International Fuzzy Systems Association Congress.
- OMG CVM, 2003,** Common Warehouse Metamodel (CWM) Specification March 2003 Version 1.1, Volume 1 formal/03-03-02. [Web Adresi]. <http://www.omg.org/docs/formal/03-03-02.pdf> [Eriřim Tarihi: 18.02.2009]
- OMG ODM, 2008,** OMG Ontology Definition Metamodel (ODM) Specification v1.0 – Beta 3, OMG Document Number: ptc/2008-09-07. [Web Adresi]. <http://www.omg.org/docs/ptc/08-09-07.pdf> [Eriřim Tarihi: 11.02.2009].
- OMG ODM Adopted Specification, 2006,** Ontology Definition Metamodel OMG Adopted Specification ptc/06-10-11 (Final Adopted Specification), October 2006.
- OMG ODM RFP, 2005,** Ontology Definition Metamodel – Third Revised Submission to OMG RFP ad/2003-03-40, OMG Document ad/05-08-01. [Web Adresi]. <http://www.omg.org/docs/ad/05-08-01.pdf> [Eriřim Tarihi: 18.02.2009].

OMG MOF, 2002, OMG Meta Object Facility (MOF) Specification v1.4, OMG Document format/02-04-03. [Web Adresi].

<http://www.omg.org/docs/formal/02-04-03.pdf> [Erişim Tarihi: 05.02.2009].

OMG MOF2, 2003, OMG Meta Object Facility (MOF) 2.0 Core Specification Version 2.0: Final Adopted Specification, OMG Document ptc/03-10-04. [Web Adresi]. <http://www.omg.org/docs/ptc/03-10-04.pdf> [Erişim Tarihi: 05.02.2009].

OWL Web Ontology Language Overview, 2009, W3C Recommendation, 10 February 2004 reformatted on 03 March 2009. [Web Adresi]. <http://www.w3.org/TR/owl-features/> [Erişim Tarihi: 15.10.2009]

Pini et al., 2005 M. S. Pini, F. Rossi, and B. Venable. Possibility theory for reasoning about uncertain soft constraints. In *ECSQARU 2005*. Springer Verlag, pp. 800-811 .

RDF, Resource Description Framework, <http://www.w3c.org/RDF>, 2007

Shardanand U. and Maes P., 1995. *Social Information Filtering: Algorithms for Automating Word of Mouth*, In Proc.of CHI'95 Conference on Human Factors in Computing Systems, ACM Press, pp. 210-217.

Oztürk M. and Tsoukiàs A., 2005. *Preference on intervals: a general framework*. In Proceedings of the IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling, Edinburg, Scotland.

Öztürk, M., Tsoukiàs, A., and Vincke, P., 2005. *Preference modelling*. In Figueira, J., Greco, S., and Ehrgott, M., editors, Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 27–72.

Park, M. J., Lee, J. H., Lee, C. H., Lin, J., Serres, O., and Chung, C.W., (2007). *An Efficient and Scalable Management of Ontology*. In Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07), volume 4443 of Lecture Notes in Computer Science, pp. 975–980.

Pierra, G., Dehainsala, H., Aït-Ameur, Y., and Bellatreche, L., 2005. *Base de Données à Base Ontologique : principes et mise en oeuvre*. Ingénierie des Systèmes d'Information, 10(2):91–115.

- Prassas, G.; Pramataris, K.,** 2001. *Dynamic recommendations in internet retailing*. In ECIS 2001, pp. 27–28.
- Pitkow, J., Pirolli, P.,** 1999. *Mining Longest Repeating Subsequences to Predict World Wide Web Surfing*. In Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS'99).
- Razmerita L., Angehrn A., Maedche A.,** 2003. *Ontology based user modeling for Knowledge Management Systems*, Proceedings of the User Modeling Conference, Pittsburgh, USA, Springer Verlag, pp. 213-217.
- Sarukkai, R. R.,** 2000. *Link Prediction and Path Analysis Using Markov Chains*. In Proceedings of 9th International World Wide Web Conference, pp. 377-386.
- Selic, B.,** 2003, “The pragmatics of model-driven development,” IEEE Software, vol. 20, no. 5, pp. 19–25.
- Schenk, D. and Wilson, P.,** 1994. *Information Modelling The EXPRESS Way*. Oxford University Press, ISBN10: 0195087143.
- Siberski, W., Pan, J., and Thaden, U.,** 2006. *Querying the semantic web with preferences*. ISWC 2006, pages 612–624.
- Shardanand, P. M.,** 1995. *Social information filtering: Algorithms for automating 'word of mouth'*. In In Proceedings of the Conference on Human Factors in Computing Systems (CHI95), pp. 210–217.
- Suh Y., Park Y., Yoon H., Chang Y., Woo W.,** 2007. *Context-Aware Mobile AR System for Personalization, Selective Sharing, and Interaction of Contents in Ubiquitous Computing Environments*. HCI pp. 966-974.
- Susan, J. T. and Gauch, S.,** 2003. *Improving ontology-based user profiles*. In University of Avignon, pp. 380–389.
- Theobald, M., Weikum, G., and Schenkel, R.,** 2004. *Top-k query evaluation with probabilistic guarantees*. In VLDB, pp. 648–659.
- Trajkova J., Gauch S.,** 2004. *Improving Ontology-based User Profiles*, Proc. of RIAO 2004, University of Avignon (Vaucluse), France, April 26-28, 2004, pp. 380-389.
- Vallet, D., Castells, P., Fernandez, M., Mylonas, P., and Avrithis, Y.,** 2007. *Personalized content retrieval in context using ontological knowledge*.

Circuits and Systems for Video Technology, IEEE Transactions on, 17(3), pp.336–346.

Verna, A., 1997. *The Knowledge Evolution: Expanding Organizational Intelligence*. Boston: Butterworth-Heinemann, ISBN: 0-7506-9842-X..

Viappiani, P., Faltings, B., and Pu, P., 2006. *Preference-based search using example-critiquing with suggestions*, Journal of Artificial Intelligence Research (JAIR), 27, 2006, pp. 465-503.

Vlachou, A, Doukeridis, C., Kotidis, Y., and Vazirgiannis, M., 2007. *Skypeer:Efficient subspace skyline computation over distributed data*. In Data Engineering, ICDE 2007, pp. 416-425.

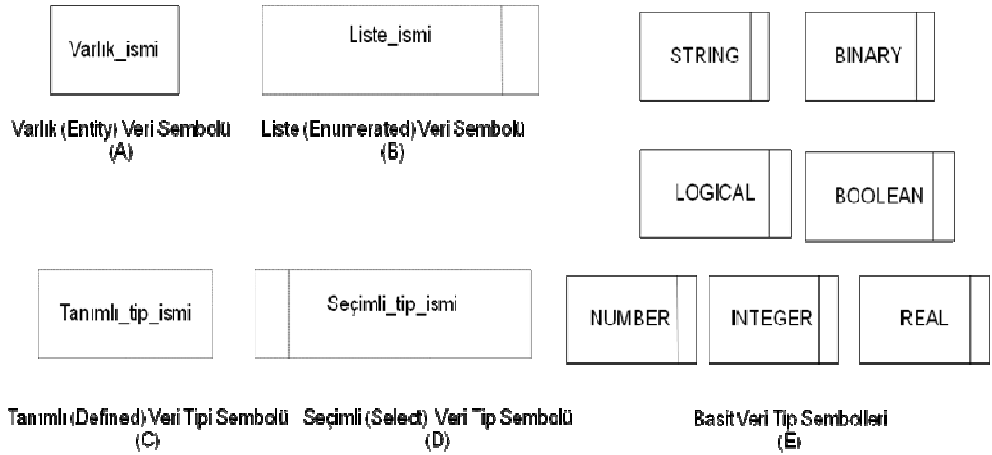
9 EKLER

EK-1: EXPRESS Veri Modelleme Dili

EXPRESS, (Schenk and Wilson, 1994), 1980 lerin sonunda ISO-STEP (ISO10303, 1994) projesinin bir parçası olarak geliştirilmeye başlanan bir “veri modelleme” dilidir. Anılan dönemde formal olarak veri yapısını temsil eden bir modelin bulunmaması STEP (Standard for the Exchange of Product model data) projesinin başlama nedeni olmuştur. Bu proje ile işletmelerde kullanılan farklı ürünler için veri modelleri standardizasyonu sağlamaktadır. Proje kapsamında ISO standartlarına uygun olarak EXPRESS dili geliştirilmiştir

EXPRESS-G Grafik Notasyonu

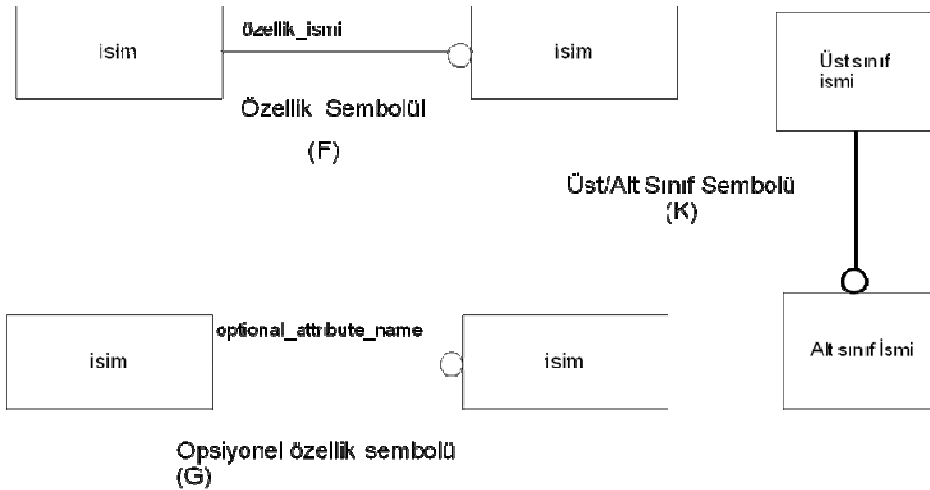
EXPRESS-G, EXPRESS dili için standart bir grafik notasyonu sağlamaktadır. Belirtilen notasyon varlıklara ilişkin tanımları, ilişkileri ve kısıtlamaları göstermekte, EXPRESS dili ile tanımlanmış bir veri modeli yapısı daha anlaşılır biçimde sunulabilmektedir.



- **Varlık Veri Sembolü:** EXPRESS dilinin en önemli kavramıdır. Kavramsal alan bilgisini yada gerçek dünya nesnelerini modellemek üzere kullanılan veri tipi olarak açıklanmaktadır. Grafiksel gösterimi yukarıda

yer alan Şekilde sunulduğu gibi dikdörtgen bir kutudur, ilgili varlığın ismi kutunun içine yazılmaktadır.

- **Seçimli Veri Tipi:** Şekilde gösterildiği gibi farklı opsiyonlar arasında yapılacak olası seçimi göstermek için kullanılmaktadır.
- **Liste Veri Tipi:** Belirlenen kısıta uygun olan veri tipleri koleksiyonunu göstermek için kullanılmaktadır.
- **Öznitelik:** Tanımlanan bir varlığın sahip olduğu karakteristik özellikleridir. Her özellik bir veri tipi ile ifade edilmektedir ve tanımlandığı tipe uygun aralıkta değer almaktadır.



- **İlişkiler:** Yukarıda yer alan şekilde gösterildiği gibi varlıkların birbirleri ile ilişkilendirilmesi için kullanılmaktadır. Her ilişki isim ve kısıt özelliklerine sahiptir. Farklı ilişkilerin gösterimi aynı şekilde belirtilmektedir. Bunlar arasında yer alan ters ilişki tanımı varlıklar arasında kurulan ilişki tanımının tersini göstermek amacı ile (INV) kısaltması ile birlikte kullanılmaktadır.
- **Alt - Üst Sınıf Gösterimi:** Alt tip yada üst tip tarifleri kalıtım özelliği kullanılarak yapılmaktadır.

- **Kurallar ve Kısıtlar:** Varlıklar ve tanımlanmış veri tiplerinin WHERE kuralı kullanılarak tanımlanan koşul değerlerine uygun aralıkta değer alabilmesi için kullanılmaktadır. WHERE kuralı global tanımlı bir kuraldır.

EXPRESS dili, sistemler arası veri değişim forumlarını, dil modelleme bilgisini (ISO10303, 1994) ve ilişkili üst-şemaleme altyapısını içermektedir. Dilin standart erişim arayüzü, kullanıcıların veriye erişebilmelerine ve aynı anda veriyi işlemelerini sağlamaktadır. Bu arayüzün geliştiriminde üst modelleme tekniği kullanılmıştır. Bu nedenle dilin özellikleri bir yazılım diliden bağımsız olarak yapılandırılmıştır. Belirtilen ara yüz gerçekleştirimi için ise C, C++ ve Java programlama dilleri kullanılmıştır. EXPRESS dili veri tipleri yapısal ve genel olarak ikiye ayrılmaktadır. Yapısal veri tipleri EXPRESS şema dokümanı içinde tanımlanmaktadır. Bunlar varlık tanımı, varlığa ilişkin özellik tanımlamaları şeklindedir. Genel veri tipleri ise prosedür, fonksiyon ve soyut varlıkların tanımlarını kapsamaktadır. Aşağıda FAMILIY şema dokümanı oluşturulan EXPRESS kod parçacığı, ve örnekleri sunulmaktadır.

```

SCHEMA Family;
  ENTITY Person
    ABSTRACT SUPERTYPE OF (ONEOF (Male, Female));
    id: NUBER;
    name: STRING;
    age: NUMBER;
    mother: OPTIONAL Female;
    father: OPTIONAL Male;
  END_ENTITY;

  ENTITY Female
    SUBTYPE OF (Person);
  END_ENTITY;

  ENTITY Male
    SUBTYPE of (Person);
  END_ENTITY;
END_SCHEMA;

```

EXPRESS dili ile oluşturulan şema dokümanları örneklendirilmektedir. Belirtilen herbir örnek aşağıda gösterildiği gibi bir ID ye sahiptir ve şema üzerinde tanımlı özniteliklerine atanan değerler bütünü temsil etmektedir. Burada vurgulanması gereken nokta herbir örneği # işareti ile başlaması gerektiğidir.

```
#501 = PERSON (1345, 'Deniz KOM', 28);  
#502 = PERSON (2468 , 'Ali FANKAM', 36);  
#503 = PERSON (7562 , 'Demir FANKAM', 18);
```

EK-2 OntoQL Tanımlama Dili

SQL dilinin bir uzantısı olarak yaratılan OntoQL (Jean et, all., 2006) dili OntoDB mimarisi (Dehainsala et. all., 2007) için veri/ontoloji tanımlama ve sorgulama dili olarak kullanılmaktadır. OntoQL dili sahip olduğu **ontoloji tanımlama** ve **düzenleme** dili yetenekleri ile OntoDB mimarisinin üst şema bölümünde değişiklik yapabilmeyi olanaklı kılmaktadır

Veri Tanımlama Dili: Bir veritabanı veya veritabanı tablosunu oluşturmayı, silmeyi ve üzerinde değişiklik yapmayı sağlamaya yarayan komutları içermektedir. Bunlar tıpkı SQL dilinde olduğu gibi CREATE ve ALTER komutlarıdır.

- **CREATE komutu** veritabanı, sınıf ve tablo oluşturmak ve kullanıcıların yeni bir tip ve bu tipe ait özniteliklerin yaratmasını sağlamak amacı ile kullanılmaktadır. CREATE komutu sözdizimi aşağıda yer almaktadır.

```
<class definition> ::= CREATE <entity id> <class id>
[<under clause>] [<descriptor clause>] [<properties clause list> ]
<under clause> ::= UNDER <class id list>
<descriptor clause> ::= DESCRIPTOR ( <attribute value list> )
<attribute value> ::= <attribute id> = <value expression>
<properties clause> ::= <entity id> (<property definition list> )
<property definition> ::= <prop id> <datatype> [<descriptor
clause>]
```

- **CREATE EXTENTION OF komutu** OntoQL diline özgü olarak OntoDB mimarisi üzerinde oluşturulacak bir sınıf tanımını tamamlamak için kullanılmaktadır. Belirtilen komutta yaratılan her sınıfın sahip olduğu özniteliklerin birarada gösterilmesi gerekmektedir.

```
<extension definition> ::= CREATE EXTENT OF <class id>(
<property id list>) [<logical clause>]
<logical clause> ::= TABLE [<table and column name>]
<table and column name> ::= <table name> [<column name list>]
```

- **ALTER komutu** veritabanında var olan nesneler üzerinde yapılacak değişiklikler için kullanılan komuttur. Yaşayan bir veritabanında değişiklik yapmak için oldukça kolaylık sağlamaktadır. Bu komuda ilişkin sözdizimi aşağıda sunulmaktadır.

```
<alter class statement> ::= ALTER <class id>
[<descriptor clause>] [<alter class action>]
<alter class action> ::= <add property definition> |
<drop property definition>
<add property definition> ::= ADD [<entity id>]
<property definition> [<descriptor clause>]
<drop property definition> ::= DROP <property id>
```

Ontoloji Düzenleme Dili: CREATE komudu kullanılarak oluşturulmuş sınıflar üzerinde değişiklik yapılmasını sağlamak amacı ile kullanılan INSERT, UPDATE ve DELETE komutlarından oluşmaktadır.

- **INSERT komutu** bir tabloya yeni bir verinin eklenmesini sağlamaktadır.

```
<insert statement> ::= INSERT INTO <class id>
<insert description and source>
<insert description and source> ::= <from subquery> | <from
constructor>
<from subquery> ::= [( <property id list>)] <query expression>
<from constructor> ::= [(<property id list>) ] <values clause>
<values clause> ::= VALUES (<values expression list>)
```

- **UPDATE komutu** bir tablodaki veri alanlarına ait değerlerin belli bir koşula bağlı olarak ya da olmadan değiştirilmesi için kullanılmaktadır.

```
<update statement> ::= UPDATE <class id polymorph>
SET <set clause list>
[WHERE <search condition> ]
<class id polymorph> ::= <class id> | ONLY (<class id>)
<set clause> ::= <property id> = <value expression>
```

- **DELETE komutu** bir tablodaki verilerin tamamının ya da bir kısmını silmektedir.

```
<delete statement> ::= DELETE FROM <class id polymorph>
```

```
[ WHERE <search condition> ]
```

Ayrıca OntoQL dili bir diğer yanı ile sorgulama gerçekleştirimini ağırlamaktadır. Sorgular mimari üzerinde “ontolojiden içeriğe”, “içerikden ontolojiye” ya da “üst-şemaden ontolojiye” doğru gerçekleştirebilmektedir.

```
<query specification> ::= <select clause>
<from clause> [ <where clause>]
[<group by clause>] [ <having clause>] [<order by clause>]
[<namespace clause> ] [ <language clause> ]
```

OntoQL Sorgulama Dili: SQL dilini temel yeteneklerini içerecek şekilde SELECT , FROM WHERE yapısını kullanmaktadır. Sahip olduğu söz dizimi aşağıda gösterilmektedir.

```
<query specification> ::= <select clause>
<from clause> [ <where clause> ]
[ <group by clause>] [ <having clause> ] [ <order by clause> ]
[ <namespace clause> ] [ <language clause> ]
```

EK-3 EXPRES Şema Dokümanı

```
PREFERENCE_MODEL;
```

```
ENTITY PREFERENCE
```

```
  ABSTRACT SUPERTYPE OF (ONEOF (PREFERENCE_DEFINITION,  
    CONTEXT_PREFERENCE_DEFINITION) );
```

```
END_ENTITY;
```

```
ENTITY PREFERENCE_DEFINITION
```

```
  ABSTRACT SUPERTYPE OF (ONEOF  
    (UNINTERPRETED_PREFERENCE, INTERPRETED_PREFERENCE) );
```

```
END_ENTITY;
```

```
ENTITY PREFERENCE_URI;
```

```
  CODE: INTEGER;
```

```
  NAME: STRING;
```

```
  CLASSIFICATION: STRING;
```

```
END_ENTITY;
```

```
ENTITY INTERPRETED_PREFERENCE
```

```
  ABSTRACT SUPERTYPE OF (ONEOF  
    (BOOLEAN_PREFERENCE, ENUMERATED_PREFERENCE, NUMERIC_PREF  
      ERENCE, INTERVAL_PREFERENCE, FUZZY_PREFERENCE) );
```

```
END_ENTITY;
```

```
ENTITY ENUMERATED_PREFERENCE
```

```
  SUBTYPE OF (INTERPRETED_PREFERENCE);
```

```
    PREF_VALUES: LIST[1:?] OF
```

```
    PROPERTY_OR_CLASS_INSTANCE;
```

```
    PREF_ATTRIBUTES: LIST[1:?] OF PREFERENCE_URI;
```

```
END_ENTITY;
```

```
ENTITY NUMERIC_PREFERENCE
```

```
  SUBTYPE OF (INTERPRETED_PREFERENCE);
```

```
    INTERPRETED_BY: NUMBER_VALUE;
```

```

    PREF_ATTRIBUTES: LIST[1:?] OF PREFERENCE_URI;
END_ENTITY;

```

```

TYPE NUMBER_VALUE= NUMBER;
END_TYPE;

```

```

ENTITY BOOLEAN_PREFERENCE
  SUBTYPE OF (INTERPRETED_PREFERENCE);
  INTERPRETED_BY: LIST[1:?] OF PROPERTY_VALUE;
  PREF_ATTRIBUTES: LIST[1:?] OF PREFERENCE_URI;
END_ENTITY;

```

```

ENTITY INTERVAL_PREFERENCE
  SUBTYPE OF (INTERPRETED_PREFERENCE);
  INTERPRETED_BY: INTERVAL_VALUE;
  PREF_ATTRIBUTES: LIST [1:?] OF PREFERENCE_URI;
END_ENTITY;

```

```

ENTITY INTERVAL_VALUE;
  min_value: REAL;
  max_value: REAL;
  WHERE min_value < max_value;
END_ENTITY;

```

```

ENTITY FUZZY_PREFERENCE
  SUBTYPE OF (INTERPRETED_PREFERENCE);
  INTERPRETED_BY: PROB_VALUE;
  PREF_VALUES: LIST[1:?] OF PROPERTY_VALUE;
  PREF_ATTRIBUTES: LIST[1:?] OF PREFERENCE_URI;
END_ENTITY;

```

```

TYPE PROB_VALUE= REAL;
  WHERE ((SELF>0) AND (SELF<1));
END_TYPE;

```

```

ENTITY UNINTERPRETED\_PREFERENCE
  SUBTYPE OF (PREFERENCE_DEFINITION);

```

```

    INTERPRETED_BY: LIST[1:?] OF
    PROPERTY_OR_CLASS_INSTANCE;
    PREF_ATTRIBUTES: LIST[1:?] OF PREFERENCE_URI;
END_ENTITY;

```

```

ENTITY CONTEXT_PREFERENCE_DEFINITION
    SUBTYPE OF (PREFERENCE);
    CONTEXT_VALUE: PROPERTY_OR_CLASS_INSTANCE;
    PREFERENCE: PREFERENCE_DEFINITION;
END_ENTITY;

```

```

TYPE PROPERTY_OR_CLASS_INSTANCE=
    SELECT (CLASS_VALUE, PROPERTY_VALUE);
END_TYPE;

```

```

TYPE CLASS_VALUE= STRING;
END_TYPE;

```

```

TYPE PROPERTY_VALUE= STRING;
END_TYPE;

```

```

END_SCHEMA;

```

EK-4: Tercih Yönetim Çerçevesi Elemanları için UML Şema

