

**WEB SAYFALARI İÇİN ANLAMSAL ERİŞİM SİSTEMİ**

**A SEMANTIC RETRIEVAL SYSTEM FOR WEB PAGES**

**EBRU SEZER**

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

BİLGİSAYAR Mühendisliği Anabilim Dalı İçin Öngördüğü

DOKTORA tezi

olarak hazırlanmıştır.

2006

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **Bilgisayar Mühendisliği ANABİLİM DALI** 'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Başkan : Prof Dr. Ersin Töreci

Üye(Danışman) : Prof. Dr. Adnan Yazıcı

Üye (Danışman) : Prof. Dr. Ünal Yarımağan

Üye : Yrd. Doç. Dr. Harun Artuner

Üye : Yrd. Doç. Dr. Kayhan İmre

## **ONAY**

Bu tez, ..... /..... / ..... tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

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

*Eşime,*

# WEB SAYFALARI İÇİN ANLAMSAL BİLGİ ERİŞİM SİSTEMİ

Ebru Sezer

## ÖZ

Günümüzde internetin içerdiği web sayfalarının çeşitliliği ve sayısı sebebi ile en geniş çoklu ortam derlemi olduğu söylenebilir. Veri türünün böylesine çeşitlendiği ve hacminin büyük olduğu bir ortamda ihtiyaca karşılık gelen bilgiye erişim; anlamsal erişim, içerik tabanlı erişim, nesne tanıma ve etiketlendirme, internet teknolojileri gibi alanlar için halen açık bir problemdir. Geliştirilen tez, kullandığı standartlar ve metodoloji uyumu sebebi ile Anlamsal Web konu başlığı altında yer almaktadır. Çalışmada tanımlanan problem “farklı veri türlerinin aynı sorgu cümlecığı ile sorgulanabilmesi” olarak özetlenebilir. Bu sorgulama biçiminin gerekliliği, farklı veri türlerinin kullanılma gerekliliğinden kaynaklanır. Yani farklı veri türleri bir anlamı vurgulamak ya da anlamı zenginleştirmek için kullanılır. Başka bir deyişle sayfanın anlamsal özelliklerini yatayda ya da dikeyde genişletirler. Bu katkının ele alınabilmesi için farklı veri türleri benzer biçimde dizinlenmeli ve sorgulanmalıdır.

Tezde seçilen veri türleri metin ve görüntülerdir. Bu seçimin sebebi, her iki veri türünün daha sık birlikte kullanımı ve birbirini tamamlayabilme ilişkileridir. Dizinlemede öncelikle içerik özellikleri çıkarılır. Metinler için Vektör Uzayı Modeli kullanılır. Görüntülerde nesnelere etiketlenir ve konumsal ilişkiler otomatik çıkarılır. Üretilen içerik özellikleri ve bulanık alan ontolojisi anlamsal özelliklerin çıkarımında kullanılır. Sayfanın anlamsal ve içerik özellikleri sayfa üst verisinde saklanır. Kullanıcı sorguları bu üst veriler üzerinde işlenir.

Bulanık ontoloji, önerilen NEK(Nesne, Eylem, Kavram) Modeli'ne göre araç kullanılarak geliştirilir. Bu öneri ile bir alanın, üç farklı katman aracılığı ile somut-soyut ya da varlık-anlam aralığında modellenebilmesi amaçlanmıştır. Kullanılan ontoloji gösterim dili Web Ontology Language (OWL) ve sorgulama dili RDF Data Query Language (RDQL)'dir.

Sistemin başarımı biri gerçek ve diğeri sentetik olmak üzere toplam 2 ayrı deney kümesi üzerinde anma (recall), duyarlık (precision) ve  $R_{norm}$  parametreleri ile

ölçülmüştür. Sistemin anma ve  $R_{norm}$  değerlerinde olumlu katkısı gözlenmiştir. Bu gözlem ile farklı veri türlerinin, sayfanın anlamsallığını zenginleştirdiği düşüncesi desteklenmiştir.

Anahtar kelimeler: Bulanık Ontoloji, Anlamsal Erişim, İçerik Tabanlı Erişim, OWL, RDF, RDFS.

Danışman: Prof. Dr. Adnan Yazıcı, Orta Doğu Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü.

Danışman : Prof. Dr. Ünal Yarımağan, Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü.

# **A SEMANTIC INFORMATION RETRIEVAL SYSTEM FOR WEB PAGES**

**Ebru Sezer**

## **ABSTRACT**

Information retrieval from web pages is still open problem for different topics: content based retrieval, semantic retrieval, object recognition, internet technologies...etc. This thesis is placed under semantic retrieval topic for its accordance to semantic web standards and methodologies. It represents a model for semantic retrieval of web pages. The problem of the model can be summarised as “querying different data types by using same query sentence”. The necessity of this querying style comes from necessity of different data types usage. This means, different data types are used to emphasize some meanings or to increase richness of the meaning. In other words, it enables vertical or horizontal expansions on the web page semantics. This semantic contribution can be handled by indexing and querying different data types in the same style.

In this study, text and images are selected as different data types. Because their together usage are more frequently encountered and they have semantic complementary relationship between each other. In the indexing process, firstly, content features of each data types are extracted. Vector space model is used for texts. Manual object annotation and automatic spatial relation extraction are used for images. Produced content features and a fuzzy domain ontology are used as input for semantic feature extraction process. This means, semantics of the web page is union of semantics of text and images. Produced semantic features are stored in the meta data of the web page with its content features. These meta datas can be queried by user.

Fuzzy domain ontology is developed by tool according to suggested ontology model called OAC(Object, Action, Concept) Model. The aim of the model is to contain all terms between concrete and abstract or entity and meaning interval. Ontology Web Language (OWL) is used to represent domain ontology and RDF Data Query Language (RDQL ) is used to query meta datas.

Performance of the system is measured by recall, precision and  $R_{norm}$  parameters on two different experiment sets like real data and synthetic data. Positive contributions are observed on recall and  $R_{norm}$  parameters. This observation supports that usage of different data types provides to increase richness or to emphasize semantics of the web page.

Keywords: Fuzzy ontology, Semantic Web & Retrieval, Content based retrieval, OWL, RDF,RDFS

Advisor: Prof. Dr. Adnan Yazıcı, Middle East Technical University , Department of Computer Engineering.

Advisor: Prof. Dr. Ünal Yarımağan, Hacettepe University, Department of Computer Engineering.

## TEŞEKKÜR

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

Sayın Prof. Dr. Adnan Yazıcı, yazara kendisi ile çalışma fırsatı vererek, tez geliştirme sürecinin tamamında değerli fikir ve tecrübeleri ile yön gösterici olmuşlardır. Tez geliştirme sürecinde yer alan literatür taraması, tez konusunun ve probleminin tanımlanması, tezde geliştirilen çözümün oluşturulması ve gerçekleştirimi, uygulama alanı seçimi ve modellenmesi, tez raporunun şekillenmesi gibi çalışma adımlarının tamamında yönetici ve danışman olarak yer almışlar ve yazara çok büyük katkı ve bilgi sağlamışlardır.

Sayın Prof. Dr. Ünal Yarımağan, yazara kendisi ile çalışma fırsatı vererek, tez sürecinin değerlendirilmesinde, gelişiminde değerli fikir ve tecrübeleri ile katkıda bulunmuş ve yön gösterici olmuşlardır.

Sayın Yrd. Doç. Dr. Harun Artuner ve Sayın Yrd. Doç. Dr. Kayhan İmre; tez gelişimine ait fikirlerini ve yorumlarını sunmuşlardır.

Sayın Dr. Kemal Arda (Ufuk Üniversitesi Tıp Fakültesi) sistemin deney ortamının hazırlanmasında, deney sürecinin devamında ve gelecek çalışmaların planlamasında mesleki bilgi ve tecrübeleri ile yazara yardımcı olup, önerileri ile yol gösterici olmuşlardır.

Sayın Radyoloji Asistanı Nazan Çiledağ, deney verilerinin işlenmesinde, abdomen ontolojinin üretiminde mesleki bilgileri ile yazara yardımcı olmuşlardır.

Öncelikle Sayın Prof. Dr. Ersin Töreci olmak üzere tüm Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü personeli manevi desteklerini sunmuşlardır.



# İÇİNDEKİLER DİZİNİ

Öz.....	i
Abstract.....	iii
Teşekkür.....	v
İçindekiler Dizini.....	vi
Şekiller Dizini.....	viii
Çizelgeler Dizini.....	x
Ekler Dizini.....	xi
Simgeler ve Kısaltmalar Dizini.....	xii
Genel Bakış.....	1
Bölüm 1. GİRİŞ.....	4
1.1. Problem Tanımı.....	4
1.2. Problemin Seçim Nedeni ve Çözümünden Beklenen Katkılar.....	9
Bölüm 2. İÇERİK TABANLI GÖRÜNTÜ ERIŞİMİ VE İLGİLİ ÇALIŞMALAR.....	12
2.1. Resim Kütüphaneleri için Anlamsal – Duyarlı Tümlleşik Eşleşirme.....	15
2.2. İçerik Tabanlı Görüntü Erişimi İçin Alan Tabanlı Bulanık Eşleşirme Yaklaşımı.....	19
2.3. Görüntü Sınıflandırma için Kümeleme Kullanarak Otomatik Ontoloji Türetimi.....	22
2.4. Resimlerin ve Kelimelerin Anlamlarının Öğrenilmesi.....	25
2.5. Kümelemeyle İçerik Tabanlı Görüntü Erişimi.....	27
2.6. İçerik Tabanlı Görüntü Erişim Sistemleri İçin Akıllı Karma Yaklaşım.....	29
2.7. İçerik Tabanlı Erişimi Destekleyen Anlamsal Şablon Uygulaması.....	30
2.8. Web Üzerinde İçerik Tabanlı Erişim İçin Çizim İle Sorgulama ve Geri Bildirim.....	33
Bölüm 3. ANLAMSAL WEB ve ONTOLOJİ.....	36
3.1. Anlamsal Web.....	36
3.2. Ontoloji.....	42
3.2.1. Ontoloji Tanımı.....	43
3.2.2. Ontoloji Türleri.....	46
3.2.3. Ontoloji İle İlgili Çalışmalar.....	50
3.3. Anlamsal Web Dilleri ile Ontoloji Gösterimi, Sorgulama.....	57
3.3.1. RDF Schema (RDFS).....	57

3.3.2. Web Ontology Language (OWL).....	60
3.3.3. RDF Data Query Languages (RDQL).....	65
3.4. Ontoloji ve Biçimsel Diller.....	66
3.5. Ontoloji ve Unified Modelling Language (UML).....	71
3.6. Ontoloji ve ERW (Entity and Relations on the Web).....	74
Bölüm 4. WEB SAYFALARI İÇİN ANLAMSAL ERİŞİM SİSTEMİ.....	77
4.1. Sisteme Genel Bakış.....	77
4.2. Örnek Uygulama Alanlarının Tanıtımı ve Bulanıklığa Giriş.....	79
4.3. NEK (Nesne, Eylem, Kavram) Ontoloji Modeli.....	85
4.4. Web Sayfası İçerik Özellikleri Çıkarma.....	104
4.5. Web Sayfası Anlamsal Özellikleri Çıkarımı.....	111
4.5.1. Web Sayfası Üst Veri Modeli.....	111
4.5.2. Alan Ontolojisi İle Üst Veri'nin Anlamsal Genişletimi.....	113
4.6. Sorgulama.....	117
4.7. Sistem Mimarisi.....	119
BÖLÜM 5. SİSTEM BAŞARIMI DENEYLERİ VE SONUÇLARI.....	121
5.1. Başarım Deneylerine Ait Ortamların Tanıtımı.....	123
5.2. Başarım Deneyleri Sonuçları.....	125
5.2.1. Deney 1'e Ait Başarım Sonuçları.....	125
5.2.2. Deney 2'ye Ait Başarım Sonuçları.....	132
5.2.3. Deneyler Arası Başarımın Karşılaştırılması.....	138
BÖLÜM 6. GENEL SONUÇ VE ÖNERİLER.....	139
KAYNAKLAR.....	142

## ŞEKİLLER DİZİNİ

Şekil 1.1: Tipik bir içerik tabanlı görüntü erişim sistemi ortamı.....	6
Şekil 2.1: SimplyCity’de sorgu işleme iş adımları.....	17
Şekil 2.2: Khan ve Wang ‘ın çalışmalarının genel sistem akışı.....	24
Şekil 2.3: Barnard’ın sisteminin kullandığı sıradüzensel ağaç görünümü.....	26
Şekil 2.4: CBICR’nin genel sistem görünümü.....	27
Şekil 2.5: HACBIR’ın genel görünümü.....	29
Şekil 2.6: STSCBIR sistemin genel mimarisi.....	31
Şekil 3.1: Anlamsal Web’in katmanlı kek görünümü.....	38
Şekil 3.2: Anlamsal Web uygulama görünümü.....	38
Şekil 3.3: RDF Deyiminin çizge görünümü.....	41
Şekil 3.4: Anlamsal Web bilgi yönetim mimarisi.....	44
Şekil 3.5: Örnek keskin ontoloji: Seyahat.....	51
Şekil 3.6 : Bulanık ontoloji modeli.....	55
Şekil 3.7: Şekil 3.6’daki model ile meteoroloji ontolojisi gösterimi.....	56
Şekil 3.8: RDF/RDFS ile modelleme örneği.....	58
Şekil 3.9: OWL ile RDF/RDFS arası kalıtım sıradüzeni.....	61
Şekil 3.10: OntoBroker mimarisi.....	69
Şekil 3.11: OntoBroker için ontoloji kesiti.....	70
Şekil 3.12: OntoBroker için örnek HTML sayfası.....	70
Şekil 3.13.a: Basit bit ER çizeneği.....	75
Şekil 3.13.b: Şekil 3.13.a ile uyumlu ERL kütüğü.....	75
Şekil 4.1: Anlamsal erişim sistemi araçları, modülleri ve ilişkileri.....	78
Şekil 4.2: Sistemin bilgi yönetim mimarisi.....	78
Şekil 4.3: Abdomene ait üç farklı CT görüntüsü.....	81
Şekil 4.4: Karaciğere ait iki farklı CT görüntüsü.....	82
Şekil 4.5: Smaç anına ait örnek görüntüler.....	84
Şekil 4.6: NEK Ontoloji modeli.....	88
Şekil 4.7.a: NEK Ontoloji modelinin bulanık sınıf ve bağıntıları: “voleybol”.....	95
Şekil 4.7.b: NEK Ontoloji modelinin bulanık sınıf ve bağıntıları: “abdomen”.....	96
Şekil 4.8.a : Genişletilmiş “voleybol” ontoloji kesiti.....	102
Şekil 4.8.b : Genişletilmiş “abdomen” ontoloji kesiti.....	103
Şekil 4.9: Web Sayfası içerik özellikleri çıkarma aşamaları.....	105

Şekil 4.10: $A_i$ ve $A_r$ alanları konumlanış örnekleri: Topolojik.....	108
Şekil 4.11: $A_i$ ve $A_r$ alanları konumlanış örnekleri: Uzaklık.....	108
Şekil 4.12: $A_i$ ve $A_r$ alanları konumlanış örnekleri: Yön (orta nokta).....	110
Şekil 4.13: $A_i$ ve $A_r$ alanları konumlanış örnekleri: Yön (bölümleme).....	110
Şekil 4.14: Web sayfası üst veri modeli.....	112
Şekil 4.15: Üst veri ve alan ontolojisi ilişkisi.....	113
Şekil 4.16: NEK Ontolojisinde kullanılan konumsal ilişkiler.....	115
Şekil 4.17: En genel sistem bileşenleri ve ilişkileri.....	119
Şekil 5.1: Anma değeri değişim grafiği (Deney 1).....	125
Şekil 5.2: $R_{norm}$ değeri değişim grafiği (Deney 1).....	126
Şekil 5.3: Duyarlık değeri değişim grafiği (Deney 1).....	127
Şekil 5.4: “Smaç” eylemini içeren görüntüler kümesi.....	130
Şekil 5.5: “Smaç” eylemini içeren görüntülerin ilgi dereceleri.....	130
Şekil 5.6: “Blok” eylemini içeren görüntüler kümesi.....	131
Şekil 5.7: “Blok” eylemini içeren görüntülerin ilgi dereceleri.....	132
Şekil 5.8: Anma değeri değişim grafiği (Deney2).....	133
Şekil 5.9: Duyarlık değeri değişim grafiği (Deney2).....	134
Şekil 5.10: $R_{norm}$ değeri değişim grafiği (Deney2).....	135
Şekil 5.11: “Akciğer” eylemini içeren görüntü kümesi.....	137
Şekil 5.12: “Akciğer” eylemini içeren görüntüleri ilgi dereceleri.....	138

## ÇİZELGELER DİZİNİ

Çizelge 4.1: NEK Ontoloji modeli sınıfları ve açıklamaları.....	90
Çizelge 4.2: NEK Ontoloji modeli bağıntıları.....	92
Çizelge 4.3.a: “Manşet” eylemi EylemKuralı örneği.....	98
Çizelge 4.3.b: “Sindirim Sistemi” eylemi EylemKuralı örneği.....	98
Çizelge 4.4: Örnek web sayfası – terim vektörü.....	106
Çizelge 4.5: Alan-alan vektörü görünümü.....	107
Çizelge 4.6: Eylem çıkarsama için örnek alan matrisi.....	115

## **EKLER DİZİNİ**

EK-1.a. OntolojiEditörü(OE) Nesne Tanımlama Arayüzü.....	155
EK-1.b. OntolojiEditörü(OE) Eylem Tanımlama Arayüzü.....	156
EK-1.c. OntolojiEditörü(OE) Kavram Tanımlama Arayüzü.....	157
EK-2. OWL ile NEK Ontoloji Modelinde Yer Alan Sınıf ve Bağını Tanımları	158
EK-3 Ağırlıklandırma Etiketlendirme Aracı(AEA)'nın Görüntü Etiketlendirme Arayüzü.....	164
EK-4. Web Sayfası Üst Veri Modelinin RDF ile Gösterimi.....	165
EK-5. NEK Ontoloji Modeli ile Voleybol Ontolojisinin OWL ile Gösterimi.....	169
EK-6. NEK Ontoloji Modeli ile Gövde Alan Ontolojisinin OWL İle Gösterimi.....	184

## **SİMGELER VE KISALTAMALAR DİZİNİ**

AAFE	: Auto Associator Feature Extractor
AANN	: Auto Associative Neural Network
ACL	: Agent Communication Language
AEA	: Anlamsal Erişim Aracı
CBICR	: Content Based Image Retrieval by Clustering
CBIR	: Content based Image Retrieval
CBRS	: Content Based Retrieval System
CT	: Computer Tomography
DAML	: Darpa Agent Markup Language
DCOM	: Disrubuted Object Component Module
ER	: Entity Relational Diagram
ERW	: Entity and Relations on the Web
HACBIR	: An Intelligent Hybrid Approach for Content Based Image Retrieval
HTML	: Hyper Text Markup Language
IR	: Information Retrieval
IRM	: Integrated Region Matching
LUV	: L: luminance, UV: chrominance
Mesh	: Medical Subject Heading
MLP	: Multi Layer Perceptron
MOF	: Meta – Object Facility
NEK	: Nesne, Eylem, Kavram
NS	: Namespace
OCL	: Object Constraint Language
OE	: Ontoloji Editörü
OIL	: Ontology Interchange and Inference
OMG	: Object Managment Group
OWL	: Web Ontology Language
PASS	: Personalized Abstract Search Services
QBE	: Query by Example
RDF	: Resource Description Framework
RDFS	: RDF Schema
RDQL	: RDF Data Query Language
Simplycity	: Semantic-Sensitive Integrated matching for Picture Libraries
STSCBIR	: Semantic Template to Support Content Based Image Retrieval
UFM	: Unified Feature Matching
UML	: Unified Modelling Language
URI	: Uniform Resource Identifier
URL	: Uniform Resource Location
VSM	: Vector Space Model
WWW	: World Wide Web
XML	: Extensible Markup Language

## GENEL BAKIŞ

Ucuzlayan saklama ortamları ve artan internet hızı, geliştirilen uygulamaların türlerinde ve büyüklüklerinde çeşitlilikler oluşmasını olumlu yönde etkilemiştir. Uygulamaların zenginleşmesine bağlı olarak saklanmak ve sağlanmak istenen bilgilerin türleri de çeşitlenmiştir. Örneğin eskiden sadece yapısal metin türünde ifade edilebilen personel bilgilerine ek olarak, bugün personelin resimini de saklamak, bir şehrin bir kesitini tarif ederek resimlerine ulaşabilmek aykırı bulunabilecek ihtiyaçlar değildir. İnternet ortamında yer alan belgelerin en az %20-30'unun görüntü içerdiği tahmin edilmektedir ve bu ihmal edilemeyecek bir orandır[6].

İnternet ortamında her tür veri aynı internet sayfasında rahatlıkla kullanılmaktadır. Arama makinaları internet ortamında bilgi ihtiyacını gidermekte en çok kullanılan araçlardır. Bu tür araçların kullandıkları veri türü metinler olmasına rağmen hepsi belli düzeyde görüntü, resim ve sunular üzerinde arama imkanı vermeye çalışmaktadır [1,2,3,4]. Zira artık bir kullanıcı, bir cins tümör üzerinde araştırma yaparken teorik bilginin yanında o tümörün olduğu belli bir organ için CT (computer tomography) görüntüleri isteyebilir ve günümüzde bu ihtiyacın karşılanması için üzerinde çalışılan birçok konu başlığı bulunmaktadır: içerik tabanlı erişim, anlamsal erişim, internet teknolojileri, nesne tanıma ve etiketlendirme, doğal dil işleme en başta gelen başlıklardır. Bu çalışmanın da, bilgi erişimi (information retrieval) başlığı altında yer alan anlamsal ve içerik tabanlı erişim yaklaşımları ile ilişkili olduğu düşünülmektedir.

İçerik tabanlı erişim sistemleri (Content Based Retrieval Systems-CBRS), yerel ya da internet üzerinde yer alan belgelere; içerikleri üzerinden sorgulayarak erişebilmeyi ifade eden yaklaşımdır. Bu yaklaşım metin, görüntü, video vb. gibi çoklu ortam veri türleri için kullanılabilir. Örneğin içerik tabanlı görüntü erişim sistemi (content based image retrieval-CBIR) ile ifade edilmek istenen görüntünün büyüklüğü, sıkıştırma şekli, türü gibi yapısal özellikleri üzerinden değil; taşıdığı anlam, barındırdığı nesnelere, renkler ve şekiller üzerinden sorgulama yapmak ve erişmektir.



Anlamsal Eriřim (Semantic Retrieval), seilen veri trnn ifade ettiėi anlamlar zerinden eriřime aılmasıdır. rneėin anlamsal grnt eriřimi ile amalanan bir grntde yer alan nesnelere de tede nesnelere bir arada var oluřlarının ifade ettiėi anlam ıkarsanmalı ve sorgulanmalıdır. Grleceėi zere anlamsal eriřim, ierik tabanlı eriřimde otomatik elde edilebilen vektrel ifadelerin yorumlanmasıdır da denilebilir. Dolayısı ile eėer yorum otomatik yapılacaksa aslında anlamsallık ıkarsanarak elde edilebilir. Bu da genellikle alana baėlı alıřmayı ve alana ait bir modeli kullanmayı gerektirir.

Her iki eriřim yaklařımı da tek bir veri tr zerine odaklandığında sadece o veri trleri iin eriřim yeteneėini ifade ederler. Dolayısı ile sadece tarihi eserlerin resimlerinin saklandığı bir sayısal mzede ierik tabanlı grnt eriřimi ya da anlamsal grnt eriřimi gerekleřtirmenin kısıtlayıcılıėından ok bahsedilemeyebilir. Ancak web sayfaları gibi gnmzn en yaygın bilgi ihtiyacı karřılama kaynakları olan ve bilgiyi farklı veri trleri ile ifade edebilen belgelerin eriřimi sz konusu ise tek veri tr kısıtlayıcılıėının tartıřmaya aılabileceėi dřnlmektedir. nk farklı trde veriler bir web sayfasına, sayfanın ierdiėi bilgiyi zenginleřtirmek ya da vurgulamak iin yerleřtirilir gibi bir ngrnn geliřtirilebileceėi dřnlmektedir. Yani sayfa toplamda tm farklı tr verilerin tmleyen katkısıyla oluřan bilgiyi ifade eder. Dolayısı ile sayfanın ierdiėi bu birlikteliėin eriřimde de korunması ve sayfanın seėilen bir veri tr zerinden deėil ierdiėi tm veri trleri zerinden benzer biimde sorgulanabilmesi gerekliliėi dřnlmřtr.

Eriřimi hedeflenen belge tr web sayfalarıdır. Bu sayfalar genelde HTML (Hyper Text Markup Language) ile gsterimi yapılmıř anlamsal olarak birbiri ile iliřkisi kurulmamıř farklı veri trlerinden oluřtuėu kabul edilirse, paralelinde bu tr bir girdiden elde edilebilen ilk zellikler ancak ierik zellikleri olabilmektedir. Aslında bu ařama, ierik tabanlı eriřimin dizinleme ařamasında kullanılacak st verinin edinimidir. Geliřtirilen sistemde ierik zellikleri eriřim iin deėil alan ontolojisi ile birlikte anlamsal bilginin retilmesi ve sayfanın anlamsal st verisinin oluřturulmasında kullanılmıřtır. Dolayısı ile geliřtirilen sistemin ierik tabanlı eriřim sistemleri, anlamsal eriřim ve paralelinde alan modelleme gibi  temel bařlıkla iliřkisi aıktır.

Tez metni de bu ilişkiler dikkate alınarak bölümlere ayrılmıştır. Bölüm 1 'de geliştirilen sistemin temel problem tanımı, problemin çözümü ile beklenen katkıları; bölüm 2' de içerik tabanlı erişim, içerik tabanlı metin ya da görüntü erişimi ile ilgili çalışmalar yer almaktadır. Bölüm 3'te anlamsal erişim üst başlığında yer alan Anlamsal Web'in tanımı ile başlanmakta sırası ile anlamsal erişim, alan modelleme: ontoloji, bulanık ontoloji, ontoloji gösterimi ve uygulama örneklerine yer verilmektedir. Bölüm 4, tez kapsamında geliştirilen anlamsal erişim sistemi; genel yapısı, modülleri, her bir modülün ayrıntılı tanıtımı, dizinleme ve sorgulama gibi iş akışları yer almaktadır. Bölüm 5'de geliştirilen sistemin başarımını ölçmek için düzenlenen deneyler ve sonuçlarından bahsedilmiştir.

# BÖLÜM 1. GİRİŞ

Bu kesimde, geliştirilen erişim sisteminin problem tanımı, çözümü ve katkılarına yer verilmiştir.

## 1.1. Problem Tanımı

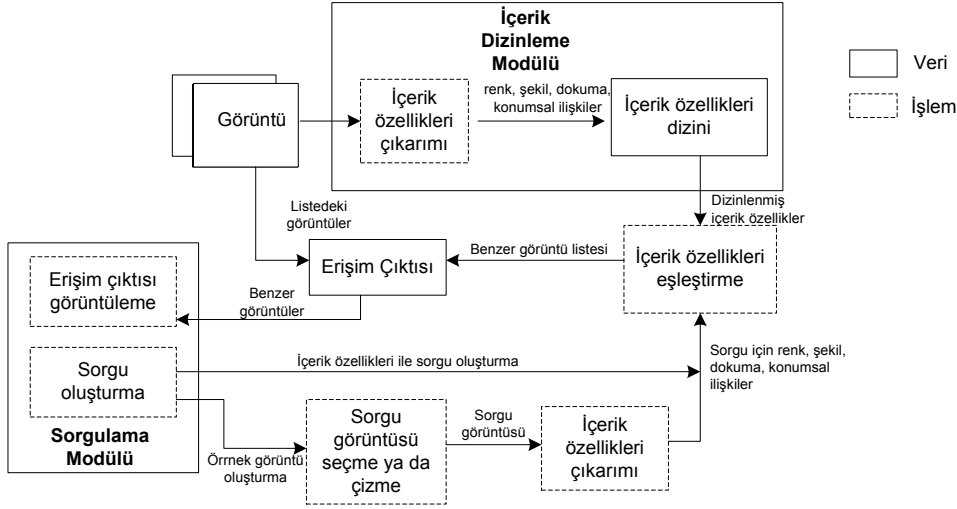
Geliştirilen erişim sisteminin öngörüsü web sayfalarında yer alan farklı tür verilerin sayfanın anlamını zenginleştirmek ya da vurgulamak için kullanıldığı biçimindedir. Özellikle metin ve görüntü veri türlerinin birlikteliklerinin bu öngörü ile genel anlamda uyumlu olduğu düşünülmektedir. Bu öngörüden hareketle çözümü hedeflenen problemin kısa tanımı: web sayfalarında yer alan metin ve görüntü türü verilerin aynı sorgu cümlecığı ile sorgulanabilmesidir. Veri türü olarak metin ve görüntülerin seçilmesinin sebebi, birlikte yer alma sıklıkları ve birbirini tümleyebilen bilgi içerebilme yetenekleridir. “Aynı sorgu cümlecığı ile sorgulayabilme” ifadesi metin ve görüntülerin aynı tür üst verilere dönüştürülmesi anlamına gelir. Sayfanın üst verisi, metin ve görüntülerin üst verilerinin birleşimi olacaktır. Her iki veri türünün üst verileri terimlerden oluşacaktır. Yani metinler zaten terimlerden oluşurken, görüntüler de terimler ile ifade edilecektir. Ancak görüntünün terimler ile ifade edilmesi bile kendi başına içerikten anlamsallığa atılmış bir adım anlamına gelir. Bu noktada öncelikle metin ve görüntü için içerik özelliklerinin neler olduğu, anlamsallığın gerekliliği ve çıkarılabilişliliği incelenecektir.

Metinler için içerik özellikleri metni oluşturan terimlerdir. Terimler diğer veri türlerine göre insana en yakın içerik özellikleridir. Çünkü bilinen anlamları vardır. Ayrıca her hangi bir dil için varolan sözlük, eş anlamlılar listesi, ilgili terimler listesi, konu başlıkları listesi gibi yapıların oluşturulması ve kullanılmasının nispeten daha kolay olduğu düşünülmektedir. Ancak tüm bu avantajlarına rağmen sorgu terimlerinin belge terimleri içinde varolması ya da olmamasına dayanan anahtar kelime arama yaklaşımı ile yetinilmemiştir. Terimlerin birlikte kullanıldıkları zaman kazandıkları anlamın çıkarılması ve terimlerin birbirleri ile ilişkilerinin tanımlanması gibi anlamsal özelliklerin tespitine çalışan çalışmalar mevcuttur[156]. Geliştirilen erişim sistemi de terimler ile bu çerçevede ilgilenmektedir. Yani alana bağlı bulanık bir ontolojide yer alan terimler ve ilişkileri web sayfasının içerik özellikleri olan terimler üzerinde sınanacak ve anlamsal çıkarımlar yapılacaktır. Bu çıkarsama için

metinlerin içerik özellikleri, terimler ve terimlerin belge içi ağırlıkları olarak kabul edilmiştir. Metinler için anlamsal özelliğin birlikte yer alan terimlerin ifade ettikleri anlam ya da kavram olduğu söylenebilir.

Görüntüler ise üç aşamada ele alınıp, işlenebilir. En alt düzeyde görüntünün ham veri hali vardır. Yani seçilen formatta sıkıştırılmış bir dizi bayt'tan oluşan halidir. Bu düzeyde ancak görüntünün sıkıştırma türü, büyüklüğü, yaratılma tarihi gibi yapısal özellikleri (structured features) mevcuttur. İkinci düzey; görsel özellik (visual features) katmanıdır. Bu katmanda görüntünün içinde taşıdığı anlamdan yoksun renk (color), şekil (shape), dokuma (texture) ve bunların görüntü üzerindeki yerleşimleri yer alır. Örneğin sarı pütürcüklü dokumanın üstünde kareye yakın, karışık renkli bir birinden ayırık nesnelere gibi. Üçüncü düzey ise anlamsal özellik (semantic features) katmanıdır. Bu katmanda öncelikle görüntünün taşıdığı nesnelere, daha sonra nesnelere birlikteliklerinin yorumları ele alınır. Örneğin plaj görüntüsü, voleybolda servis atışının görüntüsü, sindirim sistemi görüntüsü ya da karaciğer segmentlerinin görüntüsü gibi. Bu üç düzey içinde yukarı çıktıkça, kullanıcı algılayışına yakın olmakla birlikte işlem karmaşıklığı artmakta, aşağıya inildikçe kullanıcının genel alışkanlıklarının dışında ama sistem için daha kolay işlenebilen ya da elde edilebilen veri gösterimlerine erişilmektedir. Şöyle ki; bir görüntü ele alındığında, en alt düzeyde yer alan sayısal ifadeler, kullanıcılar için anlamsızdır, doğrudan kullanılamaz. Bir üst seviyedeki görsel özellikler kullanıcıya bir miktar daha yakındır. Örneğin su altı görüntüleri arayan bir kullanıcı için arka planın mavi-yeşil tonlardan oluşması kuvvetle muhtemeldir. Kendi ihtiyacını tarif ederken de beklediği görsel özellik değerlerini kullanır. Günümüzde 50 den fazla sistem bu tür orta düzeyli sorgulamalara imkan vermektedir[34]. Oysa ki her ihtiyacın su altı görüntüsünde olduğu gibi beklenen değerleri az sayıda olmayabilir ya da ayırt edici olmayabilir. İnsan için ifade etmenin en kolay yolu kendi doğal dilinde yer alan terimler ile oluşturduğu ifadelerdir. Yani farklı türde olmalarına rağmen "su altı resimleri", "sol lob beyin tümörleri", "voleybolda smaç", "sindirim sistemi" gibi ifadelerin tümü doğal dil terimleri ile oluşturulmuştur ve kullanıcı doğasına çok daha uygundur. İçerik tabanlı görüntü erişim sistemlerinin gelişim sürecine ve türlerine Bölüm 2'de yer verilmiştir. Ancak daha öncesinde tipik bir içerik tabanlı erişim sistemi ortamını şekil ile ifade etmenin problemin tanımı

açısında açıklayıcı olacağı düşünülmektedir. Şekil 1.1 ile tipik bir içerik tabanlı görüntü erişim sistemi örneklenirken; verinin bulunduğu ham ve içerik özellik düzeyi, bu düzeyde yapılabilen sorgulama türleri, sistemdeki konumlanış ve ilişkileri ile gösterilmiştir:



Şekil 1.1: Tipik bir içerik tabanlı görüntü erişim sistemi ortamı

İçerik tabanlı görüntü erişim sistemlerinin tasarım ve gerçekleştirmelerini etkileyen en önemli etken, erişimini amaçladıkları çoklu ortam verisinin doğasından gelen belirsizlik ya da kesin olamama halidir. Tipik bir içerik tabanlı görüntü erişim sisteminde kullanıcıların isteklerini ifade etme biçimleri hiç bir zaman tam, kesin olamamaktadır. Çünkü herhangi bir görüntü, birçok bakış açısına göre farklı anlamlar içerebilmektedir. Hatta bu farklı bakış açıları ya da sonuçları, öznel değerlendirmeler dahi olabilmektedir. Aynı görüntüyü değerlendiren iki kişi, görüntünün üzerindeki nesnelere aynı şekilde adlandırabilir ama görüntünün ifade ettiği olayı, kavramı, duyguyu farklılıklarla değerlendirebilirler yani terimlere dönüştürdüklerinde aynı terimleri seçmeyebilirler. Görüntülerini değerlendirmedeki bu farklılık bilgi ihtiyacını ifade etmekte de kendini gösterir. Dolayısı ile bir görüntünün farklı derecelerde birçok anlamı, özelliği ve ifadesi mümkündür.

Var olan içerik tabanlı erişim sistemleri bu belirsizlik ve tam ifade edememe sorununu kullanıcıdan bilgi alma biçimleri ile çözmeye çalışmışlardır ve en etkin yöntem olarak örnek ile sorgulama (query by example - qbe) geliştirilmiştir. Örneğin "renkleri şu görüntüye benzeyen görüntüler" tarzında bir sorgu, tam olarak

bu belirsizliğe denk düşmektedir. Bu keskin olmayan sorgu, işlendiğinde gerçekleştirilecek işlem; örnek görüntü ile sonuç görüntülerin renk özelliklerine ait vektörel değerlerinin yakınlığını, eşleştirmede kullanmaktır. Örnek ile sorgulamada, örneği hazır olan bir görüntü galerisinden seçmek ya da kullanıcıya kaba taslak hali ile çizdirmek halen kullanılan yaklaşımlardır. CBR sistemleri için bir ileri ki adım, kullanılan görsel özelliklerin ağırlıklandırılması olmuştur. Örnek olarak sunulan görüntü ile istenen görüntünün renk değerlerinin bir birine yakın olması, şekil değerlerinden daha önemli olabilir. Kullanıcı bu yönde bir tercihi halen kullanılan sistemlerde sunabilmektedir. [64,65,66,67,68,]

Günümüzde ticari ya da araştırma amaçlı görüntü erişim sistemlerinde orta düzeyde yer alan yerel - genel renk dağılımı, şekil, dokuma biçimleri gibi görsel özellikler etkin biçimde kullanılmaktadır ama üst düzeyde anlamsal erişimin halen açık bir konu olduğu düşünülmektedir.

Kısaca gelişim süreci gözden geçirilen CBIR sistemlerinde görüldüğü üzere halen en büyük açığın kullanıcı sorgularını ifade etme biçimi ve düzeyi olduğu düşünülmektedir. Çünkü günümüzde kullanıcının artık doğal dil ile ifade etmeye geçmesi artan bilgi ihtiyacı türünün çeşitlenmesinin doğal bir sonucudur denebilir. Aslında kullanıcıların kavramsal düzeyde sorgu oluşturması ihtiyacı neredeyse ilk CBIR çalışmaları ile birlikte tespit edilmiştir. Yani bu açık, yeni tespit edilmemiştir. Ancak bu soruna getirilen ilk çözüm yahoo image surfer[2]'da olduğu gibi anahtar kelime arama yöntemidir. Yani ilgilenilen kelimeyi mantıksal adında taşımak ya da taşımamak görüntünün ilgili, ilgisizlik ayrımını yapmakta kullanılan kriter olmuştur. Zamanla sadece görüntünün adını değil bulunduğu internet adresini ayırıştırarak çıkarılan anahtar kelimeler de ele alınmıştır. Ama halen doğal dil terimleri ile ifade edilen, dinamik ve yeni kavramları kolayca kapsama yeteneğinde bir sisteme duyulan ihtiyacın tatmin edici biçimde karşılanamadığı düşünülmektedir. Özetle, bu çalışmanın görüntü türü veriler için hedefi özellikle 2000 tarihinden sonra başlatılan birçok benzeri proje ile aynıdır: görüntüler için etkin anlamsal sorgulama. Bu önerilen sistemin çözüm için ortaya koyduğu problemlerden biridir.

Bu noktaya kadar yapılan açıklamalarla metin ve görüntü türü verilerin içerik ve anlamsal özelliklerinin neler olabileceği açıklanmıştır. Dolayısı ile sistemin asıl

amacı, büyük ağırlığı görüntüler için anlamsal sorgulama olarak görünmekle birlikte aslında bir internet sayfası için tam anlamsal dizinleme ve erişimi hedeflemektedir. Yani internet sayfalarının büyük çoğunluğunda yer alan iki ana veri türü olan metin ve görüntüler birlikte ele alınarak sayfaların anlamsal özelliklerine ulaşılabacaktır.

Bu hedefler doğrultusunda tasarlanan sistemin genel özellikleri şu biçimde listelenebilir:

1. Metin ve görüntü olmak üzere iki ayrı veri türü için koşt çalışıp, iki ayrı veri türü için anlamsal özellikler çıkarılacak ve birleşimleri kullanılarak sayfa için tam anlamsal dizinleme söz konusu olacaktır. Şöyle ki, vektör uzayı modeli ile metin erişimi yaklaşımında terimlerin ağırlıkları derlemde ve belgede geçme sıklıklarına göre üretilir. Anlamsal erişimde bu terimlerin birlikte kullanıldığı zaman ifade ettikleri yeni özellikler çıkarılmalıdır. Tam anlamsal dizinleme gerçekleştirildiğinde de görüntülerden çıkarılan nesne, eylem ve kavram adları da sayfanın anlamsal içeriğine eklenecektir. Bunun sonucunda terimlerin belgede ve derlemde geçme sıklıkları da değişecektir. Bu katkının anma (recall) değeri üzerinde olumlu etkisi beklenmektedir.
2. Sistem, içerik özellikleri ile anlamsal özellikler arasında kalan boşluğu alana bağlı bulanık bir ontoloji kullanarak dolduracaktır. Önerilen ontoloji modeli ile seçilen bir alanda yer alan terimler nesne, eylem ve kavram kategorilerine bölünmekte ve bu kategorileri birbiri ile ilişkilendirilmektedir. Metinlerin içerik özelliğini oluşturan terimler bu ontolojide ilişkili oldukları diğer terimler ile birlikte kullanım durumlarına göre metnin anlamsal özelliğini belirlemektedir. Yani metine, ontolojide tanımlanmış ilişkiler sebebi ile yeni terimler yeni ağırlıklar ile ya da zaten varolan terimler yeniden hesaplanmış ağırlıkları ile eklenmektedir.

Görüntüler içinse görüntüde yer alan nesnelerin adları ve bu nesnelerin birlikte ifade ettiği eylem ve/veya kavram adları ontolojiden alınan bilgi ile çıkarsanmakta ve görüntünün anlamsal özelliğini oluşturmaktadır.

3. Sistem önerdiği modele göre ontoloji oluşturma, değiştirme, inceleme gibi yetenekleri içeren bir araç içermektedir. Oluşturulan ontolojiler XML (Extensible Markup Language) tabanlı ontoloji gösteril dili OWL ile ifade edilmektedir ve paylaşılabilir özelliktedir.
4. Kullanıcı kendi profilini yani kendi ontolojisini oluşturabilmektedir. Yani sistemin kullandığı genel bir ontolojinin yanı sıra; kullanıcının, özellikle yorumlarını ifade eden karmaşık terimleri kendi ontolojisi ile tanımlaması mümkündür. Örneğin 'mutluluk' terimi için hangi durumları mutluluk olarak değerlendirdiğini ontolojisinde ifade ederse, onun bakış açısı ya da yorumu için aslında bir profil elde edilmiş olacaktır.
5. Sistemin anlamsal erişim hedefi görüntü verilerinde nesne adlandırma ya da nesne etiketlendirme adı verilen alt aşamanın gerçekleşmesini zorunlu kılmaktadır. Ancak bu konu başlığında yapılan çalışmalarda nesne etiketlendirme başarımlarının ancak önceden tanımlı ve kapalı bir nesne alt kümesi için kabul edilebilir olduğu gözlemlenmiştir. Sistemin nesne etiketlendirme işlemine ihtiyacı vardır. Ancak bu işlemi otomatik yapacak hazır araç bulunamaması sebebiyle kullanıcıya nesne etiketlendirme yapabileceği bir araç sunulmuştur. Kullanıcı sunulan etiketlendirme aracı ile tipik bir görüntüde yer alan nesnelere etiketlendirir. Nesnelere birliktelikleri ve konumlanışları otomatik tespit edilmektedir.
6. Kullanıcı aynı sorgulama biçimi ile ister metin ister görüntü isterse hem metin hem de görüntüleri aynı anda sorgulayabilecektir. Yani sorgulama terim tabanlıdır. Terimler "ve" ve "ya da" işlemleri ile bağlanabilmektedir. Sorgulama ontoloji sorgulama dili olan RDQL ile gerçekleştirilmiştir. Ancak sorgulama aracının sunduğu arayüz kullanıcı dostu olacak biçimde tasarlanmış ve kullanılan teknolojiden soyutlanmıştır.

## **1.2. Problemin Seçilme Nedeni ve Çözümde Beklenen Katkıları**

Bilgi ihtiyacını bir derlem üzerinden gidermek günümüzde birçok yazılımın temel ya da yan işlevlerinden biridir. Derlem ile kastedilen çoğu zaman internet olmakla



birlikte özel hazırlanmış bilgi havuzu da olabilir. Günümüzde metin türü veriler, bu tür ihtiyaç duyma ve giderme döngüsünde çoktan yerini almıştır. Ancak görüntüler için aynı tespiti yapmanın fazla iyimserlik olacağı düşünülmektedir. Örneğin uzaktan eğitim günümüzde Türkiye dahil birçok ülkenin gündemindedir. Bu alanda birçok ürün geliştirilmiştir ve doğaldır ki uzaktan eğitim hizmeti veren her üründe, öğrenciye çeşitli şekillerde arama imkanı veren araçlar vardır. Ancak bu araçlar günümüzde halen metin tabanlıdır. Oysa ki kullanılan kaynakların hepsi sadece metin kaynaklı değildir. Dolayısı ile aramanın hedefi, varolan kaynakları kapsayacak bir arama olmalıdır. Geliştirilen sistemin hedefi de bu noktada aynı basitlikte ve anlamsal düzeyde sorgulamayı metin ve görüntüler için gerçekleştirmektir. Çünkü uzaktan eğitim alanında verilen örnekle devam edilirse, dersin ortasında öğrencinin görüntü sorgulamak için sorgusunu çizmek, seçmek ya da tarif etmek için uğraşması çok makul değildir. Çünkü bu tarz sorgulama pratik değildir. Uzaktan eğitim burada geliştirilen sistem için bir uygulama alanı olarak gösterilmiş olmakla birlikte aslında sistem arama eyleminin gerekli olduğu tüm alanlara hizmet etmeyi hedeflenmektedir ki bunların başında da internet gelmektedir. Ancak ihtiyacı açıklamak için bir örnekte medikal arşiv sistemlerinden verilecektir. Tıbbi görüntüleme cihazlarının ürettiği veriler temelde görüntülerdir. Bu görüntüler sunucular üzerinde saklanır. Ancak görüntülerin erişimleri ile ilgili halen etkin bir çözümün varlığı gözlemlenmemiştir. Bu alanda örneğin “Karaciğer 7. segment” biçiminde bir bilgi ihtiyacını çizerek ifade etmek güçtür. Zira karaciğer’in bütünüyle görünümü, görüntülerde şekil ve yer değiştirmektedir. Dolayısı ile diğer organlar ile konumsal ilişkileri ile ancak bir tarif yapılabilir. Bu da öncelikle her sorgulayan kullanıcının ilgili tarifi yapabilecek düzeyde alan bilgisine sahip olmasını gerektirirken, aynı zamanda birden çok tarifi de yapabilme yetisini gerekli kılar.

Görüntü türü verilerin bilgi ihtiyacını gidermek için kullanımının artması, erişimin ve sorgulamanın kolaylaşmasına bağlı olduğu düşünülmektedir. Günümüzde bu veri türü üzerinde bir talep olduğu, bu alanda yapılan çalışmaların sayıca fazlalığına bakılarak söylenebilir. Ancak günlük hayatta kullanım oranının metinlere göre oldukça düşük olduğunun da birçok arama motoru kullanıcısının ortak kanaati olduğu düşünülmektedir. Çünkü görüntü erişim sistemlerinin başarımı, metin erişim sistemlerine göre bir hayli düşüktür ve günümüzdeki ticari

sistemler ile orta düzeyde sorgulanma gerçekleřtirmek zahmetlidir. Görüntü eriřim sistemlerinin kullanım kolaylıđı ve etkin eriřimi bu veri türünü günlük yařantımıza sokacaktır. Orta düzeyli sorgulama bir miktar başarı kazanıp, kendi ekolünde ticari ürün çıkmasını sağlamıřtır ama rutin bilgi ihtiyacı giderme aracı olarak kullanım oranları düřüktür.

Sistem, hedeflediđi tam anlamsal eriřimi gerçekleřtirdiđinde beklenen olumlu katkılar kısaca řöyle listelenebilir:

- Kullanıcı metin tabanlı sorgulama alışkanlıđını deđiřtirmeden görüntüler için kullanacaktır. Bu kolaylık, bilgi ihtiyacı giderme sürecinde, görüntülerin kullanım yaygınlıđını olumlu yönde etkileyecektir.
- Görüntüler için görsel özellikle ifade edilemeyen birçok rol, eylem, kavram adı da sorgulamaya açılacaktır.
- Belgeler salt metin ya da salt görüntüsü ile sorgulanmanın üstüne çıkacak aynı sorgu ile iki ayrı veri türü için de eriřimi başlatılabilecektir. Yani belge daha üst bir konumdan ve daha zengin içerikle sorgulanacaktır.
- Metin tabanlı dizinleme ve eriřimde çıkarsanamayan anlamsal özelliklere ek olarak görüntülerden gelen anlamsal özellikler sayfanın anlamsal vurgusunu ve zenginliđini yansıtacaktır. Bu katkı ile eriřim performansı olumlu yönde etkilenecektir.
- Sistemin kullanacađı ontoloji internet ortamında paylařıma açık bir standart ile gösterilecektir. Dolayısıyla kendi ontolojisini internet ortamında paylařıma sunduđu gibi aynı standartta başka ontolojileri de kullanabilecektir.

## **BÖLÜM 2. İÇERİK TABANLI GÖRÜNTÜ ERİŞİMİ VE İLGİLİ ÇALIŞMALAR**

Bu alanda yapılan çalışmaların gelişim sürecine bakıldığında 1996 IBM tarafından geliştirilen QBIC, CBIR sistemleri için milat olarak kabul edilebilir[61]. Çalışmalar özellikle 1999 yılına kadar yerel veri tabanında yer alan görüntülerin erişimi üzerine yoğunlaşmıştır. Görüntülerin ifade edilmesinde ilk kullanılan özellik renk (colour) olmuştur[8]. Bunu dokuma (texture) ve şekil (shape) özellikleri takip ederken bir yandan da bu özelliklerin birbirine göreli konumlarını ifade eden konumsal ilişkiler (spatial relations) ortaya çıkmıştır. Bu noktada, görsel özellik çıkarma konusunda görüntü işleme algoritmaları yaygınlaşırken diğer yandan büyük hacimli olan bu tür verilerin sıkıştırılması başka bir çalışma konusu olmuştur. Aynı zamanda görüntü, birden çok görsel özellik ile ifade edildiğinden çok boyutlu etkin dizinleme yöntemleri üzerinde çalışılmıştır[40,41,42]. Özellikle 1996-98 yılları arası tam bir sistem patlaması gerçekleşmiştir ve bu aralıkta geliştirilen sistemler birbirlerine çok benzemektedir. O dönem çıkan sistemler iki ana kategoriye ayrılabilir:

1. Görüntüyü orta düzeyde ifade etmek için seçilen özelliklerin keskin değerleri üzerinden erişim yapmak: Bu tür sistemlerin genel çalışma hatları şu biçimdedir: bir ya da daha çok görsel özellik (renk, şekil, dokuma), içeriği ifade etmek için kullanılır. Bu özellikleri çıkarmak için gerekli algoritmalar seçilir ya da geliştirilir. Çıkarılan özellik bilgisi vektörel dir yani keskindir ve çok boyutludur. Sorgulama örnekle yapılır. Yani kullanıcı kendisi için doğru olan bir örnek görüntü seçer ya da çizer. Sorgu görüntüleri de derlem görüntüleri ile aynı içerik çıkarma algoritmasına tabi tutulur. Yani aynı tür vektörel ifadelere dönüştürülür. Sorgu ve derlem görüntülerinin vektörel değerleri arasındaki uzaklık (distance); eşleştirmenin ana kriterini oluşturur. Kullanılan görsel özellik çıkarma algoritmaları, uzaklık formülü, özelliklerin türü, sayısı, oluşturulan arayüzlerdeki farklılıklara rağmen sistemlerin ana mantığı bu biçimdedir. Kullanılan veri türünden kaynaklı belirsizlik, özellikleri ağırlıklandırma ve örnekle sorgulama ile ele alınmaya çalışılmıştır. [6,7,8,9]

2. Görüntüyü orta düzeyde ifade etmek için seçilen özelliklerin keskin değerlerini önceden tanımlı bulanık kümelere yerleştirerek erişim yapmak: Bu kategoride çok daha az oranda çalışma yapılmıştır. Doğrudan bulanık modeller kullanılarak CBIR işlemi gerçekleştirilmeye çalışılmıştır. Bu tür çalışmaların doğmasındaki asıl motivasyon kullanıcının sorgularını keskin değer ile ifade edememesi ve aslında milyonlar ile ifade edilen farklı görsel özelliğin insanlar tarafından o ayrıntıda algılanamamasıdır. Amaçlanan keskin vektörel değerleri insanlar için anlamlı bulanık kümelere dönüştürmektir. Örneğin renk özelliği için ana renkler ve bunları açık, çok açık, koyu, çok koyu ve tam gibi tonları insan algılayışı için yeterlidir. Sorgulama ve erişimde bu tür bulanık terimlerin eşlik ettiği bulanık kümeler kullanılır. [16,17,18,19,20,21,22]

Ancak dikkat edilirse her iki türdeki çalışmalar sorgulama ve erişim modelini orta düzeyde olmaktan kurtaramamıştır.

Özellikle 1999 yılından sonra ilk kategoriye giren çalışmalar için ikinci atılım dönemi başlamış ve 2001 yılına dek devam etmiştir. Bu dönemin ana özelliği internet üzerinde içerik tabanlı erişimi gerçekleştirmek olmuştur. Bu konuda yapılan çalışmalar da üç kategoriye ayrılmaktadır:

1. Genelde 1998'e kadar yerel veri tabanı için geliştirilmiş CBIR sistemlerinin dizinleme ve erişimde kullandığı görsel özellik türü ile çıkarsama biçimi aynı kalmak koşuluyla internet üzerinde uygulamaya geçirilmesi. [7,8,9,55,62]
2. Ajan, robot teknolojilerini kullanmak. Doğrudan internet üzerinde geliştirilen ve görüntünün içeriğini daha etkin ifade etme, etkin sorgulama sorunları ile değil, hızlı ve etkin ajan kullanımı üzerine çalışılmıştır. Çalışmaların asıl ana alanı ajan teknolojisidir. [7, 38]
3. İnternet üzerinde sadece görsel özellikler ile değil; yardımcı olacak etiket (tag), dizin adı, web adresi gibi yapısal bilgileri ya da WordNET'i kullanmak. Bu sistemler görsel özellikleri kullanma konusunda kendinden önce gelen sistemleri kapsarlar. Ancak ek olarak yeni bir özellik eklenir. Anahtar kelime

olarak ifade edilen bu özelliğin içine çeşitli kaynaklardan topladıkları terimleri yerleştirirler. Bu tür sistemlerin çıkış motivasyonu kullanıcıya üst düzeyde sorgu sunma imkanı vermektir. Ancak kullandıkları terim toplama kaynakları sınırlı olduğundan daha çok internet ortamının anlamsal kataloglanması noktasında kalmışlardır. Kataloglama konusunda da daha önce İngilizce için hazırlanmış terimlerin birbiri ile anlamsal ilişkilerini sıradüzensel biçimde gösteren WordNET<sup>1</sup>'ten yardım alınmaktadır. Dolayısı ile sistemler İngilizce doğal diline WordNET'ten dolayı bağımlıdır. Bu kategoriye tam girmese de, yine WordNET'i kullanan ve yapay sinir ağları ile öğrenme ya da makina öğrenimi gibi yaklaşımlar da mevcuttur. Bu çalışmalarda görüntüler görsel özelliklerine göre WordNET'te tanımlı bir kavrama eşleştirilmeye çalışılmıştır. Başarımlar düzeyleri düşük olup, yarı otomatik işlemlerdir. [24,25,26,31,32,35,37,52,57,58,59]

Bilgi erişiminde ontoloji kullanımı yönünden yaşanan süreç Bölüm 3'te özetlenmiştir.

CBIR sistemlerin erişim başarımları anma ve duyarlık (recall, precision) parametrelerine göre metin tabanlı sistemlerden daha geridedir. Kapsam olarak bilgi erişimi, görüntü işleme, yapay zeka, doğal dil tanıma gibi çok çeşitli konuların kesiştiği bir noktada yer alan CBIR üzerine gerçekleştirilmiş ve internet ortamında deneme erişimlerine izin veren deneysel çalışma sayısı ellinin üzerindedir[34]. Bu çalışmaların her biri CBIR bağlamında farklı bir aşamaya ağırlık vermişlerdir. Yani bir tanesi etkin bir kenar yakalama algoritmasını denerken bir diğeri, kullanıcının sorgulama biçimi üzerine farklılıklar denemiştir. İşte tam bu sistem bolluğu içerisinde yeni bir çalışma alanı çıkmıştır. Bir arama makinesi görünümü ile sorguyu alıp altta birden çok sistem için her birinin gereksediği sorgu biçimine çevirip; bir sorgu ile bir kaç sistemi aynı anda tetiklemek ve hepsinin en benzer olarak tespit ettikleri erişim çıktılarında tek bir erişim çıktısı hazırlayıp, kullanıcıya sunan sistemler gündeme gelmiştir. Bu yolla tek tek ele alındıklarında düşük anma ve duyarlık değerlerine sahip CBIR sistemlerin bir arada daha etkin değerlere çıktığı görülmüştür. Bu alanda yapılan en kapsamlı çalışma MetaSEEK'tir[8].

---

<sup>1</sup> WordNet tüm nesnelere göre hiyerarşik konumunu gösteren ve içinde barındırdığı nesnelere için örnek görüntü içeren hazır yapıdır.

Ticari yazılım üreticilerinin CBIR üzerine verdikleri desteğe de kısaca değinmek yerinde olacaktır. İlk olarak bu alana ilgi duyan yazılım firmaları internet üzerinde ya da veri tabanı üzerinde erişim ve işlem yapanlar olmuştur. Hatta daha netleştirmek gerekirse bu alandaki ilk destek metin tabanlı arama makinaları ve ilişkisel modelde veri tabanı sistemleri üreticilerinden gelmiştir. Ancak bu destekler ön açıcı mahiyette olmaktan çok geliştirilmiş ve kabul edilebilir etkinliğe ulaşmış yöntemlerin uyarlanması şeklinde oluşmuştur. Bu özelliklerinden dolayı burada açıklanan yöntemlerin içinde kaldıklarını belirtmek yeterlidir.

Bu noktaya kadar yapılan çalışmaların tarihsel gelişimleri ve türleri ana hatları ile incelenmiştir. Ancak bu genel çerçevenin yanı sıra, CBIR alanında özellikle 2000 yılından sonra ortaya çıkan önemli çalışmalar mevcuttur. Elbetteki bu çalışmalar da yukarıda belirtilen kategorilerce kapsamaktadır. Ancak yapılan literatür taramasında incelenen CBIR sistemleri arasında her biri, bir yönüyle öne çıkan bu çalışmaları tek tek incelemek ve tez kapsamında önerilen sistemin varolanlar ile farkını bariz olarak sergilemek ve CBIR ile ilişkisini ortaya koymak için gerekli olduğu düşünülmektedir. Ayrıca görüntü için içerik özelliklerinin ne olduğu ve nasıl elde edilebildiğine ilişkin güzel örneklerdir. Ayrıntılı ele alınan sistemler CBIR alanında önde gelen çalışmalar olmalarının yanı sıra özellikle bazılarının hedeflediği sorun tezde hedeflenen sorunla aynıdır. Ancak ortada çok açık bir yöntem ve yaklaşım farklılığı mevcuttur. İlerleyen kesimde bu sistemler tanıtılmıştır.

## **2.1. Resim Kütüphaneleri İçin Anlamsal – Duyarlı Tümlü Eşleştirme (*Simplicity: Semantic-Sensitive Integrated matching for Picture Libraries*)**

Simplicity, 2001 yılında Pensilvanya Üniversitesi'nde geliştirilen bir CBIR sistemidir[101]. Ancak Simplicity'nin tam anlamıyla bir içerik tabanlı erişim yaptığını söylemek yerine; içerik tabanlı erişimle aynı tarzda vektörel üst veri (metadata) kullandığını söylemek daha doğru olur. Zira Simplicity ile amaçlanan görsel özelliklerden çıkarılan üst verileri kullanarak, görüntüleri önceden tanımlanmış ve elle yaratılmış anlamsal sınıflardan birine atamaktır. Bu anlamsal

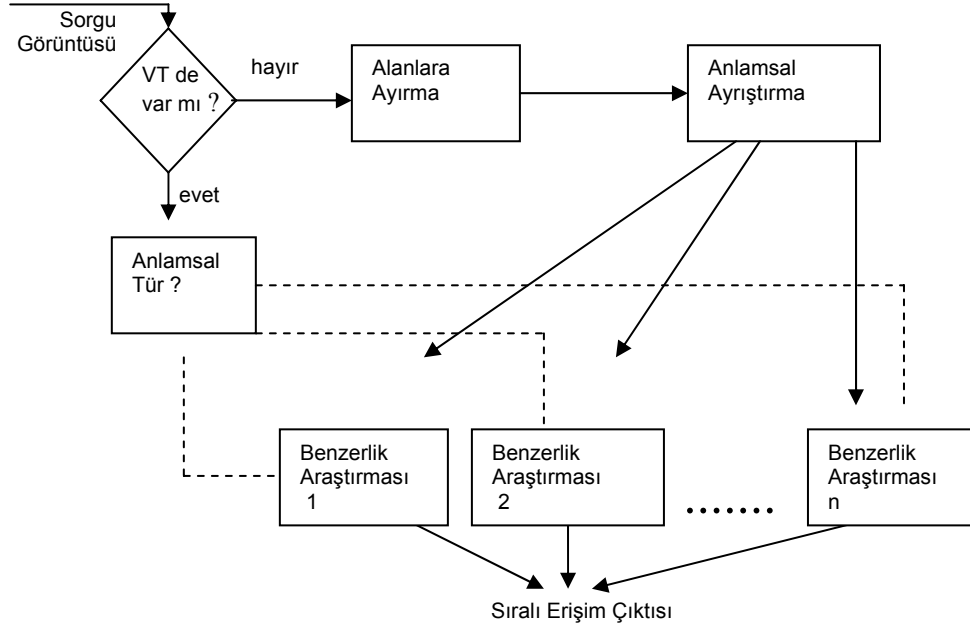
sınıflar çizim-fotoğraf, dokuma arka planlı-düz arka planlı, iç mekan-dış mekan, şehir-manzara, insanlı-insansız...vs biçiminde örneklenebilir. Aslında seçilen anlamsal sınıflara göre farklı görsel özellikler önem kazanır. Örneğin arka planın dokumalı ya da düz olması gibi iki anlamsal sınıf seçilirse açıkça şekil özelliğinin anlamı azalırken, dokuma özelliğinin önemini artacaktır. Oysa anlamsal sınıflama iç mekan-dış mekan olarak belirlenir ise dokuma özelliği önemini yitirirken şekil özelliğinin önem kazanacağı açıktır. Aslında SimplyCity'nin ana fikri de burada yatmaktadır. Bu fikir "görsel özellik çıkarma biçimi, anlamsal özellikler ile aradaki boşluğu doldurma biçiminden etkilenir" [101] biçiminde ifade edilebilir.

SimplyCity'de, "dokumalı- düz arka plan" ve "çizim-fotoğraf" olmak üzere kullanılan iki anlamsal ayırım vardır. Simplycity'de bu ayrımları başarı ile gerçekleştirmek için Tümleşik Alan Eşleme (IRM-Integrated Region Matching) yöntemi geliştirilmiştir ve sistemin bu yönü sistemin özgün özelliğidir[101]. Bu yöntemin geliştirilmesini tetikleyen temel sorun bölümlenme (segmentation) olarak tanımlanabilir. Şöyle ki, bir köpek resmini arka plan ve köpek nesnesi olarak iki bölüme ayırmak mümkündür. Ancak aynı görüntüyü arka planın yanı sıra köpeğin başı, köpeğin ayakları, köpeğin kuyruğu biçiminde çok daha fazla bölüme ayırmak mümkündür ve aslında iki yaklaşım da tamamen yanlış ya da doğru değildir. SimplyCity bu tür bölümlenme sorunlarını IRM ile en aza indirmeyi hedeflemiştir[101].

Simplycity'de, görüntüler IRM ile 4x4 lük piksel parçalarına ayrılır. Bu parçalar, seçilen görsel özellik(ler) kriter olmak koşulu ile istatistiksel kümeleme (statistical clustering) yöntemine tabii tutulurlar. Kümelemedeki amaç; benzer piksel parçalarını birleştirerek alanları elde etmektir. Sonuçta elde edilen alanlar sınıflandırıcı (classifier) adı verilen bir modül aracılığı ile önceden tanımlı n adet anlamsal sınıftan birine atanır. Simplycity'de bu atama işlemi tamamen keskin küme (crisp set) kurallarına göre gerçekleştirilir[101].

Simplycity'de sorgulama QBE yöntemi ile gerçekleştirilir. Eğer sorgulamada kullanılan görüntü sistem veri tabanında saklı değilse, sorgu görüntüsü de yukarıda anlatılan 4x4'lük piksel parçalarına ayrıştırılma, istatistiksel kümeleme ile alanları tespit etme ve sınıflandırıcı modül ile bir anlamsal sınıfa atanma işlemlerinden geçer. Sorgu görüntüsünün anlamsal sınıfı tespit edildikten sonra

ilgili kümede yer alan görüntüler ile sorgu karşılaştırılır ve sıralı bir erişim çıktısı oluşturulur. Şekil 2.1’de SimplyCity’nin genel çalışma mantığı ifade edilmiştir[101].



Şekil 2.1: SimplyCity’de sorgu işleme iş adımları

Simplycity’deki erişimin genel görüntüsünü verdikten sonra aslında özgün olan IRM yöntemini ve anlamsal sınıf tespitinin ayrıntılandırılmasının anlaşılabilirlik açısından gerekli olduğu düşünülmektedir.. Anlamsal sınıf tespitinde tüm anlamsal sınıfları açıklayıp okuyucu ilgisini dağıtma ihtimalinden dolayı fikir verebilmesi için sadece dokumalı-dokumasız ayrımı açıklanacaktır.

IRM yönteminde 4x4’lük piksel parçaları 6 görsel özellik kullanılarak ‘k-means’ algoritması ile kümelendir. Bu özelliklerden üçü piksel gruplarının ortalama renk bileşenleridir. Bunlar LUV (L: luminance, UV: chrominance)’dir. Diğer kullanılan üç özellik ise yine bu piksel parçalarının L özelliklerine, LH ve HH bandlarına uygulanan Daubechies-4 ya da Haar Wavelet dönüşümleridir. Aslında bu 4x4’lük piksel parçaları 2x2’lik 4 grup şeklinde yorumlanmaktadır ve bir 4x4 lük parçayı oluşturan 2x2’lik alanlar  $\{c_{k,l}, c_{k+1,l}, c_{k,l+1}, c_{k+1,l+1}\}$  olarak gösterildiğinde bir piksel parçasının bir özelliği aşağıdaki formül ile son değerini alır[101]:

$$f = \left( \frac{1}{4} \sum_{i=0}^1 \sum_{j=0}^1 c_{k+i,l+j}^2 \right)^{1/2}$$



Her bir piksel parçası için özellikler çıkarıldığında ‘k-means’ ile kümeleme yapılarak alanlar tespit edilir. SimplyCity’de, alanların tespiti yapıldıktan sonra anlamsal sınıf olarak seçilen dokumalı – dokumasız ayrımı  $X^2$  istatistik ölçümü ile yapılır. Şöyle ki; görüntü, 16 adet bölge (zone)’ye ayrılır ve  $\{Z_1, Z_2, \dots, Z_{16}\}$  olarak gösterilebilir. Görüntünün k-means algoritması ile m adet alana ayrıldığı varsayılırsa  $\{r_i: i=1..m\}$  biçiminde gösterilebilir. Her  $r_i$  alanı için  $Z_j$  bölgesi içinde yer alma yüzdesi  $p_{i,j}$  ise ve bölgeler arasında düzgün dağılıma çokluk olasılık işlevi (probability mass function)  $q_j = 1/16$  ise, i alanı için  $X_i^2$  şu biçimde hesaplanır[101]:

$$X_i^2 = \sum_{j=1}^{16} (p_{i,j} - q_j)^2 / q_j = \sum_{j=1}^{16} 16(p_{i,j} - 1/16)^2$$

Daha sonra “m” alan sayısı olmak üzere

$$X^2 = 1/m \sum_{i=1}^m X_i^2$$

olarak görüntü için toplam  $X^2$  istatistik ölçümü ile yapılır. Eğer  $X^2 < 0.32$  ise görüntü dokumalı, değilse dokumasız anlamsal sınıflarına atanır[101].

Anlamsal sınıflama işleminden sonra görüntü eşleştirme yani sorgulamaya sıra gelir. Simplycity’nin öncelikli hedefi bölümlenme sorununu en aza indirerek anlamsal sınıfları ortaya çıkarmak biçiminde özetlenebilir. Bu bakış açısından hareketle Simplycity içinde yer alan görüntü eşleştirme yaklaşımı da bölümlenme hatalarına karşı önlemler içermektedir. Şöyle ki;  $R_1$  ve  $R_2$  ayrı iki görüntü olsun.  $R_1 = \{r_1, r_2, \dots, r_m\}$  ve  $R_2 = \{r'_1, r'_2, \dots, r'_n\}$  biçiminde alanlardan oluştuğu varsayalım.  $R_1$  ve  $R_2$  arasındaki benzerlik araştırmasında; sadece karşılıklı gelen  $r_1$  ile  $r'_1$ ,  $r_2$  ile  $r'_2$ ’nin benzerliğine bakmak yerine  $r_1$ ’in tüm  $R_2$  alanları ile ve  $r_2$ ’nin tüm  $R_2$  alanları ile ve..... $r_m$ ’in tüm  $R_2$  alanları ile benzerliğine bakılır. Ancak burada S ile gösterilebilen bir önem matrisi (significance matrix) devreye girer. Mesela  $r_2$ ’nin  $r'_6$ ’ya benzemesinin ne kadar önemli olduğu önem matrisindeki  $s_{2,6}$  nolu hücrenin değeridir. Dolayısıyla matrisin genel görüntüsünü ve bu matrise bağlı olarak  $R_1$  ve  $R_2$  görüntüleri arasındaki benzerliği, şu biçimde gösterilebilir[101]:

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \dots & \dots & \dots & \dots \\ s_{m,1} & s_{m,2} & \dots & s_{m,n} \end{bmatrix} \text{ ve}$$

$$\text{Sim}(R_1, R_2) = \sum_{i,j} s_{i,j} \times \text{Sim}_{i,j}$$

Benzerlik formülünde yer alan  $s_{i,j}$  önem matrisinde i. satır ve j. sütünü ifade etmektedir.  $\text{Sim}_{i,j}$  ise k boyutlu öklidyen uzayda (k dimensional euclidean space) uzaklığın tersidir.

Simplycity'nin başarısı 200.000 görüntü üzerinde sınanmıştır. Kendi yaklaşımına benzer yaklaşımlar taşıyan WBIIS (Wavelet Based Image Indexing and Searching) ile karşılaştırılmış ve daha başarılı bulunmuştur.

## **2.2. İçerik Tabanlı Görüntü Erişimi İçin Alan Tabanlı Bulanık Eşleştirme Yaklaşımı (A Region Based Fuzzy Feature Matching Approach to Content Based Image Retrieval)**

Bu çalışma Y. Chen ve J. Wang tarafından 2002 yılında Pennsylvania Üniversitesi'nde gerçekleştirilmiştir. Bu çalışmanın da SimplyCity adlı çalışmayı gerçekleştiren ekip tarafından yapıldığını belirtmek gerekir[104]

Bulanık mantığın görüntü erişim sistemlerinde kullanılması, görüntülerin doğasında yer alan yoruma açık ve keskin biçimde anlamlandırılmama özelliğinden kaynaklanan ve oldukça sık rastlanılan bir seçimdir ve bünyelerinde bulanık mantık yaklaşımını bulunduran birçok çalışma mevcuttur. Bu sistemin alanında getirdiği yenilikler iki madde altında toplanabilir[104]:

1. Kullanılan yaklaşım alan tabanlı bulanık özellik eşleştirmedir. Özellikle şekillere ait net belli olmayan sınır (blurry boundary)'lar söz konusu olduğunda ortaya çıkan bir bölümlenme sorunu vardır. Bu sorunu ele almak için, görüntüler alanlara bölünür ve aslında her alan bir bulanık kümedir. Uzayda küme olarak gösterilen her alanın şekil, renk ve dokuma özellikleri mevcuttur. Dolayısıyla her görüntü aslında birçok bulanık küme ve özelliğin

birleşimiyle elde edilir. Bu yaklaşımdan beklenti, bulanık kümeler ile net belli olmayan sınırlar arası geçişin karakterize edilmesidir. Her alan, bu kümelerden bir ya da bir kaçına belli üyelik dereceleri ile ait olacak ve her görüntü bir grup bulanık kümeyle ifade edilecektir. Böylesi bir yaklaşım klasik alanlara bölme ve alanları karşılaştırma yaklaşımından farklıdır. Zira geleneksel yaklaşımda bir alanın sadece bir görsel özelliği vardır. Oysa bu yaklaşımda, bir alan birçok görsel özelliği üyeliğiyle doğru orantılı olarak taşıyabilir gibi bir yorum ortaya çıkabilir.

2. Kullanılan bulanık küme yaklaşımına bağlı olarak görüntüler arası benzerlik araştırması da geleneksel yaklaşımdan farklı olarak bulanık küme işlemlerine göre gerçekleştirilir. Birleştirilmiş özellik eşleştirme (Unified Feature Matching – UFM) adı verilen yaklaşımda eşleştirme üç aşamada yapılır. İlk adımda sorgu görüntüsünün tüm görsel özellikleri, kayıtlı görüntünün görsel özellikleri ile bulanık küme kuralına göre eşleştirilir. Daha sonra ikinci adım olarak kayıtlı görüntünün görsel özellikleri aynı kurallarla sorgunun görsel özellikleri ile eşleştirilir. Son olarak toplam benzerlik, UFM ölçümüne göre ilk iki adımın sonuçlarının uygun ağırlıklandırılması ile hesaplanır.

Sistemde görüntü bölümlenmesi, görüntülerin üzerinde yer alan renkler ve renk dağılımlarına göre 'k-means' algoritması ile kümelenebilmesi ile gerçekleştirilmiştir. Sistemin bölümlenme işlemine ilk girişi yaptığı nokta SimplyCity ile tamamen aynıdır. Yani görüntünün 4x4'lük piksel parçalarına ayrılması ve daha sonra 'k-means' ile bu parçaların belli bir eşik değeri üzerinde benzerlik taşıyanların birleştirilmesi ile alanların tespit edilmesi aynı biçimde gerçekleştirilir (ayrıntılı bilgi için bkz. bölüm 2.2). Alanlar tespit edildikten sonra yapılan işlemler SimplyCity'den ayrılmaktadır. Şöyle ki alanlar için ilk yapılması gereken, alanların bulanık özellik uzayında yer alan kümelere üye olma derecelerinin hesaplanmasıdır. Bu iş için 'Cauchy' işlevi kullanılmıştır. Alanların kümelere olan üyelikleri tespit edildikten sonra UFM yaklaşımı ile görüntüler arasında benzerlikler hesaplanır.

Sistemde benzerlik, alanlar arası benzerliklerin bir araya gelmesiyle oluşturulur. Burada bezerliklerin bir araya gelmesi matematiksel olarak OR işleci ile ifade edilebilir. Yani görüntülerde sadece konum olarak birbirine karşılık gelen alanların benzerliğine bakmak yerine; bir görüntü alanı diğer görüntünün tüm alanları ile karşılaştırılır. Daha doğrusu bulanık mantık bağlamında; bir görüntünün alanlarından birinin bulanık kümeler üyeliği ile diğer görüntünün tüm alanlarının bulanık kümeler üyeliği, bulanık küme işlemlerine göre ele alınır. Şöyle ki; bulanık küme işlemlerinde bir  $x$  değişkeninin  $A$  ve  $B$  kümelerine ait üyelikleri  $\mu_A$  ve  $\mu_B$  ise;  $x$  değişkeninin aynı anda hem  $A$  hem de  $B$  kümelerine üye olma derecesi,  $A$  ve  $B$  kümelerinin kesişimi yani  $\mu_{A \cap B}(x)$  ile ifade edilir ve  $\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$ 'dir. Bulanık benzerlik ölçüsü  $S(A,B)$  ile gösterilirse  $S(A,B) = \sup \mu_{A \cap B}(x)$ 'dir.

Sistemde  $D$  ve  $F$ , birer görüntü ise bu görüntüler, alanları ve alanlarının benzerliği <bulanık küme, üyelik derecesi> çokluları ile ifade edilir.  $C_d$  ve  $C_f$  sırasıyla  $D$  ve  $F$  görüntülerinin ait olduğu bulanık kümelerin sayısı ise,  $D_i$  ve  $F_j$  görüntülerin alanları olmak üzere  $D = \{D_i: 1 \leq i \leq C_d, i \in \mathbb{N}\}$  ve  $F = \{F_j: 1 \leq j \leq C_f, j \in \mathbb{N}\}$  olarak gösterilebilir. Dolayısıyla  $D_i \in D$  olmak üzere her  $D_i$ 'nin  $F$ 'ye olan benzerliği

$$I_i^D = S_{j=1}^{C_f} (D_i \cup F_j)$$

olarak gösterilebilir[104]. Aslında burada ifade edilmek istenen  $D$  görüntüsünün ait olduğu bir küme yani taşıdığı bir alan,  $F$  görüntüsünün ait olduğu kümeler yani taşıdığı alanlar ile karşılaştırıldığıdır.  $D$  görüntüsünün  $F$ 'ye olan benzerliği olarak ortaya çıkan ifade vektördür ve

$$I^D = [I_1^D, I_2^D, \dots, I_{C_d}^D] \text{ biçimindedir.}$$

Aynı şekilde  $F_j \in F$  olmak üzere her  $F_j$ 'nin  $D$ 'ye olan benzerliği

$$I_j^F = S_{i=1}^{C_d} (F_j \cup D_i)$$

olarak gösterilebilir[104]. Yine yukarıda olduğu gibi aslında yapılan tüm  $F$  alanlarının tek tek tüm  $D$  alanları ile karşılaştırılmasıdır. Bu karşılaştırmadan

çıkan benzerlik te vektördür ve F'nin D'ye benzerliği şu biçimde gösterilebilen vektörel ifadedir:

$$I^D = [I_1^D, I_2^D, \dots, I_{cf}^D]$$

Dolayısıyla D ve F olarak adlandırılan iki görüntünün birbirine benzerliği  $L^{(D,F)}$  olmak üzere

$$L^{(D,F)} = \begin{bmatrix} I^D \\ I^F \end{bmatrix}$$

biçiminde ifade edilebilir[104]. Sistemin bu noktada yaptığı iş, bu vektörel benzerliklerin uygun ağırlıklar ile toplanmasıdır ki; uygun ağırlıkları tayin etme noktasında konumsal ilişkiler devreye girer. Yani görüntüde benzer konumsal durumda ve ilişkilerde olan alanların benzerlikleri uygun ağırlıklar ile öne çıkartılır.

### **2.3. Görüntü Sınıflandırma İçin Kümeleme Kullanarak Otomatik Ontoloji Türetimi (*Automatic Ontology Derivation Using Clustering for Image Classification*)**

Bu çalışma 2003 yılında Texas Üniversitesi'nde, L. Khan ve L. Wang tarafından gerçekleştirilmiştir. Çalışmanın temel hedefi görüntü veri tabanlarında anlamsal erişim ve sorgulamayı; görüntülerin alanına bağlı bir ontoloji ile sınıflandırıp, ontolojideki ilişkileri de kullanarak gerçekleştirmektir. Sistem gerçekleştiriminde iki temel sorun tespit edilmiştir[100]:

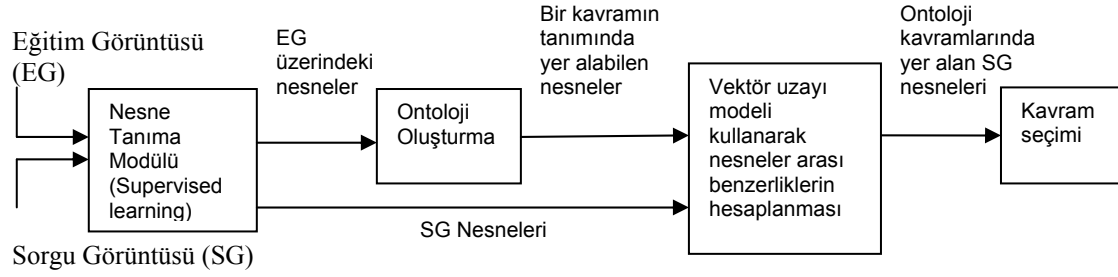
- Görüntünün anlamsal özelliklerinin doğru çıkarılması: Anlamsal özellik olarak tespit edilen kavramların, görüntüyü ifade etmek ya da tanımlamak için kullanılacak kalitede olması gerekir. Seçilen kavramların kalitesinin anma (recall) ve duyarlık (precision) değerlerini doğrudan etkilediği açıktır. Dolayısıyla sistemde, doğru anlamsal özellik tespiti hedeflenmiştir.
- Otomatik ontoloji oluşturulması: Görüntülerin bir ontoloji olmaksızın anlamsal özelliklerinin doğru tespiti mümkün değildir. Bu sebeple vektör

uzayı modeli ve kümeleme tekniği kullanılarak otomatik ontoloji oluşturulması hedeflenmiştir.

Bu çalışmayı gerçekleştiren L. Khan ve L. Wang, çalışmalarının özgün yanı olarak ontoloji kullanarak görüntü sorgulama ve erişimini gerçekleştiren ilk çalışma olduğunu tespit etmişlerdir. Ancak burada hemen bir noktayı belirtmekte fayda görülmektedir. 1994 Yılında Glasskow Üniversitesi'nde başlatılan FERMI projesi ontoloji terimini kullanmamakla birlikte erişim ve sorgulama biçimiyle ilk çalışma olarak tespit edilebilir.

Khan ve Wang, bu çalışmada ontolojilerini görüntülerin benzerliği üzerine inşaa etmişlerdir. Benzerlik araştırmasında renk ve şekil özellikleri bir arada kullanmışlardır.

Çalışmanın genel görüntüsü şu biçimdedir[100]: Sistemde ilk olarak görüntüler üzerinde yer alan nesnelere tanıma işlemi gerçekleştirilir. Khan ve Wang'ın bu çalışmalarının temel bileşenlerinden biri görüntüler üzerinde yer alan nesnelere tanınmasıdır. Burada benimsenen yöntem kenarların ve alanların tespit edilip yapay sinir ağlarında denetlenmiş (supervised) sınıflayıcı tekniği kullanarak, önceden tanımlı nesne adlarından birine eşleştirilmesidir. Nesnelere belirlendikten sonra her nesne için renk ve şekil özelliklerini vektörel olarak ifade eden üst veri oluşturulur. Bu noktada nesnelere arası benzerlik, vektör uzayı modelinde uzaklıkların hesaplanmasıyla ortaya çıkabilecektir. Hedeflenen ontolojide her bir kavram bir grup nesne ile tanımlanacaktır. Bu sebeple görüntü kümeleme yaklaşımında önce nesnelere kümelenir ve her bir nesne kümesine ağırlık atanır. Daha sonra her bir görüntü için, üzerinde bulundurduğu nesnelere ve ağırlıklarına göre vektörel üst veri oluşturulur. Bu noktada iki görüntü arasındaki benzerliği yine vektör uzayı modelinde yer alan uzaklık yaklaşımı ile hesaplamak mümkündür. Son olarak görüntülerin birbirlerine olan benzerliklerine göre hiyerarşik bir ontoloji elde etmek amacıyla sıradüzensel toplayıcı (agglomerative) kümeleme algoritması kullanılır. Şekil 2.2'de sistemin iş akışı görülmektedir[100]:



Şekil 2.2: Khan ve Wang 'ın çalışmalarının genel sistem akışı

Çalışmanın genel hatlarından da fark edileceği üzere metinler için geliştirilen klasik IR (Informatin Retrieval) yaklaşımı, görüntülere uygulanmıştır. Nasıl ki metinler için terimler ve terimlerin gerek derlemdeki gerekse belgedeki sıklığına bağlı olarak hesaplanan ağırlıklar ile vektör uzayı oluşturuluyorsa; bu çalışmada da nesnelere ve nesnelere görüntülerde yer alma sıklığına göre ağırlıkları oluşturuluyor ve metin belgelerinin <terim, ağırlık> çokluları, görüntüler için <nesne, ağırlık> çoklularına dönüştürülmüş oluyor. Dolayısıyla N adet görüntü içeren bir veri tabanında M adet nesne çıkarılmış ise, bu nesnelere renk ve şekil benzerliklerine göre t adet anlamsal küme ayrılacaktır. İki nesnenin benzerliğinde; şekil ve renk özelliklerinin benzerlik toplamlarındaki etkileri de ağırlıklandırılmıştır. Yani  $G_i$  ve  $G_j$  iki ayrı nesne olmak üzere; benzerlikleri şu biçimdedir[100]:

$$\text{Sim}(G_i, G_j) = \text{Sim}_{\text{renk}}(G_i, G_j) \times \text{Ağırlık}_{\text{renk}} + \text{Sim}_{\text{şekil}}(G_i, G_j) \times \text{Ağırlık}_{\text{şekil}}$$

Kullanılan Sıradüzensel Toplayıcı (Hierarchical Agglomerative) Algoritma şu biçimdedir[100]:

1. Her bir görüntüyü bir iskelet küme yerleştir ve tüm iskelet kümeler için kümeler arası uzaklığı hesaplayarak artan sırada listele
2. Birbirine en benzer iki küme tespit et birleştir ve yeni oluşan kümeyle birlikte kümeler arası uzaklığı yeniden hesapla ve artan sırada listele
3. Birden çok sınıf olduğu sürece 2. adımı tekrar et.

Khan ve Wang'ın çalışmasında üretilen ontoloji, sadece ata-çocuk ilişkisini tespit edip kullanmaktadır. Ata ve çocuk ilişkisinin tespitinde kullanılan mantık şu biçimde özetlenebilir: ata kavram, çocuk kavramların özelliklerinin bir kısmını mutlaka taşıyacaktır.

## 2.4. Resimlerin ve Kelimelerin Anlamlarının Öğrenilmesi (*Learning Semantics of Words and Pictures*)

Bu çalışma 2000 yılında Berkeley Üniversitesi'nde K. Barnard ve D. Forsyth tarafından gerçekleştirilen bir çalışmadır. WordNet'i kullanarak erişimi destekleyen en etkin sistemdir. Bu sebeple tez metni kapsamına alınmıştır. WordNet adı verilen İngilizce için hazırlanmış olan kavramlar arası ilişki sözlüğünü kullanmasıdır. WordNET, sıradüzensel ilişkileri ele alan bir ağaç gibi düşünülebilir. Ağaçta alt çocuk düğümlerin, üst ata düğümlere ait bir parça ya da tür olduğunu düşünmek mümkündür.

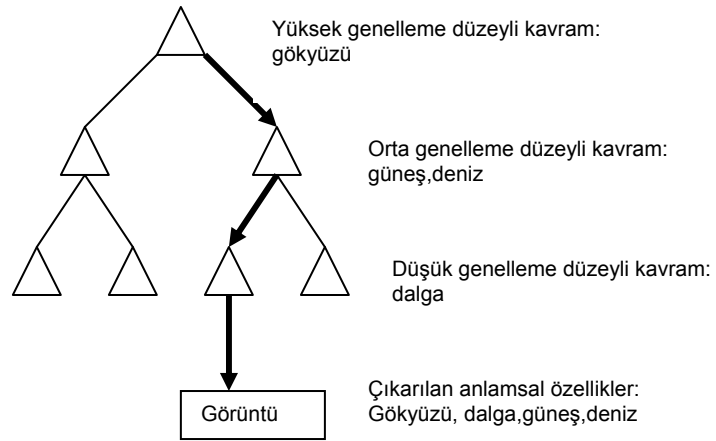
Barnard'ın çalışması görüntüden çıkarılan görsel ve anlamsal özellikleri kullanarak dizinleme, erişim ve seçme hizmetleri vermeyi hedeflemiştir. Sistem görüntüler üzerinde denetlenmemiş (unsupervised) öğrenme ile nesne tanıma işlemini gerçekleştirmektedir. Sistem, görsel ve anlamsal özelliklerin bir arada bulunuşlarına (co-occurrence) göre modelleme yapmaktadır. Barnard çalışmasında görüntüleri, özelliklerine göre sıradüzensel bir yapı ile ele almayı tercih etmiştir. Sıradüzensel yapının hem genelleme seviyelerini ifade etmekte güçlü olduğu hem de seçme eyleminin doğasına uygun olduğu düşünülmüştür[99].

Barnard'ın sorgulama tercihi daha çok seçmeden yana olmuştur. Anahtar kelime ya da örnek ile sorgulamak yerine kurduğu sıradüzensel yapıyı bir ağaç gibi ifade edip, her bir düğümden bir örnek sunarak seçme eylemini desteklemeyi tercih etmiştir. Bu tarz sorgulamanın doğru soruyu oluşturmak için en iyi yöntem olduğuna inanmıştır. Dolayısıyla Barnard'ın sisteminde geri bildirim yoktur[99].

Barnard, görüntüler için önerdiği modeli üretici sıradüzensel model (generative hierarchical model) olarak adlandırmaktadır. Bu modelde görüntüleri kümelere dağıtmak için asimetrik kümeleme ve özelliklerin dağılımını eklemlendirmek için simetrik kümeleme, sıradüzensel bir kombinasyonla kullanılmıştır. Modelden anlaşılacağı gibi görüntüler ağaç türünde sıradüzensel bir yapıda tutulurlar ve ağacın yaprakları görüntü kümelerini taşır. Kümelerde görüntülerin görsel özellikleri ile anlamsal özelliklerinin bir kısmı saklanır. Daha doğrusu anlamsal özellik olarak tespit edilen kavramların genelleme seviyeleri WordNET aracılığı ile



tespit edilir ve en az genel kavramlar yapraklardaki kümelerde kalırken daha genel kavramlar ağacın iç düğümlerinde yer alır. Dolayısıyla bir kavrama ait görüntüler, o kavramın alt ağacındaki tüm kümelerdir. Şekil 2.3, sözü edilen ağaç yapısı için örnek teşkil etmektedir. Şekilde yer alan görüntüde güneş, deniz, dalga ve gökyüzü kavramlarının anlamsal özellik olarak çıkarıldığı kabul edilsin. En basit kavram olan dalga en alt genelleme seviyesinde kalırken daha genel güneş ve deniz bir üst seviyedeki düğümün anlamsal özelliği olmuştur ve bu ikinci düzey de bir üst seviyede anlamsal özelliği gökyüzü olan düğümün çocuğu olmuştur[99].



Şekil 2.3. Barnard'ın sisteminin kullandığı sıradüzensel ağaç görünümü

Barnard'ın modelinde bir görüntünün sadece bir kümeye ait olması beklenmemektedir. Bir miktar bulanık mantık modeli kullanılmıştır bile denilebilir. Dolayısıyla görüntülerin üyeliklerinin yettiği her kümeye, üyelikleriyle doğru orantılı bir ağırlıkla kayıt edildiği söylenilebilir.

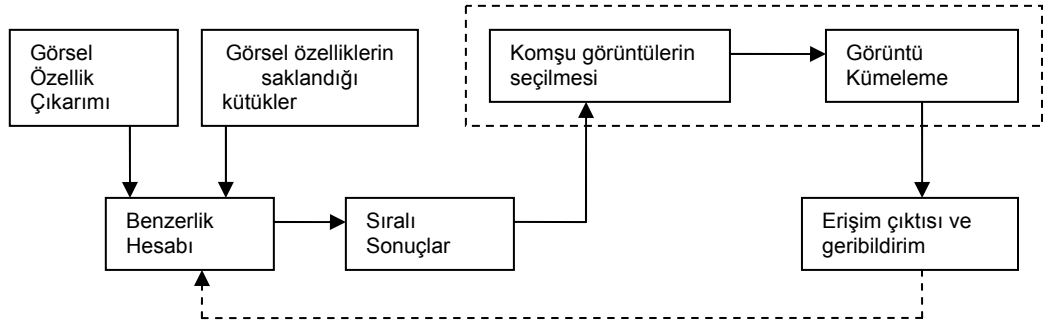
Barnard'ın sisteminde otomatik etiketlendirme için gereken öğrenme beklentilerin enüste çıkarılması (expectation maximization) algoritması ile gerçekleştirilmiştir. Görsel özellik olarak renk, dokuma, şekil ve uzaysal ilişkiler kullanılmıştır[99].

Son olarak Barnard'ın sisteminde, görüntülerde tanınan nesnelere için, geleneksel IR yaklaşımında yer alan durma listesi özelliği mevcuttur. Yani görüntülerde sıkça yer alan nesnelere, dizinleme ve erişim sürecinde ihmal edilmiştir[99].

## 2.5. Kümelemeyle İçerik Tabanlı Görüntü Erişimi (*Content Based Image Retrieval by Clustering -CBICR*)

CBICR, Y. Chen, J. Wang, R. Krovetz tarafından Pennsylvania State Üniversitesi'nde, 2003 yılında gerçekleştirilmiş içerik tabanlı görüntü erişim sistemidir. Görsel özellikler ile sorgulamanın hem pratik olmayan yönlerine hem de görsel özellik ile kullanıcının isteğini tarif edemeyeceği, tespitine dayanarak; görsel özellikler ile anlamsal özellikler arasındaki boşluğu kümeleme ve denetlenmemiş (unsupervised) öğrenme ile doldurmayı hedeflemiş bir çalışmadır. CBICR'nin temel yaklaşımı aynı anlamsal özellikleri taşıma eğilimi olan görüntülerin, bir özellik uzayında kümelenebileceğidir[112]. Böylesi bir kümeleme yaklaşımı ile kullanıcıya benzerliğe göre sıralı bir listeyi erişim çıktısı olarak sunmak yerine; tüm küme erişim çıktısı olarak sunulacaktır. CBICR'de kümeler dinamiktir. Yani kümeler sorguda alınan örnek görüntüye göre oluşturulmaktadır.

CBICR Sistemi'nin, yukarıda anlatılan genel özellikleri, şekil 2.4'te gösterilmiştir[112].



Şekil 2.4: CBICR'nin genel sistem görünümü

CBICR'nin tipik bir CBIR sistemine göre iki önemli özelliği vardır[112]:

1. Tipik CBIR sistemlerinde kullanıcı sorgusu olan görüntünün, veri tabanında saklanan görüntüler ile benzerliği hesaplanarak azalan sırada en iyi k tanesi kullanıcıya yönlendirilir. CBICR'de ise sorguda verilen görüntü, dinamik olarak kümelemeyi tetikler. Sorgu ile benzer görüntülerin yanında benzer görüntülere benzeyen diğer görüntülerden bir küme oluşturulur ve kullanıcıya yönlendirilir.

2. CBICR'nin asıl önemli kısmı kümeleri oluşturduğu modülleridir. Bu modüllerin girdisi ise veri tabanındaki görüntülere ait görsel özelliklerin saklandığı kütüklerdir. Dolayısıyla birçok CBIR sisteminde varolan farklı özellik çıkarma biçimlerine adapte edilerek kullanılması kolayca mümkündür.

CBICR'de benzerlik ve uzaklık birbirinin tersi kavramlar olarak kullanılmış ve kümelemede uzaklık esas alınmıştır. Ayrıca  $i$  ve  $j$  iki görüntü ve  $d$  uzaklık olmak üzere  $d(i,j)=d(j,i) \geq 0$  olarak ele alınmıştır.

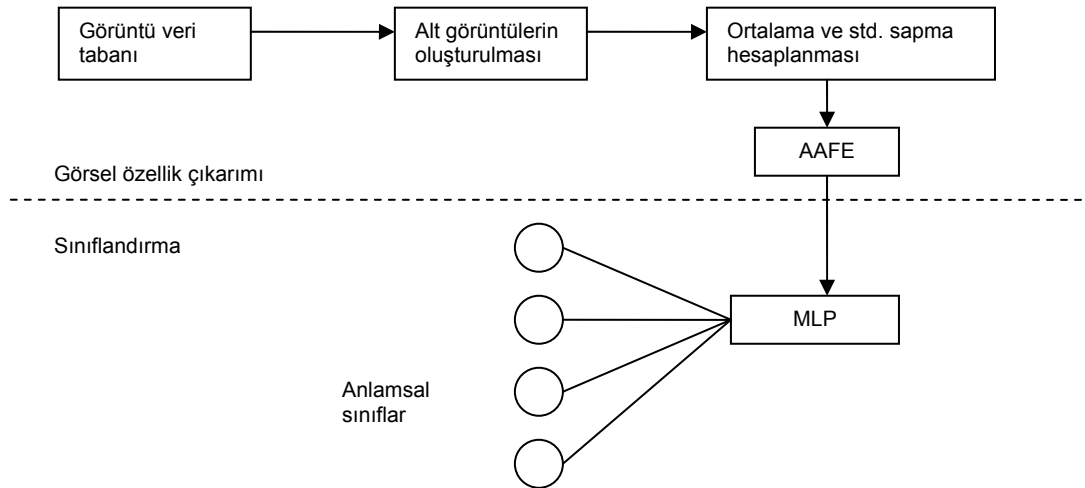
CBICR'de sorgu görüntüsü ( $i$ ) alınıp görsel özellikleri çıkarıldığında  $i$ . tohum olarak kabul edilir ve tohuma en benzer  $k$  adet komşu görüntü tespit edilir. Daha sonra seçilen bu komşular yeni tohumları oluşturur ve bu tohumlara en benzer komşular tespit edilir. Bu yinelemenin sonu sezgisel olarak tanımlanmıştır ve çeşitli sayılar ile sistem sınınamaya alınmıştır. Genel olarak tohumların tüm komşuları kümeye zaten dahil edilmiş olması yinelemenin sonlanma koşuludur.

Görüntülerin gösterimi için ağırlıklandırılmış çizge kullanılmıştır. Bu çizge  $G$  ile gösterilirse  $G = (V, E)$ 'dir.  $V$ ,  $n$  adet görüntüyü gösteren düğümler olmak üzere  $V = \{1,2,\dots,n\}$ 'dir.  $E$  ise düğümler arası uzaklığı gösterir ve  $i, j$  görüntüleri için  $w_{i,j} \geq 0$  değerini taşır. CBICR içinde uzaklıklar akrabalık matrisi (affinity matrix) adı verilen iki boyutlu yapıda saklanır.  $i$ . Satır ve  $j$ . Sütün,  $i$  ve  $j$  görüntüleri arasındaki uzaklığı gösterir[112].

Görüntüler için çizge gösterimi kullanıldığında kümeleme problemi, çizge problemine dönüşür. CBICR'de çizge parçalama (partitioning) problemi "Ncut" algoritması ile aşılmıştır. "Ncut" algoritmasının ne kadar yineleneceği yani çizgenin ne kadar alt çizgelere ayrılacağı CBICR içinde sezgisel olarak belirlenen bir  $T$  eşik değeri ile belirlenmiştir. Başka bir deyişle, çizgede  $T$  adet alt çizge olduğunda parçalama işlemi sona erer[112].

## 2.6. İçerik Tabanlı Görüntü Erişim Sistemleri İçin Akıllı Karma Yaklaşım (An Intelligent Hybrid Approach for Content Based Image Retrieval - HACBIR)

HACBIR, 2002 yılında Kanada'nın Nipissing Üniversitesi'nde S. Kulkarni ve B. Verma tarafından gerçekleştirilmiştir. Sistemin genel özellikleri daha önce incelenen sistemlere benzer olarak görsel özellik çıkarma, görsel özelliklere göre görüntüleri anlamsal sınıflara ayırma ve örnek görüntü ile anlamsal sınıfları kullanarak sorgulamaya imkan sağlama olarak sıralanabilir. HACBIR, sadece dokuma özelliğini kullanan bir sistemdir. Özellik çıkarımı için otomatik birleştirici sinir ağları (auto associative neural network-AANN) tekniği kullanılmıştır[109]. Sınıflandırma için çok katmanlı algılayıcı (multi layer perceptron-MLP) tekniği kullanılmıştır[109]. HACBIR, hem AANN hem de MLP'yi birlikte kullandığı için karma yaklaşım kategorisine girmektedir. HACBIR, karma yaklaşımların tipik bir örneğidir. HACBIR için anlamsal sınıflar, doğal dilin kavramlarına karşılık gelmemektedir. Her bir anlamsal sınıf, dokuma özelliğine göre bir tür ifade etmektedir. Örneğin pürüzlü, düz, tanecikli, kabarcıklı vb. gibi. Sistemin genel görünümünün açıklamalardan önce şekil 2.5 ile ifade edilmesi uygun görülmüştür[109].



Şekil 2.5: HACBIR'in genel görünümü

Otomatik birleştirici özellik çıkartıcı (auto associator feature extractor-AAFE) fikri, girdinin ve çıktının aynı örüntü olduğu girdi:saklı:çıkı eşleşmesine dayanmaktadır. AAFE, aynı örüntüleri öğrenme çabası sergilerken gizli katmanı aracılığı özellik

vektörünü çıkarır. AAFE aslında tek gizli katmanlı ileri doğru beslemeli yapay ağıdır. AAFE’de n adet girdi var ise n adette çıktı ve p adet gizli birim olacaktır. AAFE’de yapay ağ ‘supervised’ öğrenme ile eğitilir. Eğitim tamamlandığında gizli katmandaki değerler, görsel özellik vektörleri olarak kabul edilir ve MLP’nin gizli katmanını beslerler[109].

MLP’nin de AAFE’de olduğu gibi n adet girdisi ancak 32 adet elle yaratılmış anlamsal sınıftan oluşan çıktısı vardır. MLP’nin girdisi görüntünün her satırındaki piksellerden elde edilen ortalama (mean) ve sapma (deviation) değerleridir. Bu değerler görüntü için görsel özelliği ifade eder ve gizli katmandaki değerlere göre 32 sınıftan birine görüntü atanır[109].

Sistem farklı sayıda gizli katman ve öğrenme yinelemesi kullanılarak kendi tarzındaki çalışmalarla test edilmiştir. Elde edilen sonuçlar şöyledir[109]:

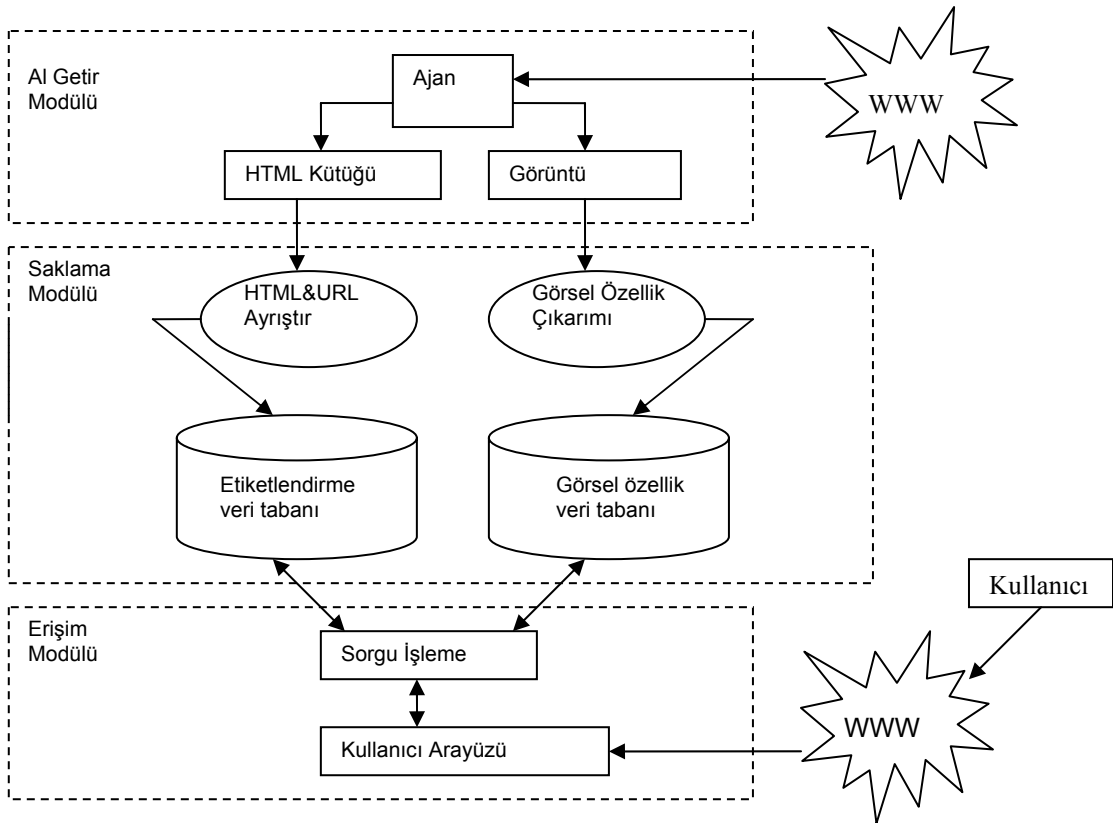
1. AANN, özellik çıkarımı için iyi bir yöntemdir. Ancak MLP için bariz bir iyilik söz konusu değildir.
2. Ancak MLP’nin kendinden bir önceki Kohonen çalışmasına göre daha iyi sonuçlar ürettiği gözlemlenmiştir.
3. Yapay sinir ağı yaklaşımı, geleneksel IR yaklaşımlarından daha iyi sonuç üretmiştir.

## **2.7. İçerik Tabanlı Erişimi Destekleyen Anlamsal Şablon Uygulaması (Apply Semantic Template to Support Content Based Image Retrieval - STSCBIR)**

STSCBIR, Y. Zhuang, X. Liu, Y. Pan tarafından Çin’in ZheJian Üniversitesi’nin Yapay Ağ Enstitüsü’nde, 2000 yılında gerçekleştirilmiştir. STSCBIR önerisi tespit edilen iki sorunu çözmek için getirilmiştir[113]. Bu sorunlardan ilki, görüntülerin görsel özellik vektörü ile anlamsal özellikleri arasındaki boşluğun, görsel özelliklerden hareketle doldurulamayacağıdır. Bu sorunu aşmak için geleneksel CBIR yaklaşımına anlamsal şablon desteği verilmiştir. İkinci sorun ise geleneksel CBIR sistemlerinde görüntü benzerliklerinin vektör benzerliğini aşmamasından kaynaklanan kullanıcı yorumunu ihmal eden yaklaşımıdır. Bu da etkin bir geri

bildirim modeli ile kullanıcı değerlendirmelerinin sisteme dahil edilmesi ile aşılmaya çalışılmıştır.

STSCBIR’de uygulama ortamı olarak internet seçilmiştir. İnternet ortamında görüntü içeren web sayfaları ajanlar tarafından indirilip html kütüğü ve görüntüler olarak ikiye ayrılır. Sistem de görüntünün etiketlenilmesi için iki yaklaşımla gerçekleştirilir. İlk olarak web sayfasının URL (Uniform Resource Location) adresi görüntünün içeriği için fikir verici kabul edilip, ayrıştırılır. Buradan elde edilen kavramlardan beklenen, en azından görüntünün ait olabileceği kategorileri tespit etmektir. İkinci yaklaşım görüntünün yakınında bulunan metin parçalarını yani paragrafları geleneksel IR yaklaşımına göre işleyerek anlam gücü yüksek kavramları tespit etmektir. Tespit edilen kavramlar ve URL’den ayrıştırılanlarla birlikte görüntünün anlamsal özelliklerini oluşturacaktır. STSCBIR sistemin genel mimarisi şekil 2.6’da gösterilmiştir[113]:



Şekil 2.6: STSCBIR sistemin genel mimarisi

STSCBIR’de, renk ve dokuma görsel özellik olarak kullanılmıştır. Renk için her piksel RGB uzayından HSV uzayına çevrilmiş, H ve S ağırlıkları kullanılarak 2

boyutlu bir vektör oluşturulmuş ve 32 adet normalize edilmiş renk değeri ile renk vektörü oluşturulmuştur. Dokuma özelliği içinse karşıtlık (contrast), işlenmemişlik (coarsness) ve yön (direction) özelliklerinden oluşan bir vektör kullanılmıştır. Böylece STSCBIR içinde bir görüntü için 35 değerin birleşiminden oluşan bir vektör söz konusudur[113].

STSCBIR içinde anlamsal şablon, etkileşimli geri bildirim ile oluşturulur. Şöyle ki; sisteme sorgu olarak zaten kayıtlı olan örnek bir sorgu sağlandığında; zaten anlamsal özelliklerine ayrıştırılmış bir görüntü olacaktır ve kullanıcı isterse örnek görüntünün yanında sorgusuna kavram da ekleyebilir. Sistemde her kavram için kavramı içeren görüntülerin görsel özelliklerinden bir görsel özellik vektör merkezi oluşturulur. İlk sorguya karşılık olarak, kavrama ve görsel özelliklerine göre ilk erişim çıktısını üretir. Kullanıcı, sunulan erişim çıktısını inceleyip, her bir görüntünün sorgusu ile ilgililik derecesini geri bildirim ile ifade eder. Sistem, geri bildirimde göre kavramları ve görsel özellik vektör merkezinin ağırlıklarını güncler. Bu etkileşimli diyalog kullanıcı tatmin olana dek sürer. Bu sürecin farklı kavramlar için yinelenmesi, o kavramlar için de anlamsal şablonun oluşturulmasını sağlar. Dolayısıyla anlamsal şablon ST ile gösterildiğinde  $ST = \{C, F, W\}$  olarak gösterilebilir. C, şablondaki kavramı, F, kavrama ait görsel özellik vektör merkezini ve W'de vektörün ağırlığını ifade eder. W'nin ST çöklusunda yer almasının sebebi, kullanıcıların görsel özellik gösterimi ve kullanımı konusunda olabilecek farklı tercihlerini modellemek içindir[113].

STSCBIR içinde renk özelliği için öklidyen uzaklık, dokuma özelliği için gaussian dağılımı kullanılmaktadır.

STSCBIR, görüntüleri etiketlendirmek için tespit edilen kavramların birbirleri ile ilişkilerini yakalamak için WordNET'i kullanır. STSCBIR'ın WordNET'ten aldığı destek sıradüzensel, parça - bütün ve eş anlamlılık ilişkilerinin tespitidir. Böylece sistem sorguda girilen kavramı ya da sorguda verilen örnekten çıkarılmış olan kavramı, WordNET'ten aldığı diğer ilişkili kavramları da ekleyerek genişletebilmektedir[113].

## **2.8. Web Üzerinde İçerik Tabanlı Erişim İçin Çizim İle Sorgulama ve Geri Bildirim (*Content-based Image Retrieval over the Web using query by Sketch and Relevance Feedback - DrawSearch*)**

Drawsearch<sup>2</sup>, İtalya'da Bari Politecnio'da, 1999 yılında, E.D: Sciascio, G. Mingolla, M. Mongiello tarafından geliştirilen prototip bir CBIR sistemidir. Genel özelliği renk, şekil, dokuma özelliklerini kullanarak; geri bildirim ile etkileşimli bir erişim sürecini gerçekleştirip kullanıcının çizerek hazırladığı sorgularla en ilgili görüntülere Web ortamında erişmektir. Sistemin dizinleme ve erişim biçimi herhangi bir yenilik içermemesine rağmen hedeflediği ortam bakımından ilgi çekmiştir. Zira özellikle 2000 öncesi çalışmalarda görülen yerel veri tabanlarında çalışma tarzının dışına çıkmıştır. Drawsearch, 2000 öncesi çalışmaların ortak tarzı olan görüntüler arasında gezinerek sorgulama işlemini uzun sürmesini eleştirmiş ve önüne koyduğu sorunu görsel sorgulama süresini kısaltmak olarak koymuştur. Zira Drawsearch taraftarlarına göre gezinerek arama internet ortamı için uygulanamaz bir yaklaşımdır.

DrawSearch, istemci-sunucu tarzında bir mimariye sahiptir. Sunucu tarafında internet ortamından elde edilen dizinlemiş görüntüler mevcuttur ve dizinleme işleminin sunucu açısından maliyeti çok düşünülmez. Zira kullanıcıya yansımayacak bir zaman diliminde gerçekleştirilir. Oysa sorguların görsel özelliklerinin çıkarılma maliyeti doğrudan kullanıcıların cevap alma zamanını etkiler[105].

DrawSearch iki farklı arayüz içermektedir. İlk arayüzde kullanıcı şekil ve renk özelliklerini kullanabilirken ikinci arayüzde sadece dokuma özelliğini seçebilmektedir[105].

DrawSearch'te bütün görsel özellikler için üretilen üst veriler vektörelidir ve aslında DrawSearch metin erişim sistemlerinde kullanılan vektör uzayı modelini kullanmaktadır. Yani terimlerin belgeler için taşıdığı anlam, görsel özellik vektörleri

---

<sup>2</sup> <http://decom03.poliba.it/DrawSearch/DrawSearchhome.html>



ile görüntüler arasında vardır. Sadece bir farkla ki, görsel özellik vektörlerinin görüntü için ağırlığı yoktur[105].

DrawSearch'te renk özelliği RGB modelinde ifade edilir. Şekil özelliğini tespit etmek bir bölümlenme problemidir. Renk ve dokuma değerlerini girdi olarak kullanan 'contour-based' bölümlenme algoritması kullanılmıştır. Bölümlere ayırma işlemi gerçekleştirildikten sonra Fourier dönüşümleriyle şekil özellikleri ifade edilmiştir. Dokuma özelliği de bölümlenmeyi gerekli kılmıştır. Dokumada hem bölümlenmek hem de vektör elde etmek için Gaussian Markovian Random Fields yaklaşımı kullanılmıştır[105].

DrawSearch'ün erişim modelini açıklamak gerekirse; renk ve şekil özelliklerin kullanılarak yapılan eşleştirmede; R ve Q görüntülerinin renk benzerliği  $SimC(R,Q)$  ile, şekil benzerliği  $SimS(R,Q)$  ile gösterilsin. R ve Q görüntülerinin vektörleri çoklu elemanlarını oluşturacak şekilde  $R = \{r_0, r_1, r_2, \dots, r_n\}$  ve  $Q = \{q_0, q_1, q_2, \dots, q_n\}$  ile gösterilirse bir görsel özellik üzerinden benzerlik şu biçimde ifade edilebilir[105]:

$$Sim(R, Q) = \sum r_i q_i / \sqrt{\sum r_i^2 \times q_i^2}$$

Toplam benzerlik tüm görsel özelliklerin benzerliklerinin toplamıdır ve şu biçimde gösterilebilir[105]:

$$Sim(R,Q) = \alpha \times SimC(R,Q) + \beta \times SimS(R,Q)$$

Formülde  $\alpha$  ve  $\beta$  görsel özelliklerin erişim sürecindeki ağırlıklarıdır. Bu ağırlıkları geri bildirimle değiştirilebilir.

Görüntüler arası benzerlik karşılaştırılması, görüntülerin yoruma açık olma özelliklerinden dolayı bir miktar belirsizlik taşır. Bu sistem, varolan belirsizliği geri bildirimle en aza indirmeyi tercih etmiştir. Sistemde, kullanıcı elde ettiği erişim çıktısından ilgili olarak değerlendirildiği görüntüleri işaretleyerek olumlu geribildirimde bulunabilir. İşaretlenmeyen görüntüler olumsuz olarak tespit edilir. Şu durumda ilgili bulunan görüntülerin görsel özellik vektörlerini orijinal sorguya ait vektörlere ekleyerek ve ilgisiz bulunan görüntülerin görsel özelliklerini de çıkararak yeni sorgu oluşturulur. Eğer Q sorgusuna ait erişim çıktısında yer alan N adet görüntü ilgili ve

M adet görüntü ilgisiz olarak yorumlandı ise ve X ilgililerin, Y ilgisizlerin vektörü ise yeni sorgu şu biçimde ifade edilebilir[105]:

$$Q^{(k+1)} = Q^{(k)} + \delta \sum_{i=1}^N X_i - \varepsilon \sum_{i=1}^M Y_i$$

Sistemde, dokuma özelliği için olumlu geri bildirim özelliği dikkate alınmaksızın sadece olumsuz geri bildirim etkisi dikkate alındığında daha iyi sonuçlar elde edildiği vurgulanmıştır.

## **BÖLÜM 3. ANLAMSAL WEB VE ONTOLOJİ**

Geliştirilen sistem anlamsal özellik çıkarımında, alana bağlı bulanık ontoloji kullanılmaktadır. Bu özelliği sebebi ile anlamsal web ve ontoloji konu başlıkları ile yakın bir ilişkisi kurulmuştur. Bu bölümde anlamsal web, anlamsal web için anahtar teknoloji olan ontolojinin tanımı ve bu alandaki çalışmaların tanıtımı yapılacaktır.

### **3.1. Anlamsal Web (Semantic Web)**

World Wide Web (WWW) kullanımı dünya üzerinde öylesine bir yaygınlık kazanmıştır ki, bugün bilgi erişim etkinliklerini WWW kullanmadan karşılamayı düşünmek bile güçleşmiştir. Bilgi erişimini elektronik ortamda sağlayan WWW'e dahil olmuş çok sayıda sayfaya her gün yenileri katılmakta ya da varolan sayfalar sürekli değişerek dinamik bir bilgi akışını sağlamaktadır. WWW'in bilgi teknolojilerine en büyük katkısı paylaşılabilirlik ve hız olmuştur.

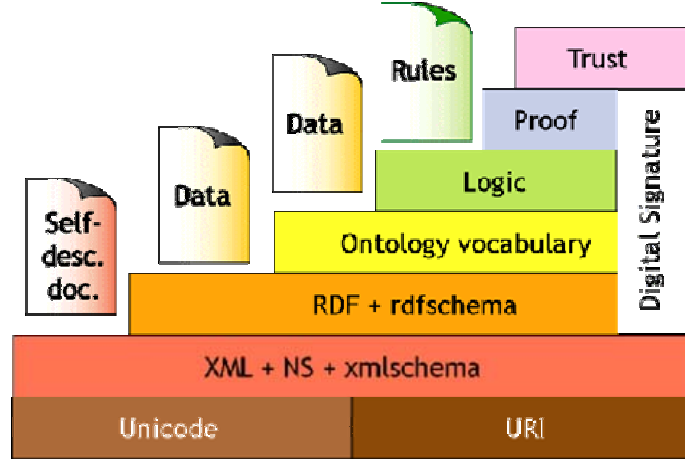
Günümüzde WWW'de yer alan sayfaların büyük bir kısmı HTML ile hazırlanmıştır. Bu dil ile sayfaların gösterimi standartlaştırılmış ve tarayıcıların hizmetine açılmıştır. Bu hali ile bile WWW'nin paylaşımına verdiği katkı çok açıktır. Ancak bu paylaşım, gösterimi ortak olan sayfaların dünyanın her yanından erişilebilmesi noktasındadır. WWW için bir sonraki adım paylaşımın tanımını ve kapsamını genişletmek yani anlamı (meaning) ve anlayışı (undersitanding) paylaşımına açmak olarak tanımlanmıştır[122]. Aslında bu cümle anlamsal web'in amacını çok net ifade eder.

Günümüzde birçok arama motoru sorgulara cevap vermekte ve bilgi ihtiyacını doğru tatmin etmeye çalışmaktadır. Genelde anahtar kelime tabanlı olan bu araçlara Altavista, Yahoo ve Google örnek teşkil eder. Yani HTML ile gösterimi yapılmış web sayfalarının içinde sorgu terimlerinin kullanılıp kullanılmaması, sorgu ile sayfa benzerliği açısından en önemli kriteri oluşturur. Bu yaklaşıma eş anlamlılar sözlüğü, birbiri ile ilgili terim kümeleri ya da ilgili doğal dile ait terimlerin sıradüzensel gösterimi destek olarak tanımlanabilir. Ayrıca terimlerin belge içinde konumlanması da ayrı ayrı ele alınabilir. Örneğin öz kısmında geçen bir terim ya da sayfa adresinde geçen terim, sıklığının yanında konumundan kaynaklanan yeni ağırlıklara sahip olabilir. Bu yaklaşımın en güzel ve başarılı sayılabilecek örneği

Google'dır. Tüm bu ek önlem ve iyileştirmelere karşın bu yaklaşımın tanımlanmış en temel problemi yüksek anma, düşük duyarlık olarak özetlenebilir [148]. Ayrıca sadece anahtar kelime kullanmak, erişimi kullanılan sözlüğe aşırı duyarlı kılar. Özellikle eş anlamlılar sözlüğü ve sıradüzensel terim gösterimi bu duyarlılığı kırmak için ortaya çıkmıştır. Bunlara ek olarak erişimin tek veri türü metin için gerçekleştiğini belirtmek gerekir. Sonuçta omurgayı anahtar kelimeler oluşturur ve anahtar kelime adı verilen kavram, metni oluşturan terimlerin bir alt kümesidir. Dolayısı ile "Ben Bilgisayar Mühendisliği Bölümü'nde profesörüm" ifadesi ile "Beni Bilgisayar Mühendisliği Bölümü'nün bir profesörü olarak düşünebilirsiniz" ifadesi arasında neredeyse hiç fark kalmaz. Çünkü HTML neyi gösterdiği ile değil nasıl göstereceği ile ilgilenir. Dolayısı ile her iki örnek ifade, HTML için bir paragraf parçası olarak ele alınıp gösterilmesi muhtemeldir.

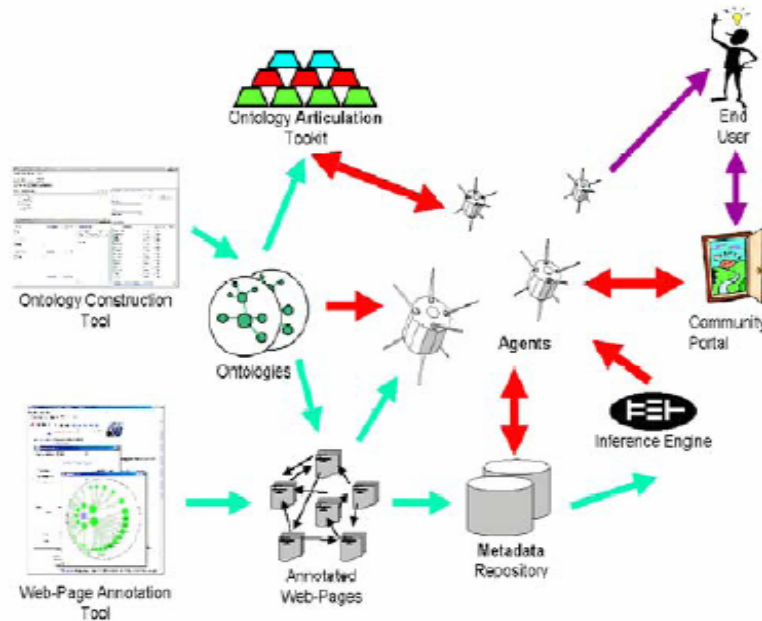
Bu noktada tanımlanan ikinci paylaşım hamlesi olan anlamı paylaşma amacı göz önüne gelir ve sayfaların içeriğini makine anlayışına açmak gerekliliği söz konusu olur. Bu da WWW için ikinci evrimin adı olan Anlamsal Web'in tanımlanmasını sağlamıştır. Anlamsal Web'in tanımı da WWW'nin mucitleri arasında olan Tim Berners-Lee tarafından yapılmıştır: "Anlamsal web, insanların ve bilgisayarların birbiri ile anlaşmasını kolaylaştırmak için, bilginin anlamı ile birlikte tanımlanmasını sağlayan, bugünkü web için bir eklentidir" [131,148]. Bu eklentiye gerçekleştirebilmek için Anlamsal Web'e eklenen üç yeni teknoloji sırası ile üst veri (metadata), ontoloji (ontology) ve güven (trust) olarak sıralanabilir. Sırası ile üst veri, veri hakkında içeriği ve niteliği tutan veri olarak açıklanabilir. Anlamsal web'de amaç verinin insan ve bilgisayarca benzer biçimde anlaşılması ise üst veri bu noktada verinin bu benzer algılanışına hizmet edecektir. Yani bilgisayarlar verinin orijinal biçimine ve gösterimine bakmadan bu verinin içeriğini, karakteristiklerini ve niteliğini üst veriden çıkarsayacaktır fikri verinin özeti olan başka bir veriyi devreye sokar. Yani insan ile bilgisayar iletişimi için bir tanım oluşturulur. Ancak üst verinin içindeki verinin de ki buna sembol denirse diğer semboller ile ilişkisi tanımlanmalıdır. Böylece bilgisayarlar tarayarak ya da ilişkiler kurarak üst verinin ne ifade ettiğini bilebilir (know). Bu düzenlemeye ontoloji adı verilir. Ontoloji, bilgisayarlara veriyi toplu, kapsamlı biçimde işleme yani bilgi üretme şansı verir. Üçüncü teknoloji güven, bilginin güvenilirlikle elde edilebilmesidir. Gerçek hayattaki doğruluk değerleri her zaman sadece doğru ya da yanlış seçimi ile üretilmez.

Güven özelliği ile WWW bilginin doğal anlaşılma biçimi ile anlaşılması yolunu açar. Bu adımlar şekil 3.1’de katmanlı kek (layered cake) olarak Tim Berners-Lee tarafından bildirilen orijinal hali ile yer almaktadır[131,132,135,148].



Şekil 3.1. Anlamsal Web'in katmanlı kek görünümü

Şekil 3.1’de bugünkü Web’den Anlamsal Web’e giden yol haritası da çizilmiştir. En altta yer alan iki katman bugünkü Web’in durumudur. Yani bilgilerin XML ya da HTML ile sunulup, gösterildiği vurgulanır. Üçüncü katman üst veri üretimini, dördüncü katmansa üst verilerin ontoloji ile düzenlenmesini ifade eder. Son üç katman Web’de elde edilen bilginin güvenilirliği ile ilintilidir. Anlamsal Web’in uygulama ortamı Şekil3.2’de gösterilmiştir[131,132].



Şekil 3.2. Anlamsal Web uygulama görünümü

Görüleceği üzere anlamsal web ontoloji üretimi ile başlar. Bu ontoloji ile alan bilgilerinin paylaşılabilirliği ve yorumlanabilirliği sağlanır. Ontoloji üretiminde ontoloji üretim araçları kullanılır. Daha sonra alanda yer alan belgelerin, ontoloji ile ilişkilendirilebilecek üst verilerinin üretimi için etiketlendirme araçları devreye girer. Üst veriler, ontoloji ile ilişkilendirilerek genişletilir ve üst veri havuzunda saklanır. Üst verilerin birlikte işlenmesi, hepsine birlikte yeniden bakarak çıkarsama yapmak ve ontolojik eşleştirme işlemlerini gerektirebilir.

Sonuç olarak Anlamsal Web için bugün en önemli araştırma ve geliştirme çabaları üst veri ve ontoloji teknolojileri üzerinden yapıldığı, 2002 ve sonrasında gerçekleştirilen araştırmaların sayısına ve niteliğine bakılarak söylenebilir[131,132,148,149]. Bu noktada geliştirilen teze ilgisini ve açıklamaları somutlaştırmak için basit bir problem ile üst veri oluşturma aşaması örneklenecektir. Ontoloji ve ontoloji tabanlı erişim konu başlıkları ayrıntıları ile bölüm 3.2’de, Ontoloji gösterim dilleri bölüm 3.3’te yer almaktadır. Örnek probleme Extensible Markup Language (XML) dilinin neden tek başına üst veri oluşturmak için yeterli olmadığı tartışması ile başlamanın uygun olduğu düşünülmüştür.

XML, işaretçi (markup) tanımlamak için kullanılan üst dildir. XML, birçok ayrıştırıcıya, uygulamalar arası veri ve üst veri aktarımını destekleyen tanımlanmış araçlara sahiptir. Ancak XML verinin sözdizimi (syntax) ile ilgilenir. Yani anlamsal destek sunmaz. Örneğin iç içe geçmiş etiketler (tags) için anlamsal içerme ya da sahip olma ilişkisini açıkça ifade etmez. Bu, etiketleri yorumlayan uygulamanın sorumluluğuna bırakılır. Örneğin “Ahmet Büyük Kesikli Matematik dersinden sorumludur” ifadesi XML ile bir kaç biçimde ifade edilebilir:

```
<ders adı = “Kesikli Matematik”>  
  <sorumlu> Ahmet Büyük</sorumlu>  
</ders>
```

```
<sorumlu adı = “Ahmet Büyük”>  
  <öğretir> Kesikli Matematik </öğretir>  
</sorumlu>
```

```
<sorumlulukListesi>  
  <sorumlu> Ahmet Büyük </sorumlu>  
  <ders> Kesikli Matematik </ders>  
</sorumlulukListesi>
```

Görüleceği üzere ilk iki gösterimde iç içelik ilişkisi tamamen zıt uygulanmıştır. Ancak ifade edilen bilgi aynıdır. Yani etiketlerin içe içe düzenlenişine ait bir standart yoktur.

RDF (Resource Description Framework) veri modeli tanımlama dilidir[82,85,93,94, 123, 124,133]. Bu özelliği üst veri tanımlamak için de uygun bir dil olmasını sağlar. RDF, WWW tarafından tanımlanmış 1999 yılında bildiri yapılmış ve kullanım yaygınlığı ile dikkati çeken bir dildir. Günümüz tarayıcıları ile de ele alınabilmektedir[150]. RDF ile bilgi, üçlü (triple) bloklar biçiminde ifade edilir. Bloklar <kaynak, özellik, değer> biçimindedir. Her bir üçlüye deyim (statement) adı verilir. Elbette her dil gibi RDF için de sözdizim kuralları vardır. Sözdizim XML tabanlıdır. Ancak bu ilişkiye XML sözdizimini olduğu gibi kapsar, eklentileri mevcuttur yorumu getirilebilir.

RDF alana bağlı modelleme dili değildir. Kullanıcı seçtiği bir alanı dilin sözdizimine uyararak modeller. Bu noktada RDFS (RDF Schema) dilinden destek alınır. RDFS, RDF ile ifade edilen modelin sözlüğünün oluşturulmasında kullanılır. RDFS ile modelde yer alan varlıkların adları, özellikleri ve ilişkileri tanımlanır. Örneğin “Öğretim görevlisi bir akademik personeldir” ifadesinde “öğretim üyesi” ve “akademik personel” varlık adlarını ifade eder. İki varlık arasında kalıtım (is-a) ilişkisi ifade edilir. Yani tüm öğretim üyeleri aynı zamanda akademik personeldir sonucu yorumla elde edilir. Bu yorumun RDFS ile ifade edildiğinde de bilgisayarlarca yapılabilmesi Anlamsal Web’in gereğidir. Ancak aşağıdaki bir dizi XML cümlecığı ile ifade edilen bilgiler akademik personelleri bulmak için sorgulanırsa, sorgu sonucu sadece “Meltem İnan”a erişilir.

```
<akademikPersonel> Meltem İnan </akademikPersonel>  
<profesör> Özlem Şen </profesör>  
<ders adı = “Kesikli Matematik”>  
    <sorumlu> Ahmet Büyük </sorumlu>  
</ders>
```

Oysa “Özlem Şen” ve “Ahmet Büyük” te akademik personeldir. Bunun çıkarsanması için tüm profesörlerin akademik personel olduğu ve derslerin sadece akademik personellerin sorumluluğunda olabileceği bilgileri de verilmelidir. Bu tür bir ifade anlamsal model oluşturma anlamına gelir ve RDFS ile ifade edilebilir.

Özetle RDFS anlamsal bilginin makinelerce erişilebilirliğini sağlayan bir dildir. RDF ve RDFS birlikte anlamlı bir veri modelinin tanımlanmasını ve tanıma uygun modelin ifadesini mümkün kılarlar.

RDF'in deyimlerini oluşturan üç parça kısaca şu biçimde tanımlanabilir:

1. Kaynaklar (Resources): Üzerinde konuşulan her tür varlık bir kaynak olarak ele alınır. Örneğin yazar, kitap, yer, kişi, otel, oda vb. gibi. Her kaynak bir URI'ye (Uniform Resource Identifier) sahiptir. URI, kaynak için biricik olan değerdir. Bir internet adresi olabileceği gibi bir kimlik numarası da olabilir.
2. Özellikler (Properties) : Özellikler özel türde kaynaklardır. Kaynakların ilişkisini tanımlayan kaynaklardır. Örneğin yazan, sahibi, sorumlusu, yaşı vb. gibi. Kaynaklar URI'leri ile tanımlanır ve URI'ler arası ilişkilerde özellikler ile ifade edilir.
3. Değerler (Values): Kaynakların özelliklerinin aldığı değerlerdir. Basit veri türünde olabilecekleri gibi başka URI'lerde değer olarak kullanılabilir.

Her üç bileşen birlikte kullanıldığında RDF deyimini elde edilir. Örneğin "Ahmet Büyük Kesikli Matematik dersinin sorumlusudur" ifadesinde "Ahmet Büyük" kaynağı, "sorumlusu" özelliği ve "Kesikli Matematik" de değeri ifade eder. Şu durumda, değer bir başka kaynak olmuştur. Bu ifadenin çizge gösterimi Şekil 3.3'de sunulmuştur.



Şekil 3.3. RDF Deyiminin çizge görünümü

Aslında tüm RDF ifadeleri Şekil 3.3'de görülen biçimde yönlü bir çizgeye çevrilebilir. Kenarlar özelliği ifade eder ve kaynak düğümden başlayıp değer düğümünde sonlanır. Şekil 3.3'deki çizgenin oluşmasına sebep olabilecek, bir başka deyişle çizge ile aynı anlama gelecek olan RDF ifadesi aşağıda verilmiştir.

```
<?xml version="1.0" encoding="UTF-16" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#
```



```
xmlns:üniversiteModel= file:///d:/UM/universite-rdf-ns# >  
<rdf:Description rdf:about = "Ahmet Büyük">  
  <üniversiteModel:sorumlusu>Kesikli Matematik<üniversiteModel:sorumlusu>  
</rdf:Description>  
</rdf:RDF>
```

RDF gösterimde birinci satır belgenin sözdizim sürümünü ve kodlama standartını ifade eder. İkinci satır RDF gösterimin başladığını, üçüncü satır kullanılan RDF etiketlerinin tanımını, dördüncü satır üniversite alanı için RDFS ile yapılan modelin tanımının nerede olduğunu gösterir. Beşinci satır "Ahmet Büyük" adında kaynağın tanımlandığını bildirir. Altıncı satır kaynağa ait üniversite modelinde tanımlanan "sorumlusu" özelliğini "Kesikli Matematik" değeri ile bildirir. İlerleyen satırlar etiket kapatma satırlarıdır. Bu biçimi ile RDF ile XML arasındaki fark hissedilememiş olabilir. Ancak asıl anlamsallığı içeren kesim RDFS ile oluşturulan modeldir. Bu model ile sorumlusu özelliğinin değerinin sadece akademik personeller olabileceği ifade edilebilir. Böylece "Ahmet Büyük" kaynağı tanımında açıkça yazılmasa bile kendisinin de akademik personel olduğu model ile çıkarılabilir. Anlamsal Web'in diğer katmanını oluşturan ontolojiler de paylaşılabilme gerekliliğini taşır. Dolayısı ile ontolojiler içinde birçok standart gösterim dili tanımlamıştır[131,132,148,149]. Bu dillerin hepsi katmanlı kek görünümünden de anlaşılacağı üzere RDF ve RDFS tabanlıdır. Dolayısı ile RDFS'in ayrıntılı tanıtımına bölüm 3.3.1 'de yer verilmiş, bu kesimde sadece üst veri modeli tanımlayabilme ve üst veri ifade edebilme özelliklerine değinilmiştir.

Günümüzde varolan Web'e üst veri ve ontoloji eklentileri Anlamsal Web'e giden yoldaki ilk kilometre taşlarıdır. İlerleyen kesimde ontoloji, özellikleri, kullanımı ve örnek çalışmaları yer almaktadır.

## 3.2. Ontoloji (Ontology)

Tez kapsamında geliştirilen sistem, bir web sayfasının içerik özelliklerini kullanarak anlamsal özelliklerini çıkarsamayı, dizinlemeyi ve erişime açmayı hedeflemiştir. Yani sayfanın üst verisinde içerik özelliklerinin yanı sıra hatta daha öncelikli olmak üzere anlamsal özelliklerinin de yer alması söz konusudur. Anlamsal özellikler seçilen bir alana bağlı olarak çıkarılmıştır. Çünkü içerik özelliği olarak metinlerden gelen terimlerin ve görüntülerden gelen nesne adlarının kendi içlerinde birbirleri ile ilişkilendirilmesinin ancak bir alan bağlamında mümkün

olacağı düşünölmüştür. Geliştirilen sistem, anlamsal özelliklerin çıkarımına uygun olduđu düşünölen bir modelleme biçimi önermektedir. Bir alanın modellenmesi doğrudan ontolojiyi çağırıştırır.

Ontoloji kavramının bahsi ilk olarak yapay us alanında geçmiştir [74,75,76]. Ancak kısa sürede birçok alanda kullanılır hale geldiđi yapılan literatür taramasında gözlemlenmiştir. Bu bölümde ontoloji tanımı yapıldıktan sonra, keskin ve bulanık ontoloji ayrımı tartışılmış ve her iki tür ontoloji ile ilgili çalışma örneklerine yer verilmiştir. Ontoloji gösterimi kendi başına bir başlık olarak bölüm 3.3’de yer almaktadır. Bölüm 3.3’de paylaşımaya açık, standartlaşma yolunda birbiriyle yarışan RDF ve RDFS tabanlı dillere yer verilmiştir. Oysa ontoloji gösteriminde biçimsel diller de kullanılmıştır. Bölüm 3.4’te önemleri ve örnek teşkil etmeleri sebebiyle ACL (Agent Communication Language) [98] ve F-Logic[83, 84] gibi iki biçimsel dil ile ontoloji tanımlayıp kullanan iki ayrı sistem (FERMI ve OntoBroker) tanıtılmıştır. Bölüm 3.5’de, ontoloji gösteriminde günümüz teknolojilerinden UML (Unified Modelling Language) incelenmiş ve en son olarak bölüm 3.6’da tam bir ontoloji gösterim dili olmamakla birlikte, web üzerinde veri tabanı işleten sitelerin veri tabanı tasarımında kullandıkları ve varlık bağıntı çizeneklerinin web üzerinde kullanılan biçimi olan ERW’in (Entity and Relations on the Web) tanıtımı yapılmıştır.

### 3.2.1. Ontoloji Tanımı

Ontolojiler verilere ait üst verilerin ifade ettiđi anlamın çıkarsanmasında kullanılan anahtar teknolojilerdir. Bir başka deyişle veriye ait içerik özellikleri ile anlamsal özelliklerin arasında kalan boşluğu dolduran bağlam bilgisidir. Ontoloji üzerine birçok tanım yapılmıştır:

“Kavramsal modelin, açık ve net tanımıdır [72]”

“İlgilenilen alanın, ortak algılanışıdır.[71]”

“Gerçek dünyada, bir alanın kavramsal modelinin biçimsel tanımıdır.[69,70,72]”

“Uygulama alanının soyut görünümünü oluşturan kavramların ve ilişkilerin bütünüdür .[125,26]”

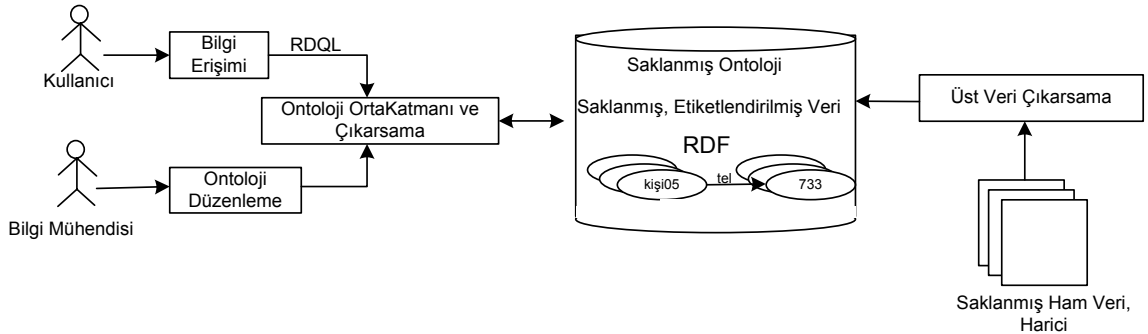
Aslında tüm farklı ontoloji tanımlarında ortak olan bileşenler şunlardır:

- Bir alan (Domain)
- Alanın kavramsal modeli (Conceptual Model)

- Modelin ifade edileceği dil (Representation Language)

Günümüzde ontoloji kurmak ve kullanmak bilgisayar bilimlerinin farklı dallarınca kullanılan bir yöntemdir: Bilgi Erişimi, Sayısal Kütüphaneler, Bilgi Mühendisliği, Doğal Dil İşleme, Bilgi Entegrasyonu, E-Ticaret, Anlamsal Web, Çoklu Ajan Sistemleri vb gibi. Rapor kapsamında; ontolojiler, Bilgi Erişimi ve Anlamsal Web açısından incelenecektir.

Anlamsal Web için ontoloji, web sayfalarının üst verilerinin anlamını çıkarsamakta kullanılır. Yani erişim sürecinde, RDF ile ifade edilmiş sayfa üst verileri ve bir anlamsal dil ile gösterilmiş ontoloji mevcuttur. Ontoloji, modellediği alandaki değişiklikleri yansıtmalıdır. Bu sebeple ontoloji düzenleme olarak adlandırılabilen ve araç kullanılarak ya da kullanılmaksızın gerçekleştirilen bir işlemin var olması da kaçınılmazdır. Ontoloji düzenleme görsel bir arayüz ile yapılıyorsa, düzenlemeleri varolan ontolojiye yansıtmak ve erişimde sorgu ile ontolojinin ilişkilendirilmesini sağlamak gibi görevler ontoloji orta katmanı ile yerine getirilir. Anlamsal Web’de ontoloji kullanımının ana hatlarını tarif eden bu açıklamalar Şekil 3.4’de çizim ile ifade edilmiştir.



Şekil 3.4. Anlamsal Web bilgi yönetim mimarisi

Şekil 3.4’de kullanıcı, bilgi erişimini bir arayüzle tetikler ancak sorgu RDQL ile ontoloji orta katmanına iner. Çünkü bu çizimde Ontolojinin gösteriminin RDF tabanlı bir dille ve üst verinin de RDF ile gösterildiği varsayılmıştır.

Bugün kullanılan ontolojilerden bir kaçı şunlardır: WordNet<sup>3</sup>, Cyc<sup>4</sup>, Sensus[4], Unified medical Language System (UMLS)<sup>5</sup>, KA2 Ontology<sup>6</sup>, Web-KB Ontology<sup>7</sup>.

Ontoloji gösteriminde birçok biçimsel dil kullanılmıştır. Bunlar hızlıca sayıldığında ACL, LOOM, CyCL, F-Logic, Conceptual Graphs, Ontolingua dikkat çekmektedir. Bu tarz biçimsel dillerin kullanımını örneklemesi için F-Logic'in OntoBroker içindeki kullanımı ve ACL'in FERMI içinde kullanımı bölüm 3.4'te incelemiştir. Sıralanan bu biçimsel diller ile bir ontolojinin gerektirdiği her tür bağıntı ve tanımları ifade etmek mümkündür. Ancak öncelikle uzun süre yapay us alanında kullanıldıkları için yaygınlık noktasında zaaf taşımaktadırlar. Ayrıca okunması ve kullanılması sonradan gelişen dillere göre daha zordur[73,74,75,76,77,78,79,80].

Biçimsel dillerin kullanım sıklıklarının azalmasındaki en büyük faktör, internetin "ikinci nesil"[85] olarak adlandırılan anlamsal web dönemine girmiş olmasıdır. Zira artık bir alan için ontoloji gösterimini gerçekleştirmenin yanında, ontolojinin dağıtımı, paylaşımı ve değişimi gibi ihtiyaçlar da mevcuttur. Bu ihtiyaçlardan hareketle bilgi erişimi açısından önemli olduğu düşünülen ontoloji gösterim dilleri şunlardır: Resource Description Framework Sheets (RDFS)<sup>8</sup>, Unified Modelling Language (UML)<sup>9</sup>, Darpa Agent Markup Language(DAML)<sup>10</sup>, Ontology Interchange and Inference (OIL)<sup>11</sup>[93,94,95]. Bu diller ilerleyen bölümlerde tek tek ele alınacaktır.

Ontoloji gösterimini herhangi bir dille gerçekleştirmek için dilin şu bileşenleri tanımlama yeteneği olmalıdır[75,76,77]:

- Varlıklar
- Varlıklara ait nitelikler
- Varlıklar arasında kavramsal düzeyde varolabilecek tüm yapısal (genelleme, içerme) ve yapısal olmayan bağıntılar

---

<sup>3</sup> [www.cogsci.princeton.edu/~wn](http://www.cogsci.princeton.edu/~wn)

<sup>4</sup> [www.e-cyc.com](http://www.e-cyc.com)

<sup>5</sup> <http://umlsks.nlm.nih.gov>

<sup>6</sup> [www.aifb.uni-karlsruhe.de/WBS/broker/ka-onto.onto](http://www.aifb.uni-karlsruhe.de/WBS/broker/ka-onto.onto)

<sup>7</sup> [www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb](http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb)

<sup>8</sup> <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

<sup>9</sup> <http://doc.ic.tsu.ru/xml/dox/sections/analysis+design/uml/dox.xml>

<sup>10</sup> <http://www.daml.org/2001/03/reference.html>

<sup>11</sup> *IEEE Intelligent Agent Systems, 16(2):38-44,2001*

- Kurallar, kısıtlar
- Kural ve kısıtların kalıtımı
- Bağıntıların kalıtımı

Yukarıda adı geçen tüm diller UML hariç bu özellikleri desteklemektedir. UML'in yetenekleri, kurallar noktasında yetersizdir. Bu sebeple bir başka dilden destek almak zorundadır. UML için varolan bu açık nokta bölüm 3.5'de ayrıntılı ele alınmıştır.

### 3.2.2. Ontoloji Türleri

Ontoloji oluşturmak için gerekli ana adımlar sınıfların tanımlanması, sıradüzensel ilişkilerin çıkarımı, diğer ilişkilerin değer aralığı ve çokluları ile belirtimi gibi özetlenebilir. Tipik bir ontolojiden beklenen temel kriterler şu biçimde listelenebilir [128]

- Netlik ve Nesnellik: Ontoloji alan bilgisini kesin ve nesnel tanımları ile doğal dildeki biçimleri ile ele almalıdır.
- Tamlik: Tanımlar gerekli ve yeterli koşul ve kısıtlar ile yapılmalıdır.
- Tutarlılık: Ontoloji güvenilir çıkarımların yapılabileceği tutarlılıkta olmalıdır.
- Tek düze genişletilebilirlik: Ontoloji alan bilgisindeki değişiklikleri yansıtacak biçimde güncellenebilmelidir. Ancak güncelleme yöntemi tek biçimde ve iyi tanımlanmış olmalıdır. Yani varolanı koruyarak eklenti yapabilmeye olanak vermelidir.

Ontolojiler modelleme düzeylerine, dillerine ve odaklandıkları bakış açısına göre türlere ayrılırlar. Ancak temel olarak üç türe bölünebilirler[127, 128]

- Üst Düzey Ontolojiler: Herhangi bir bağlama göre tanımı değişmeyen zaman, nesne, eylem, uzay gibi kavramların modellenmesidir. Bu türde genel bir ontoloji tanımı henüz tatminkar düzeyde yapılmamıştır.
- Alan ve Görev Ontolojileri: Biyoloji ya da tıp gibi üreysel alanlarda sözlük tanımı için oluşturulan ve alandaki tanımları bu sözlük ile gerçekleştiren ontolojiler.

- Uygulama Ontolojileri: Özel bir alanın kavramlarını ve olabilecek her tür ilişkisini ele alan ontolojilerdir. Kavramlar ontolojide kendilerine tanımlanan role uyumlu ele alınmak ve kullanılmak zorundadır.

Ontolojiler, oluşturma biçimlerine göre de üçe ayrılırlar:

- Elle Oluşturulan Ontolojiler: Bu tür ontolojilerde alan uzmanları alanda varolan kavramları, ilişkileri ve kısıtları tümüyle tanımlar. Bu tanımların bir gösterim diline çevrimi genelde otomatik yapılır. Ontolojideki tüm güncellemeler de uzmanlarca gerçekleştirilir. Doğru modelleme adına oldukça iyi sonuç üretilir. Çünkü ontoloji mühendiliği tamamen uzmanların sorumluluğuna bırakılmıştır.
- Yarı otomatik Oluşturulan Ontolojiler: Ontolojinin içerdiği ilk tanımlar alan uzmanları tarafından yapılır. Ontolojinin kullanıldığı uygulama içindeki geri bildirimler ile ontoloji tanımları ve ilişkileri genişletilir. Bu tür sistemlerin bazılarında otomatik genişleme ile yapılan eklentiler dönem dönem alan uzmanlarınca ele alınarak düzenlenir.
- Otomatik Oluşturulan Ontolojiler: Ontolojinin bir kapalı derlemeden otomatik üretilmesidir. Yani uzman müdahalesi ya da bilgisi ontoloji üretim aşamasında yoktur. Bu tür çalışmaların tipik girdisi bir alana ait sınıflandırılmış derlemdir. Örneğin tıbbi bir uygulama için “migren”, “beyin tümörü” gibi sınıflar altında bu başlıklar ile ilişkili belgeler yer alır. İstatistiksel yaklaşımlar, doğal dil işleme mantığı ya da sinir ağı ile öğrenme gibi yaklaşımları ile terimler ve ilişkiler belirlenir. Ancak burada çıkarsanacak ilişki türleri ontoloji üretiminin başında tespit edilir. Örneğin hedeflenen içerme ilişkisi ise istatistiksel yaklaşımda iki terim arasındaki içerme ilişkisinin tespiti, birlikte ve ayrı ayrı geçtikleri belgelerdeki sıklıklarına göre üretilir. Eğer içerme ilişkisi, doğal dil işleme ile tespit edilecekse cümle yapılarına bakılır. Örneğin İngilizce’deki “made up of” , “contain”, “consist of” terimlerinin geçtiği cümleler ayrıştırılarak içeren ve içerilen ilişkisinin tespitinde kullanılabilir. Bu tür ontolojilerin, ele aldığı sınırlı sayıda ilişki için belli doğruluk dereceleri üretebildiği ve kullanıldığını

gösteren çalışmalar literatür taramasında tespit edilmiştir ve bölüm 3.2.3'de örneklenmiştir.

Ontolojiler içerdikler varlıkların ve ilişkilerin keskinliğine göre iki sınıfa ayrılabilir:

- Keskin (Crisp) Ontolojiler: Eğer ontolojinin modellediği alanda yer alan kavramlar ve ilişkiler kesin tanımlanabiliyorsa, özellikle ilişkiler için var ya da yok ikilemi ile karar verilebiliyorsa elde edilen ontoloji yapısı keskindir. Ontolojinin keskinliği uygulama alanı ve üretim biçimi ile belirlenir. Keskin bir alanın elle geliştirilen modeli de keskin olacaktır. Örneğin sayısal müze sistemlerinde, alan keskindir. Çünkü varlıkların somut sanat eserleri ve sanatçılar olduğu söylenebilir. İlişkilerin de keskin değer aldığı düşünülmektedir. Örneğin her eserin bir yaratıcısı vardır ve bu da sanatçıdır. Burada yaratıcısı olma ilişkisi 0 ya da 1 değeri ile var ya da yok olarak kodlanabilir. Özetle gerçek hayattaki algılanışları var ya da yok biçiminde olan kavramların ve ilişkilerin içerildiği ontoloji keskin olur.
- Bulanık (Fuzzy) Ontolojiler: Modellenecek alandaki kavramların ve ilişkilerin var ya da yok tercihlerine tam oturmadığı, yani alanın bulanık olduğu durumlarda ortaya çıkan modelde bulanık olur. Bulanık ontoloji, keskin ontolojinin, bulanık ilişkiler ile genişletilmiş biçimidir. Bulanık ilişki taşıyan kavramlar da bulanık kavramlar olur[147]. Bulanık ilişki 0 ya da 1 olarak yorumlanması güvenilirlik sorunu yaratan, alternatif olarak [0..1] kapalı aralığında değerlendirilmesi gereken ilişkilerdir. Örnek olarak bulanık alan tıp seçilirse “baş ağrısı”, “migren” ve “grip” kavramları bulanık kavramlar olacaktır. Çünkü açıkça “başağrısı” her iki kavramın ya da hastalığın belirtisidir ama tek başına yeterli değildir. Ayrıca her iki kavramın belirtisi olma derecesi doktorlar için de bulanıktır. Yani bir doktor eğer “başağrısı” varsa sonuç “migren”dir ya da “grip”tir diyemez. Bulanık ontoloji üretiminin bir diğer sebebi otomatik ontoloji üretimi olduğu söylenebilir. Çünkü bu tür ontolojilerde başlangıçta belirtilen ilişkiler, çeşitli yöntemler ile otomatik çıkarılır. Burada kullanılan yöntemler ilişki için bir derece üretir. Yani ilişkinin doğası keskin olabilir ama otomatik ilişki çıkarmak belli derece ile gerçekleştirildiğinde ilişki bulanıklaşır. Örneğin yine “migren”, “grip” ve “baş ağrısı” kavramları kullanılırsa, istatistiksel yaklaşımla “migren” ve “baş

ağrısı” arasındaki ilişki belgelerde birlikte kullanım sıklıklarından doğrudan etkilenecektir. Burada kullanılan sıklığa bağlı formüllerde dereceli sonuç üretecektir. Ancak eşik değeri uygulanırsa ve eşik değerinin üstü 1’e altı 0’a çevrilirse ilişki keskinleştirilir. Ancak yapılan literatür taramasında keskin iken bulanıklaştırılan ilişkilere rastlanılmış fakat bulanık iken keskinleştirilen ilişkilere rastlanılmamıştır.

Keskin ve bulanık ontolojileri bilgi erişiminde kullanan çalışmalar bölüm 3.2.3’te örneklenmiştir.

Anlamsal Web’in bir anlamda anlamsal çekirdeğini oluşturan ontolojiler üzerine birçok araç son yıllarda hızla hizmete girmiştir. Bu araçlar yedi kategoride toplanabilir[129, 130]

- Ontoloji geliştirme araçları
- Ontoloji birleştirme ve bütünleştirme araçları
- Ontoloji değerlendirme araçları
- Ontoloji tabanlı etiketlendirme araçları
- Ontoloji sorgulama dilleri ve araçları
- Ontoloji kütüphane sistemleri
- Ontoloji öğrenme araçları

Özellikle ontoloji geliştirme araçları sayıca diğer altı kategorideki araçlara göre daha fazladır. Bu araçlar ile alan bilgisini görsel arayüzler ile modelleme, modelin farklı diller ile ifade edilmesi, bir anlamsal dil ile ifade edilen ontolojinin görsel modele çevrilmesi, ontolojilerin farklı dillere çevrilmesi, ontoloji kütüphaneleri içinde saklanması gibi işlevleri içerir. Bu amaçla geliştirilen çalışmaların bir kısmı ad ve adresleri ile şu biçimde listelenebilir[130].

- Apollo (<http://apollo.open.ac.uk/>)
- LinkFactory (<http://www.landc.be/>)
- OntoEdit ([http://www.ontoprise.de/products/ontoedit\\_en](http://www.ontoprise.de/products/ontoedit_en))
- OILed (<http://oiled.man.ac.uk/>)
- Ontolingua (<http://www-ksl.stanford.edu/>)
- Ontosaurus (<http://www.isi.edu/isd/ontosaurus.html>)



- OpenKnoME (<http://www.topthing.com>)
- Protege 2000 (<http://protege.stanford.edu/>)
- SymOntoX (<http://www.symontox.org>)
- WebODE (<http://webode.dia.fi.upm.es/>)
- WebOnto (<http://kmi.open.ac.uk/prpjects/wenonto>)

Ontoloji kütüphane sistemleri ile ontolojilerin kayıt edilmesi, güncellenmesi, erişilmesi ve sorgulanması işlemleri yerine getirilir. Bu amaca hizmet eden araçlardan ve kütüphane yazılımlarından bazıları şu biçimde listelenebilir:

- SESAME (<http://www.openrdf.org>)
- RDFDB (<http://guha.com/rdfdb>)
- RDFSTORE (<http://rdfstore.sourceforge.net>)
- JENA (<http://hpl.hpcom/semweb/jena-top.html>)
- KAON (<http://kaon.semanticweb.org>)

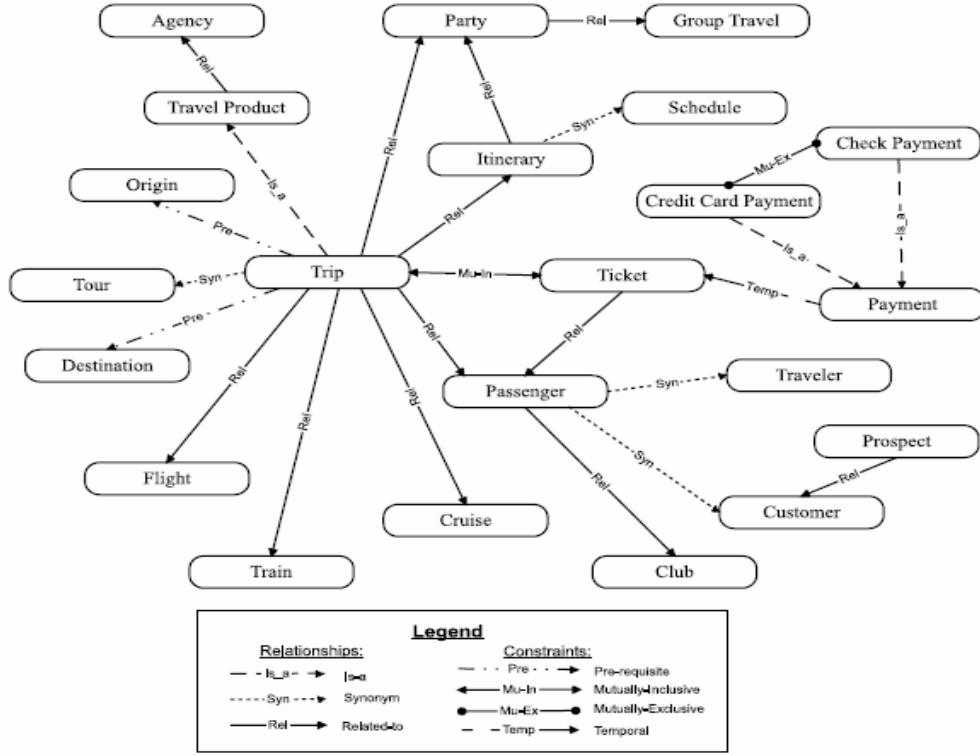
Geliştirilen tez kapsamında önerilen ontoloji modeli üreyseldir. Yani önerilen modelleme biçimi ile seçilen bir alan modellenebilir. Bu da aynı anda farklı alanlar için farklı ontolojilerin varolması anlamına gelir. Bu ontolojilerin yönetimi ve erişimi için Jena 1.2[151] kütüphane yazılımı kullanılmıştır. Kullanım biçimi ve yetenekleri geliştirilen sistemin tanıtıldığı bölüm 4'te yapılmıştır.

### **3.2.3 Ontoloji İle İlgili Çalışmalar**

Bu bölümde elle ya da otomatik, bulanık ya da keskin türde oluşturulan ve bilgi erişiminde ya da bilgi mühendisliğinde kullanılan ontoloji üretimi ve kullanımına ait varolan çalışmalardan örnekler sunulmaktadır. Bu örneklerin seçiminde çalışmaların belli bir yaklaşımı örneklemesi ve alanının önemli çalışmalarından olduğu düşüncesi belirleyici olmuştur.

Kavramsal Modelleme için Ontolojilerin Yaratımı, Kullanımı ve Yönetimi (Ontologies for conceptual modelling: creation, use and management) adlı çalışma 2002 yılında Oakland ve Georgia State Üniversite'lerinin ortak çalışması olarak yayınlanmıştır. Çalışmada, "seyahat" anlamsal başlığı için alan modellemesi sezgisel bir yolla gerçekleştirilmiştir[137]. Bu çalışmada verilen ontoloji kesiti

keskin bir ontolojinin güzel bir örneğidir. Ontoloji baştan üç tür ilişki için elle oluşturulmaktadır. Bu ilişkiler kalıtım, eşanlamlılık ve ilgililik ilişkileridir. İlgililik ilişkisi, birlikte ele alınması gerektiğini belirten kalıtım ve eşanlamlılık haricindeki tüm ilişkilerdir[137]. Bu ontolojinin bir kesiti şekil 3.5'te yer almaktadır[137].



Şekil 3.5: Örnek keskin ontoloji: Seyahat

Şekil 3.5'te görüleceği üzere kalıtım, eşanlamlılık ve ilgililik ilişkileri kutular içinde yer alan varlıkları birbirine bağlayan kenarlar üzerine yazılmıştır. Bu ontolojinin önemi çalışmanın amacında saklıdır. Şöyleki, yaratılan bu ontoloji "Seyahat" alanı ile ilgili veri tabanı tasarımlarının (varlık bağıntı çizeneklerinin) doğrulanması için kullanılmaktadır[137]. Unutulmuş varlık ya da bağıntılar için "seyahat" ontolojisi tasarımcıya rehberlik eder. Alan uzmanı, ontolojiyi şekil 3.5'te belirtilen ilişkileri görsel arayüze sahip bir araç kullanarak oluşturur. Çalışmanın sezgisel özelliği alan modelleyen uzmana önermekte olduğu 7 adımlık süreçtir. Ancak model üretimi adına otomatik bir özellik içermemektedir.

ABC Ontoloji ve Modeli (The ABC Ontology and Model) adlı çalışma 2002 yılında Cornel Univerisitesi ve sanayi işbirliği ile gerçekleştirilmiştir. Ontolojinin kullanım alanı müzeler için sayısal kütüphane uygulamalarıdır[139]. CIMI müzeleri

tarafından sağlanan eser, sanatçı üst verilerinden hareketle elle oluşturulmuştur. Ontoloji, sayısal kütüphanede yer alan eserlerin sorgulanmasında doğrudan kullanılır. Üst veriler RDF ile ifade edilmiştir. Ontolojide parça-bütün, zaman, olay, altolay, yaratan, öncülü, ardılı, yeri, zamanı, kopyası gibi ilişkiler yer almaktadır. Dolayısı ile ontoloji tam olarak bir uygulama ontolojisidir ve alanının özelliklerini taşır[139]. [137]'de yer alan çalışmaya bakıldığında üst düzey ontoloji ile uygulama ontolojisi farkı açıkça görülmektedir. [137]'de ki çalışma da ele alınan ilişkiler her alanda yer alabilecek ilişkiler iken, bu çalışmada doğrudan eser ve sanatçı bağlamında özel ilişkiler tanımlanmıştır. Tanımlanan ABC ontolojisinde otomatik olgular yaratılır. Şöyle ki; sayısal kütüphanede eser ve sanatçılar için tutulan yapısal bilginin iyi tanımlanmış ve eksiksiz oluşu, eser ve sanatçı ayrımının yapılmış olması otomatik işlemeyi mümkün kılmıştır.

Ontoloji tabanlı Bilgi Erişim Modeli (Ontology based Information Retrieval Model) adlı çalışma, 2004 yılında Madrid Üniversitesi'nde gerçekleştirilmiştir[140]. Bu çalışmada kullanılan ontoloji ilişki içermez. Sadece kavram (concept) adı verilen soyut ya da somut anlama gelebilecek konu başlıklarını içerir[140]. Örneğin bu yaklaşımla sanat alanı modellenenirse, resim, sanatçı, müze, eser gibi kavramlar tanımlanabilir. Ancak kavramlar arası ilişki yoktur. Çünkü çalışmanın amacı etiketlenmiş belgeleri kavramlar altında gruplayarak erişime açmaktır[140]. Bu yaklaşım da anahtar tabanlı erişime alternatif olarak sunulmuştur. Ancak çalışmada ilk göze çarpan unsur alternatif olduğu yaklaşımın ağırlıklandırma biçimini kullanmasıdır. Şöyle ki; sisteme dahil olan belge elle etiketlenir. Bu etiketler belgenin üst verisini oluşturur. Bu etiketlerin belgede geçme sıklığı ve kavramın olgusu olan diğer belgelerin etiketlerinde geçme sıklığı kullanılarak kavram ile belge eşleştirilir. Çalışmanın ana amacı belgelerin sınıflanması, sorguların belge sınıfları ile işlenmesidir. Sorgulamada RDQL kullanılır[140].

Anlamsal web için ontoloji tabanlı bilgi erişim modeli (Ontology-based information retrieval model for the semantic web) adlı çalışma Çin'de Savunma Teknolojileri Üniversitesi'nde 2005 yılında gerçekleştirilmiştir. Bu çalışmada bir alan birçok alt alanın birleşimidir yaklaşımı ile bir alanın altalanları için varolan ontolojilerin birleştirilmesi amaçlanmıştır[141]. Alt alan ontolojilerinin tanımlandığı ve belgeler ile ontoloji olgularının ilişkilendirildiği ortamlarda, alt alan ontolojilerinin içerdiği

sınıfların eşitliği için bir çıkarsama gerçekleştirilmiştir. Bu çıkarsamada vektör uzayı yaklaşımı esas alınmıştır. Yani sınıflar ile ilişkilendirilen belgelerin vektörel benzerlikleri sınıfların bezerliklerinin tespitinde kullanılmıştır. Bu çalışmada birleştirilecek alt alan ontolojilerin OWL ile gösterilmesi gerekmektedir. Çalışma bu anlamda ontoloji üretmeyi değil varolan ontolojileri otomatik birleştirmeyi amaçlar. Birleştirme de eşit sınıfların tespiti ve birleştirilmesinden oluşur[141].

Ontoloji tabanlı görüntü erişimi (Ontology based image retrieval) adlı çalışma 2003 yılında Helsinki Üniversitesi'nde gerçekleştirilmiştir[142]. Bu çalışmada bir alan ontoloji uzmanlarca araç kullanılarak yaratılır ve uygulama ontolojisi türündedir. Bu ontolojide kullanılan kavramlar ile görüntülerin etiketlenmesini sağlayan bir araç geliştirilmiştir. Bu araç ile elle etiketlenen görüntüler, doğrudan anlamsal özelliklerini kazanmış olur. Böylece ontoloji üzerinden sorgulamaya açılır[142].

Anlamsal Web üzerinde bilgi erişimi (Information retrieval on the semantic web) adlı çalışma Maryland Üniversitesi'nde 2002 yılında gerçekleştirilmiştir[143]. Çalışmada alan modellemesi için olay tabanlı (event based) bir öneri getirilmiştir. Ontoloji elle üretilir ve DAML+OIL ile ifade edilir. Alan ile ilgili belgeler ontolojideki terimlere göre etiketlenir. Bu işlem otomatiktir[143]. Erişim etiketler üzerinden gerçekleştirilir.

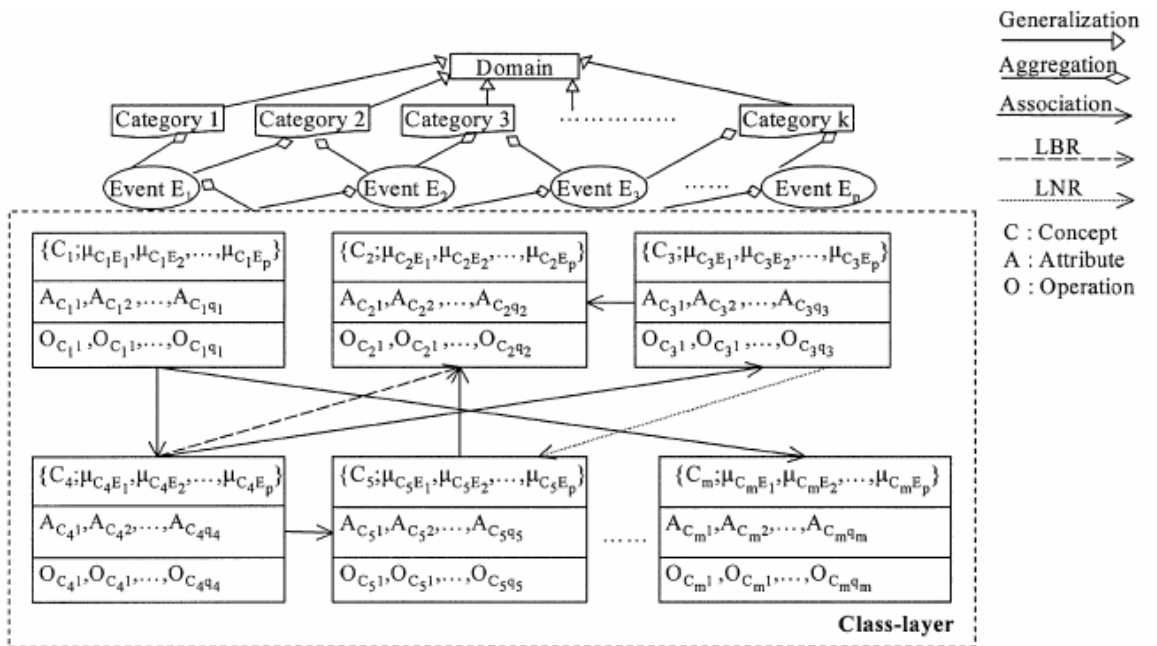
Elbette, bu noktaya kadar ele alınan örneklere benzeyen birçok başka çalışma literatürde bulunmaktadır. Ancak genelinin ana hatları burada verilen örnekler ile uyumludur. Yani keskin ontolojiyi elle üreten, elle modellemeyi bir araç ile desteleyen, ontolojiyi bir anlamsal dil ile gösteren ve belge etiketlemesini bu ontolojiye göre elle ya da otomatik gerçekleştiren sistemler, var olan ontolojileri birleştiren ya da varolan ontolojileri yarı otomatik genişleten çalışmalardır[144]. Bu noktadan sonra bulanık ontolojiler örneklenecektir.

PASS (Personalized Abstract Search Services) adlı çalışma 2001 yılında Texas A&M ve Pennsylvania Üniversite'lerinin ortak çalışması olarak yapılmıştır[145]. Bu çalışma ile bir alan için otomatik ontoloji üretimi amaçlanmıştır. Baştan belirlenen ilişkiler iki terim arasında geniş ve dar anlamlılık ilişkisidir. Farkedileceği üzere aslında sadece dar anlamlılık ilişkisi vardır. Geniş anlamlılık, dar anlamlılığın

tersidir. İstatistiksel bir yaklaşım kullanılmış ve geniş anlamli terimin dar anlamli terimden daha sık kullanılacağı varsayımı ile hareket edilmiştir. Şöyle ki, terimlerin sıklığının hesaplandığı derlemler kapalı derlemdir yani seçilen alana özeldir. Bu kapalı derlemde geniş anlamli terimlerin kullanım sıklığı fazla olacaktır beklentisi ile vektör uzayı modeline uygun biçimde iki terim arasında birlikte kullanılma sıklıklarına bağlı dar anlamlilik ilişkisi derecesi üretilir[145]. Bu derece normalize edilmiştir yani [0-1] kapalı aralığındadır. Geniş anlamlilik ilişkisi, elde edilen bu değer 1'e tümleyenidir. İşleme ilk olarak tüm terimler arası dar anlamlilik ilişkisi hesaplanarak başlanır. Daha sonra zincir taraması yapılır. Yani x terimi ile y terimi arasında dar anlamlilik ilişkisi var iken  $(x, x_1), (x_1, x_2), \dots, (x_n, y)$  çiftleri arasında da dar anlamlilik ilişkileri mevcut ise  $(x, y)$  çifti arasında ilişki kurmak yerine bu zincir kullanılarak anlam sıradüzeni önerilir. Elde edilen bulanık ontoloji belge özetlemede kullanılmıştır[145].

Medikal belge erişimi için bulanık ontoloji (a fuzzy ontology for medical document retrieval) adlı çalışma 2004 yılında Aucland Teknoloji Üniversitesi'nde gerçekleştirilmiştir[146]. Bu çalışma ile MesH'in (Medical Subject Heading) keskin yapısının bulanıklaştırılması amaçlanmıştır. MesH, tıbbi terimlerin anlam genişliği ve ilgililiğine göre sıradüzensel gösterimidir. Amaç bu yapıda yer alan ilgililik ilişkisini bulanıklaştırmaktır. Çünkü mevcut keskin yapının sorgu genişletmede sorun oluşturduğu tespit edilmiştir[146]. Örneğin "başağrısı" terimi MesH içinde toplam 16 farklı başlıkta yer almaktadır. Sorgu genişletmek için 16 başlığın hepsini kullanmanın duyarlık sorunu yarattığı, ancak keskin ilişkiden kaynaklı 16 başlık arasında seçim yapılamadığı belirtilmiştir[146]. İlişkilerin bulanıklaştırılması ile varolan 16 başlık arasından ilgililik derecesine göre seçim yapmak amaçlanmıştır. Bulanıklaştırmak için öncelikle her konu başlığına ait olduğu uzmanlarca tespit edilen belge derlemleri oluşturulmuştur. Örneğin "migren" başlığı altında sadece migrene değinen belgelere yer verilmiştir. Her bir terimin toplam tüm belgelerde geçme sıklığı ve sadece başlığa ait belgelerde geçme sıklığı kullanılarak terim ile başlık arasındaki ilgiyi [0-1] aralığında gösteren değerler üretilmiştir. Böylece keskin ilişkiler bulanıklaştırılıp, sorgu genişletme sorununa alternatif çözüm üretilmiştir[146]. Görüleceği üzere bu çalışmada varolan medikal model aynen korunmuş, sadece modeldeki ilişkiler derecelendirilmiştir.

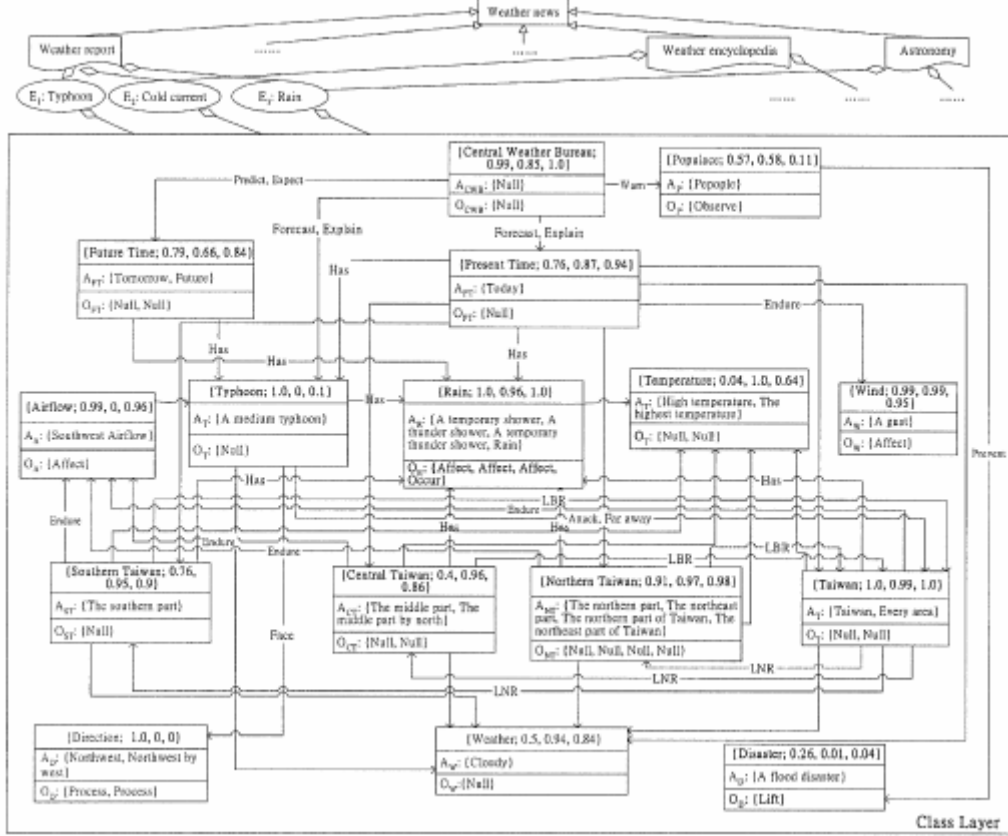
Çince elektronik haberlerin özetlenmesi için ağırlıklandırılmış bulanık ontoloji (weighted fuzzy ontology for chinese e-news summarization) adlı çalışma 2005 yılında Cheng Kung Üniversitesi'nde yapılmıştır. Bu çalışma da mevcut keskin ontolojiyi bulanıklaştırmakla ilgilenmiştir. Hava durumu ile ilgili haberlerin otomatik özetlenebilmesi için haber metinleri içinde terim seçme gereksinimi tanımlanmış ve terimlerin seçimi bulanık ontolojiye göre yapılmıştır[147]. Bu çalışmanın kullandığı bulanık ontoloji modelinin, bulanık ontolojiler için güzel bir örnek olduğu düşünülerek şekil 3.6'da sunulmuştur. Ayrıca şekil 3.7'de modelden meteoroloji alanı için üretilen ontoloji örneğine yer verilmiştir.



Şekil 3.6 : Bulanık ontoloji modeli[147]

Şekil 3.6'da yer alan "Domain" sınıfı üst alan adına, "Category" sınıfı üst alan içinde yer alan alt alanlara, "Event" alt alan içinde yer alan olaylara karşılık gelmektedir. Sınıf katmanı olarak ifade edilen kesimde tüm alt alanlar için ortak olan kavramlar yer almaktadır. Her bir kavram bir sınıf olarak tanımlanmıştır. Her sınıf içinde  $\mu_{C_y E_n}$  biçiminde gösterilen semboller, y. sınıfın n. olay ile ilgililik derecesini göstermektedir. Görüleceği üzere modelde sadece sınıflar ile olaylar arası ilişkiler bulanık ele alınmıştır. Diğer ilişkiler keskin bırakılmıştır. Ancak çalışmada amaç haber özetlemek olduğu için, bu bulanık omurganın yeteceği düşünülmüştür. Örneğin "Event" sınıfının bir olgusu "Rain" ise, sınıf katmanında yer alabilecek bir "Temperature" olgusu ile ilgi derecesi oluşması muhtemeldir.

Dolayısı ile bu olay ile ilgili haberlerin özetlenmesinde “Temprature” olgusunun kullanılıp kullanılmayacağına ilgililik derecesine göre karar verilir[147]. Bu açıklamalar Şekil 3.7 ile örneklenmiştir.



Şekil 3.7: Şekil 3.6'daki model ile meteoroloji ontolojisi gösterimi

Çalışmada gerçekleştirilen bulanıklaştırma işleminde öncelikle keskin ontoloji terimleri ve her bir olaya ait derlem terimleri birlikte kümeleniyor. Daha sonra POS (part of speech), SD (semantic distance) ve TWS (term word similarity) değerlerine göre ontoloji terimleri ve terim kümeleri arasında  $\mu$  değerleri hesaplanıyor. Hesaplanan  $\mu$  değerleri ile önceden tanımlı bulanık kurallar işletilip sonuçlarında yer alan terimler aynı kümede toplanıyor. Aynı sonuçları üreten kuralların sonuç kümeleri Or'lanıyor. Berraklaştırma (defuzzification) gerçekleştiriliyor. Her bir ontoloji terimi ile olay arası ilişkiler, ara basamaklarda üretilen dereceler kullanılarak kuruluyor[147].

Bu kesimde çeşitli türlerde ontoloji esaslı çalışmaların tanıtımlarına ve kaynak gösterimlerine yer verilmiştir. Bu çalışmaların hepsi Anlamsal Web tabanlı

olmamakla birlikte türlerini örneklemeleri için seçilmiştir. Özellikle bilgi erişimi gerçekleştirilen çalışmalarda Anlamsal Web'in varlığı ya da amaçlandığı literatür taramasında gözlemlenmiştir. İlerleyen kesimde Anlamsal Web'de bağlamındaki ontoloji gösterim dillerine yer verilmiştir.

### 3.3. Anlamsal Web Dilleri ile Ontoloji Gösterimi ve Sorgulama

Anlamsal Web Dilleri olarak adlandırılan, XML tabanlı dillerin en büyük özellikleri paylaşılabilirliği desteklemeleridir. Yani anlamsal bilgi sunumunda, standart olmayı hedefledikleri söylenebilir. Bu diller XOL (XML based ontology-exchange language), SHOE (Simple HTML Ontology Extensions), RDF (Resource Description Frameworks), RDFS<sup>12</sup> (Resource Description Framework Schema), DAML<sup>13</sup> (Darpa Agent Markup Language), OIL<sup>14</sup> (Ontoloji Interchange and Inference) [93,94,95], OWL (Web Ontology Language) olarak listelenebilir. Bu dillerden özellikle ontoloji gösteriminde DAML, OIL ve OWL öne çıkmaktadır. Bu üç dil XML, RDF ve RDFS tabanlıdır ve eklentiler içeren bir görünüme sahiptirler. OWL'yi WWW, DAML'ı DARPA ve OIL'de Avrupalı araştırmacılar önermiştir. Her üç dilin birbirine çevrimi mümkündür. DAML ile OIL birlikte DAML+OIL biçiminde de kullanılmaktadır. DAML ile ontoloji gösterimi OIL ile çıkarsama yeteneği desteklenir. Geliştirilen tez kapsamında ontoloji gösterimi OWL ile yapılmıştır. Bu tercihin ana sebebi OWL için hazır kütüphane yazılımlarının kolay erişilebilir ve kullanılabilirliğinin yanı sıra web tarayıcıları tarafında ele alınabilmesidir. OWL'nin Anlamsal Web'in katmanlarındaki (bknz. Şekil 3.1) yeri ontoloji katmanıdır. Dolayısı ile altta kalan XML ve üst veri yani RDF, RDFS katmanlarını kullanır. XML ve RDF'in Anlamsal Web perpektifi ile incelemesi bölüm 3.1'de yapılmıştı. Bu bölümde RDFS ve OWL'nin tanıtımı yapılacaktır.

#### 3.3.1. RDF Schema (RDFS)

RDF ile adlandırılmış tür ve özellikler tanımlanabilir. Ancak tanımlanan tür ve özelliklerin başka tür ve özelliklerin tanımında kullanımı söz konusu değildir. Yani RDF <kaynak, özellik, değer> üçlüsü ile bir URI'yi kaynak olarak gösterip, o kaynak için adlandırarak bir özellik tanımlamayı ve o özellik için basit veri türü ya

<sup>12</sup> <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

<sup>13</sup> <http://www.daml.org/2001/03/reference.html>

<sup>14</sup> *IEEE Intelligent Agent Systems*, 16(2):38-44,2001



da başka bir kaynak türünde değer atamayı sağlar. Ancak kaynak biricik bir ad ile adlandırılan, doğrudan bir nesne görünümündedir. Yani RDF ile “www.cs.hacettepe.edu.tr” kaynağının “yazarı” adlı özelliğinin değeri “Metin Şener”dir biçiminde bir üçlü oluşabilir. Ancak burada kaynağın türünün web sayfası, ilişkinin türünün sahip olma ve değer türünün de sadece bir yazar türü kaynak olabileceği bilgileri yoktur. Bu biçimdeki tür tanımları RDFS ile yapılır. Yani RDFS diğer sınıf ve özelliklerin kullanabileceği sınıf ve özellik tanımlarını yapabilmeyi sağlar. Özetle alan için sözlük tanımlanabilir.

RDFS’in en temel katkıları kaynak, özellik tanımlama, kaynak ve özellik kalıtımı, özellik için değer ve tanım aralığı tanımdır. Şekil 3.8 ile RDF ve RDFS kullanarak motorlu araçlar ile ilgili küçük bir modelleme örneği sunulmuştur[134].

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">

  <rdf:Description rdf:ID="MotorVehicle">
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="PassengerVehicle">
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="Truck">
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="Van">
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="MiniVan">
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Van"/>
    <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
  </rdf:Description>

</rdf:RDF>
```

Şekil 3.8: RDF/RDFS ile modelleme örneği

Şekil 3.8’de görüleceği üzere öncelikle RDFS’in sınıf türü kullanılarak “MotorVehicle” adında bir sınıf tanımlanmıştır. Bu tanımlama ile yeni kaynakların türünü “MotorVehicle” olarak tanımlamak yani tanımdan nesne türetmek mümkündür. Daha sonra “PassengerVehicle” adında bir sınıf tanımlanıp “MotorVehicle” sınıfı ile kalıtım ilişkisi kurulmuştur. “Truck” ve “Van” sınıfı tanımları da benzer biçimde gerçekleştirilmiştir. “Mini Van” sınıfı çoklu kalıtım örneği olarak hem “Motor Vehicle” hem de “PassengerVehicle” sınıfları ile kalıtım ilişkisine sahiptir.

RDF ve RDFS’in çekirdek sınıf ve özellik tanımları kısaca şu biçimdedir[133]:

- rdfs:Resource: Her şey RDF için kaynaktır.
- rdfs:Literal: Özelliğin alacağı değer tamsayı ya da dizgi gibi basit veri türlerinden biridir.
- rdfs:Class: Kaynağın tür tanımınıdır (sınıf tanımınıdır).
- rdf:Property: Türü özellik olan kaynakları ifade eder.
- rdf:Datatype: Türü basit veri türü olan kaynaklardır.
- rdfs:subClassOf: Geçişken bir ilişkidir. Kaynaklar arası kalıtımı ifade eder.
- rdfs:subPropertyOf: Özellik kalıtımını ifade eder.
- rdfs:range: Özelliğin değer aralığını ifade eder.
- rdfs:domain: Özelliğin tanım aralığını ifade eder.
- rdfs:label: Kaynağın mantıksal adıdır.
- rdfs:comment: Açıklama satırıdır.

Özetle, RDF kaynak ve özellik arasında değer alan ikili bir ilişki tanımlarken; RDFS kalıtım, değer ve tanım kümeleri tanımlamayı destekler. Bu yeteneklerine karşın WWW Konsorsiyum ontoloji oluştururken daha fazla gereksinim olduğunu tespit etmiştir. Bu gereksinimler şu biçimde listelenebilir[148]:

- İyi tanımlanmış, okunabilir söz dizimi
- Anlatım gücü yüksekliği
- Etkin çıkarsama desteği
- Doğal dilde kurulan ifadelere yakınlık
- Modelin netliği

Özellikle modelin netliği maddesi etkin çıkarsama maddesi ile doğrudan ilişkilidir. Bir ontoloji oluştururken aşağıda verilen türde çıkarsamalara ihtiyaç duyulabilir:

- Eğer  $x$ , A sınıfının bir olgusu ise ve A sınıfı da D sınıfından kalıtılıyorsa  $x$ 'in aynı zamanda D sınıfının olgusu olduğu çıkarsanmalıdır.
- Eğer A sınıfı, B sınıfı ile eşitlik (equivalence) ilişkisine sahipse ve B sınıfı da C sınıfı ile eşit ise A ile C arasında da eşitlik ilişkisi olduğu çıkarsanmalıdır.
- Eğer B ve C sınıfları ayrık sınıflar iken, A sınıfı  $B \cap C$ 'nin alt sınıfı ise A'dan yaratılan  $x$  olgusu için hata üretilmelidir. Çünkü A sınıfı boş küme sahiptir.

Bu tür çıkarsamaların varlığı ontoloji üretirken tutarlılığın denetimini, maksadı aşan ilişki tanımlarının engellenmesini ve olgularının otomatik sınıflandırılmasını sağlar.

RDFS'in yeteneklerinin sınırlı kaldığı bir kaç nokta şu biçimde listelenebilir:

- Sınıfların birbirinden ayrık tanımlanabilmesi. Örneğin "kadın" ve "erkek" sınıfları "insan" sınıfını kalıtırlar ancak birbiri ile tamamen ayrıktırlar. Yani aynı olgu hem kadın hem de erkek sınıfının olgusu olamaz.
- Sınıflar arası kesişim, birleşim, tümleyen ilişkileri tanımlanamaz.
- Özelliklerin alacağı değerlerin adedi yani kaç ayrık değer alabileceği tanımlanamaz.
- Özellikler için geçişken, biricik ve tersi gibi özel niteleyicileri desteklemez.

Yukarıda listelenen ve örneklenen gereksinimleri karşılamak için WWW, 2001 yılında OWL'nin tanıtımını yapmıştır[131,132,148].

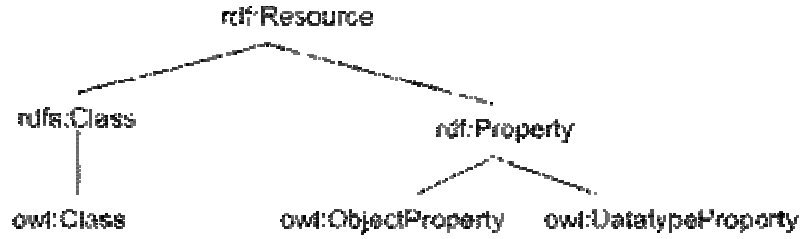
### 3.3.2. Web Ontology Language (OWL)

OWL, RDFS yeteneklerini olduğu gibi kapsayan, sınıfları ve özellikleri RDF'in tanımladığı gibi kullanan, ifade etme zenginliğini artırmak için eklentiler içeren anlamsal web dilidir. OWL üç altdilden oluşur. Bu dillerin kapsamı genişleyecek şekilde sıralaması şu biçimdedir:

- OWL Lite: Sınıf, özellik tanımlama, olgu yaratma, kalıtım ve kısıt tanımlama desteği verir.
- OWL DL: Çıkarsama yapmak için gerekli kararlılık ve tutarlılık özelliklerini kaybetmeden OWL Lite yeteneklerini genişletir.

- OWL Full: OWL DL yeteneklerini en geniş noktaya ulaştırırken kararlılık ve tutarlılık kaygısı götürmez.

OWL ile RDF/RDFS ilişkisi Şekil 3.9 ile örneklendirilmiştir [148].



Şekil 3.9: OWL ile RDF/RDFS arası kalıtım sıradüzeni

Şekil 3.9'da görüleceği üzere diller birbirlerinin özelliklerini kalıtıp zenginleştirmektedir. Bu sıradüzen sebebi ile OWL ile ifade edilen her ontolojinin RDF/RDFS ile ifade edilmesi hatta bu dönüşümün otomatik yapılması mümkündür[151]. İlerleyen kesimde OWL ile ontoloji gösteriminde ihtiyaç duyulan en temel yapılara ait örnekler ve açıklamalar yapılmıştır. Daha fazla bilgi için [123,124,129,130,133,134,135,149] nolu kaynaklar kullanılabilir.

Her OWL belgesi <rdf:RDF> işaretçisi ile başlayıp </rdf:RDF> ile sonlanır. Öncelikle belgede kullanılan işaretçi dillerin isim uzayları(namespaces) başlık (header) kısmında tanımlanır. Örnek başlık şu biçimdedir:

```

<!DOCTYPE owl [
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="&owl;"
xmlns:xsd="&xsd;" >
  
```

Görüleceği üzere her OWL, RDF/RDFS'i ve RDF'te XML kullandığı için her üç dilde başlıkta yer alır. Bundan sonraki kesim ontolojiye ait bilgiler verir. Ontolojinin adı, kullandığı diğer ontolojiler, sürümü gibi bilgilere yer verilir. Aşağıda bu kesime ait bir örnek verilmiştir[135]

```
<owl:Ontology rdf:about="file://d:/sarap/">
<owl:priorVersion rdf:resource="http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine"/>
</owl:Ontology>
```

OWL ile basit sınıf tanımı <owl:Class> işaretleyicisidir. Örnek tanımla şu biçimdedir[135]

```
<owl:Class rdf:ID="Şarap"/>
<owl:Class rdf:ID="Bölge"/>
<owl:Class rdf:ID="İçecek"/>
```

OWL, RDFS'ten kalıtıldığı <rdfs:subClassOf> işaretçisi ile kalıtımı modeller[135]

```
<owl:Class rdf:ID="İçki">
<rdfs:subClassOf rdf:resource="#İçecek" />
</owl:Class>
```

Ontoloji olgusu yaratmak için olgunun tür adı kullanılır. “BoğazKereTurasan” olgunun adıdır ve türü “ÜzümŞarabı”dır[135]

```
<owl:Class rdf:ID="Üzüm"/>
<owl:Class rdf:ID="ÜzümŞarabı">
<rdfs:subClassOf rdf:resource="#Üzüm" />
</owl:Class>
<ÜzümŞarabı rdf:ID="BoğazKereTurasan" />
```

OWL ile iki tür özellik yaratılabilir. Bunlardan ilk <owl:ObjectProperty> işaretçisi ile ifade edilen ve özelliğin değer ve tanım aralıklarının diğer ontoloji sınıfları olduğunu bildiren gösterimdir[135].

```
<owl:ObjectProperty rdf:ID="üzümdenYapılma">
<rdfs:domain rdf:resource="#Şarap"/>
<rdfs:range rdf:resource="#ÜzümŞarabı"/>
</owl:ObjectProperty>
```

Yukarıda tanımlanan özelliklere bağlı olarak “Şarap” sınıfı için “üzümdenYapılma” özelliği tanımlanmıştır. Özelliğin alabileceği değer adedi en az 1 olarak belirtilmiştir[135].

```
<owl:Class rdf:ID="Şarap">
<rdfs:subClassOf rdf:resource="#İçecek"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#üzümdenYapılma"/>
<owl:minCardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:minCardinality>
```

```
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

OWL'deki diğer özellik türü <owl:DatatypeProperty> işaretçisi ile tanımlanır. Bu işaretçi özelliğin basit veri türünden bir değer aldığını gösterir[135].

```
<owl:Class rdf:ID="ÜretimYılı" />

<owl:DatatypeProperty rdf:ID="üretimYılıDeğeri">
<rdfs:domain rdf:resource="#ÜretimYılı" />
<rdfs:range rdf:resource="&xsd;positiveInteger"/>
</owl:DatatypeProperty>
```

Sınıf tanımında bir özellik değeri içeren olgunun yaratımı aşağıda örneklenmiştir[135]

```
<ÜretimYılı rdf:ID="Yıl1998">
<üretimYılıDeğeri rdf:datatype="&xsd;positiveInteger">1998</yearValue>
</ÜretimYılı>
```

OWL ile birden çok sınıfı bir araya getirip koleksiyon (collection) tanımlamak mümkündür. Bu koleksiyon içindeki sınıflar için kesişim, birleşim ve tümleyen tanımları sırası ile <owl:intersectionOf>, <owl:unionOf> ve <owl:complementOf> işaretçileri ile yapılır. Aşağıda her üçüne ilişkin örnekler verilmiştir.

```
<owl:Class rdf:ID="BeyazKöpüklüŞarap">
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#BeyazŞarap" />
<owl:Class rdf:about="#KöpüklüŞarap" />
</owl:intersectionOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Meyve">
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#TatlıMeyve" />
<owl:Class rdf:about="#TatsızMeyve" />
</owl:unionOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Yiyecek">
<owl:complementOf rdf:resource="#İçecek" />
</owl:Class>
```

Koleksiyon içinde yapılabilecek bir diğer tanım numaralandırılmış (enumerated) tür tanımımızdır ve <owl:oneOf> işaretçisi ile belirtilir.

```
<owl:Class rdf:ID="ŞarapRengi">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Beyaz"/>
    <owl:Thing rdf:about="#Pembe"/>
    <owl:Thing rdf:about="#Kırmızı"/>
  </owl:oneOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Pasta">
  <owl:disjointWith rdf:resource="#Et"/>
  <owl:disjointWith rdf:resource="#Sebze"/>
  <owl:disjointWith rdf:resource="#İçecek"/>
  <owl:disjointWith rdf:resource="#Tatlı"/>
  <owl:disjointWith rdf:resource="#Meyve"/>
</owl:Class>
```

OWL ile özellikler için tersi ya da simetrik nitelermeleri yapılabilir. Ters nitelmesi eğer  $P(x,y)$  doğru ise  $P(y,x)$ 'de doğrudur anlamına gelir. Simetrik nitelmesi ise eğer  $P(x,y)$  doğru ise  $P(y,x)$ 'de doğru anlamına gelir. Örnek gösterimler aşağıda verilmiştir.

```
<owl:ObjectProperty rdf:ID="üreticisiVardır">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="şarapÜretir">
  <owl:inverseOf rdf:resource="#üreticisiVardır" />
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="komşuBölge">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:domain rdf:resource="#Bölge" />
  <rdfs:range rdf:resource="#Bölge" />
</owl:ObjectProperty>
```

Tez kapsamında geliştirilen ontoloji modeli ve üretilen alan ontolojisi OWL ile gösterilmiştir. Bu sebeple raporda OWL'nin çekirdek yapıları örnekler ile açıklanmıştır. Tam bir OWL belgesi için Ek-5'de ve Ek-6'da yer alan alan ontolojisi incelenebilir. Ontoloji sorgulama Anlamsal Web dillerinden biri ile ifade edilen ontolojilerin sınıf, ilişki ve olgularının filitrelenerek erişilebilmesidir. Bu noktada sorgulama dili olarak RDQL (RDF Data Query Languages) kullanılmıştır. İlerleyen kesimde ontoloji sorgulama üzerinde durulmuştur.

### 3.3.3. RDF Data Query Languages (RDQL)

RDQL, RDF ile gösterilen ontolojilerin sorgulanması için tanımlanmış bir dildir. Ontoloji gösterimi OWL ile yapılmış bile olsa diller arası kalıtım sıradüzeni sebebi ile RDF üçlülere biçimde gösterime çevrilmesi mümkündür. Başta Jena[151] olmak üzere birçok ontoloji kütüphane yazılımı bu desteği vermektedir.

RDF, modeli <kaynak, özellik, değer> üçlülere ile ifade eder. RDQL de bu üçlüde yer alan tüm bileşenler için filitreleyerek sorgulamayı destekler. Yani kaynaklar, özellikler ve değerler tek tek ve birlikte deyim mantığını bozmadan sorgulanabilir. Filtrelemek ile koşullu seçme ifade edilmektedir. Örneğin daha önce OWL ile ifade edilen modeli kullanarak üretim yılı 2002'den küçük şarapların adlarının erişimi istendiğinde, değeri 2002'den küçük olan üretim yılı özelliğine sahip şarap türü olguların adlarına erişilecektir. RDQL ile yazılmış örnek sorgu şu biçimde olacaktır:

```
SELECT ?x
WHERE (?x ns:üretimYılıDeğeri ?y)
AND ?y > 2002
USING ns: FOR file:///d:/sarap/
```

Görüleceği üzere sorgu RDF deyim mantığını kullanır. "WHERE" cümlecisinde parantez içinde yer alan üçlü aslında RDF'in <kaynak, özellik, değer> üçlüsüdür. Burada sorgulanan bileşen soru işareti ile bırakılıp diğer bileşenlerin adlarına ve değerlerine (var ise) yer verilir.

Tipik bir sorguda yer alacak kesimler ve açıklamaları şu biçimdedir[149,151]:

SELECT Cümlecisi

Sorgu sonucu dönecek olan değişkenlerin listesini içerir.

FROM Cümlecisi

Sorgulamanın yapılacağı model

WHERE Cümlecisi

RDF deyimlerini içerir.

AND Cümlecisi

Mantıksal ifadeleri içerir

USING Cümlecisi



Modelin adresine bir kısa isim (alias) atanır ve sorgunun diğer kesimlerinde model tanımlanmış sözcükler bu kısa isimle nitelenir.

RDQL için gerekli tek girdi sözdizim hataları içermeyen OWL-DL seviyesinde gösterimi yapılmış bir ontolojidir. Dolayısı ile ontoloji kütüphane araçları, ontolojilerin gösterimini OWL ile üretip, saklar, sorgular ve erişirler.

### **3.4. Ontoloji ve Biçimsel Diller**

Bu bölümde ontolojisini biçimsel bir dil ile ifade eden iki önemli çalışma örneklenecektir.

#### **Çoklu Ortam Bilgi Erişiminin Biçimselleştirilmesi ve Deneyselleştirilmesi (Formalization And Experimentation on the Retrieval of Multimedia Information-FERMI)**

FERMI projesi, 1994 yılında Glaskow Üniversitesi'nde çoklu ortam verilerinden görüntü ve metin türü veriler üzerinde, anlamsal sorgulamayı gerçekleştirme temel hedefiyle başlatılmış ve 2001 tarihinde prototipi tamamlanmıştır[88]. Kendinden önce gelen ve yine aynı ekip tarafından geliştirilen MIRLOG [86] projesinin devamı niteliğindedir. Bu sebeple sadece FERMI'yi incelemek yeterli görülmüştür.

FERMI, anlamsal sorgulama temel amacının yanında, o tarihe dek geliştirilen arama ve eşleştirme yaklaşımlarını da içermiş ve çoklu ortam verileri için varolan tüm teknikleri içeren bir çerçeve oluşturmayı hedeflemiştir. Dolayısı ile sistem metin türü veriler için klasik istatistiksel yöntemlere dayalı IR modelini benimsemekle birlikte; görüntüleri daha çeşitli düzeyde işleme imkanı sunmuştur. FERMI'de bir görüntüyü örnek ile, değer ile ve anahtar kelime ile sorgulamak mümkündür. Yani FERMI'de kullanıcı aramak istediği görüntü için örnek görüntü çizebilir, görsel özellik değerlerini kullanabilir ya da İngilizce doğal dilinde yer alan terimleri kullanabilir. Sistem genel olarak incelendiğinde, anlamsal sorgulamaya verdiği destekten ziyade varolan teknikleri toparlayıp içinde barındırması ile öne çıkmaktadır. Zira anlamsal sorgulamaya verdiği destek nesne tanıma yöntemlerinin tanıdığı nesne adlarını ve öncelikle genelleme ilişkisini kullanmaktan öteye geçmemektedir.

FERMI'nin hedeflediği belgeler metin ve/veya görüntü içermektedir. Bu noktada kısaca her iki verinin ele alınışı özetlenecektir. FERMI'de bir görüntü dört düzeyde ele alınmaktadır[87, 88]: yapısal görünüm'de görüntünün görsel özellikleri ve edinildiği ortamdan getirdiği özellikler; konumsal görünüm'de görüntünün içerdiği nesnelere arası konumsal ilişkiler; sembolik görünüm'de görüntüde yer alan nesnelere gerçek dünyada karşılığı olabilecek terimler ve algılanan görünüm'de görüntüyü ele alan kişinin görüntünün geneline ait yorumu bulunur. Dördüncü düzey, FERMI'de yer almakla birlikte desteklenmemiştir.

İlk iki düzey, üçüncü düzey için zemin oluşturmaktadır. Zira ilk düzeyde görüntü, görsel özelliklerine göre incelenmekte ve alanlarına bölünmektedir. Burada her bir alan muhtemel bir nesneye ya da nesne parçasına denk düşmektedir. İkinci düzeyde ise nesnelere arası konumsal ilişkileri çıkarılmaktadır. Dolayısıyla üçüncü düzeyde yer alan nesnelere adlandırma adımı için girdi oluşturma niteliğindedir.

Metin türü veriler için kullanılan yaklaşım klasik IR yaklaşımıdır. FERMI bu konuda yenilik önermemektedir. Belge için seçilen anlam gücü yüksek terimler, belgenin anlamsal özelliğini oluşturmaktadır.

FERMI anlamsal sorgulamayı gerçekleştirmek için kendi içinde "bağlam bilgisi" adını verdiği ve ACL tanımlama mantığı (description logic) ile oluşturduğu bir bilgi tabanı kullanmaktadır[88]. ACL ile kavramları, genelleme ilişkisini ve kavramların birbirlerine göre rollerini (yani bağıntıları) tanımlamak mümkündür. Ancak kısıtları, ön ve ardıl koşulları tanımlamak mümkün değildir. FERMI'nin kurduğu bilgi tabanında rollerden çok genelleme ilişkisi öne çıkmaktadır. Dolayısıyla bu bilgi tabanının bağlamın tam bir modeli yani ontolojisi olduğu söylenemez. Ancak bağlamda varolan kavramları ve kavramlar arası ilişkileri gösterdiği ve genişletilmeye açık bir bilgi tabanı olduğu aşikardır. ACL'in söz dizimsel kuralları şu şekildedir [88]:

kavram ::=  $\perp$  |  $\{ \text{en üst kavram} \}$   
 $\top$  |  $\{ \text{en alt kavram} \}$   
<basit- kavram > |  
<kavram>  $\cap$  <kavram> |  
<kavram >  $\cup$  <kavram> |  
r<kavram> |  
 $\forall$  <role>. <kavram> |  
 $\exists$  <role>. <kavram> |

FERMI, alan tabanlı nesne tanıma işleminde elde ettiği kavramları kendi bilgi tabanı ile ilişkilendirir. Anlamsal sorguları metinsel ifade biçiminde kabul eder ve bu ifade de yer alan terimler için bilgi tabanı üzerinden erişim çıktısı üretir[88].

FERMI her iki veri türü için de anlamsal özellik çıkarırken, süreçlerin birbirini etkilemesi öngörülmemiştir. Yani belgenin metin kısmına ait anlamsal özelliklerinin görüntülere ait anlamsal özelliklere ya da anlamsal özellik çıkarma sürecine etkisi yoktur[88].

FERMI, çoklu ortam verilerinin benzerliklerini araştırırken, mutlak benzerlik değil yakınlık araştırması yapmıştır. Bunun içinde bulanık mantık kurallarını kullanmıştır[88].

### **OntoBroker**

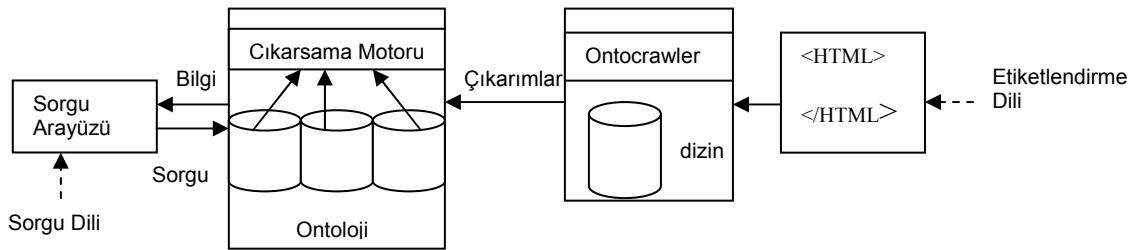
OntoBroker, Karlsruhe Üniversitesi'nin Uygulamalı Bilimler ve Biçimsel Tanımlama Yöntemleri Enstitüsü'nde gerçekleştirilen ve 1998 tarihinde bitirilen, temel hedefi WWW ortamında yer alan belgeler için anlamsal sorgulamayı bir ontoloji ile gerçekleştirmek olan bir projedir[83,84]. Web üzerinde anlamsal sorgulamayı bir ontoloji ile gerçekleştiren tipik bir sistemdir ve bu anlamda incelenmesi emsali olan diğer sistemleri de örnekleyebilmesi yönünde gerekli görülmüştür.

Projenin temelinde, WWW'de yer alan tüm belgeler hedeflenmiştir. Genel işlem mantığı; belgelerin anlamsal özelliklerini biçimsel bir yolla ifade etmek, bu ifade edilmiş anlamsal özellikleri çıkarsamak ve kurulan ontoloji ile çıkarsanan anlamsal özellikleri ilişkilendirmek biçimindedir. Ancak görüldüğü üzere, belgelerin içerdikleri veri türlerini ayırıştırma kaygısı yoktur. Genel olarak HTML formatında ele alınan belgelerin geneline ait anlamsal çıkarsamalar gerçekleştirilir.

Sistem üç temel bileşenden oluşur[83, 84]:

1. Sorgu Arayüzü: Kullanıcı düzeyine göre iki ayrı sorgulama biçimi mevcuttur. Gelişmiş kullanıcılar F-Logic[84] dili ile doğrudan sorgu hazırlayabilirler ya da verilen bir araç ile kullanıcı sınıf ve nitelikleri seçerler. Araç F-Logic diline uygun sorgu cümlesini üretir.
2. Anlamsal özellikleri tespit etme modülü (Ontocrawler)[83]: HTML sayfalarında, sayfayı hazırlayan kişinin, Ontobroker'ın HTML uzantılarını kullanarak yerleştiği anlamsal özellikler çıkarılır.
3. Çıkarılma Motoru: Ontocrawler tarafından çıkarılan ve dizinlenen özellikler ile varolan ontolojiyi birlikte kullanarak, kullanıcı sorgusuna karşılık erişim çıktısını üreten bileşendir.

OntoBroker'ın genel sistem mimarisi şekil 3.10 ile gösterilmiştir[84]:



Şekil 3.10. OntoBroker mimarisi

OntoBroker'da tanımlama dili (description language) olarak F-Logic kullanılmıştır. F-Logic içerme (has-a), genelleme (is-a) ve nesnelere arasında kurallara dayalı ilişki kurma, nicelik belirteçleri (FORALL, EXIST) ve mantıksal işlemleri (AND, OR, NOT) desteklemektedir. F-Logic kurallarını şu biçimde özetlemek mümkün olabilir[84]:

- Alt sınıf gösterimi ( $C1::C2$ ),  $C1$  sınıfı  $C2$ 'nin alt sınıfıdır.
- Sınıfa ait olgu gösterimi ( $O:C$ ),  $O$  nesnesi  $C$  sınıfının olgusudur.
- Sınıfa ait nitelik gösterimi ( $C1[A=>>C2]$ ),  $C1$  sınıfına ait  $A$  niteliğini değeri  $C2$  sınıfına ait bir olgudur.
- Niteliğe ait değer gösterimi ( $O[A=>>V]$ )  $O$  olgusunun  $A$  niteliğinin değeri  $V$ 'dir
- Parça bütün ilişkisi ( $O1<:O2$ ),  $O1$ ,  $O2$ 'nin bir parçasıdır.

OntoBroker içinde kullanılan ontoloji, içerik tabanlı ve alana özel hazırlanmıştır. Önceden tanımlanmış sınıflar ve yine önceden tanımlanmış çeşitli türde bağıntılar mevcuttur. Sayfasını hazırlayan kişi sayfa ile ilgili anlamsal özellikleri sayfanın içine HTML uzantıları ile yerleştirir. Bu maliyete tüm Web kullanıcılarının katlanmayacağı düşünülerek, belli bir alanda çalışma yapan kişilerin HTML sayfalarının oluşturduğu grup hedeflenmiştir. Örneğin OntoBroker için tipik bir ontoloji kesiti şekil 3.11’de gösterilmiştir:

Kavram Hiyerarşisi	Nitelik Tanımlaması	Kurallar
Nesne[].	Kişi [adı=>>STRING; soyadı=>>STRING; email=>>STRING; .....; yayın=>>Yayın].	FORALL Kişi1, Kişi 2 Kişi1:Araştırmacı [birlikteÇalışır->>Kişi2] <- Kişi2:Araştırmacı [birlikteÇalışır: Kişi1]
Kişi::Nesne[].	Çalışan [Üye=>>Orgizasyon;....]	FORALL Kişi1, Yayın1
Çalışan::Kişi.	Araştırmacı[araştırmaAlanı=>> AraştırmaKonuları; üyesi=>>AraştırmaGrupları; birlikteÇalışır=>>Araştırmacı].	Yayın1:Yayın [yazar->> Kişi1] <-> Kişi1:Kişi [yayın->>Yayın1]
AkademikKadro::Çalışan.	Yayın[ yazar=>>Kişi; başlık=>>STRING; yıl=>>NUMBER; özet=>>STRING].	
Araştırmacı::AkademikKadro.		
Yayın::Nesne		

Şekil 3.11. OntoBroker için ontoloji kesiti

Bu ontolojiyle uyumlu hareket etmek isten bir kişinin Web sayfası ilgili uzantıları kullanarak şekil 3.12’de görülen biçimde hazırlanabilir:

```
<html>
<a onto="page:Araştırmacı"></a>
<head><TITLE>Richar Benjamin</TITLE></head>
<H1><A HREF="pictures\xx.gif"> <IMG SRC ="photo.gif"></A>
<a onto="page[photo=href]">HREF=http://www.cs.hun.edu.tr/esezer/esezer.gif</a>
<a onto = "page[adı=body]">Ebru</a>
<a onto = "page[soyadı=body]">Sezer</a>
</h1><p>
<a onto="page[üye=body]">Bilgisayar Mühendisliği Bölümü</a>
.....
```

Şekil 3.12. OntoBroker için örnek HTML sayfası

OntoBroker'ı kullanarak arama gerçekleştirmek için kayıt olmak gerekiyor. Sistem kayıtlı sayfaları periyodik dolaşarak bilgileri güncelleştiriyor.

### **3.5. Ontoloji ve Unified Modelling Language (UML)**

Ontoloji gösteriminde biçimsel diller ve paylaşımına açık XML tabanlı diller incelenmiştir. UML her iki türden de farklı olarak, nesneye yönelik yaklaşım ile sistem geliştirmede kullanılan bir modelleme dili olarak ortaya çıkmıştır[73,74,75,76,77]. Ancak ontolojilere yönelimin artmasıyla birlikte UML bu konuda da dikkat çekmiştir. Zira temel tanım olarak bir ontolojiyi kısaca “belirli bir bağlama ait kavramsal modelin tam tanımıdır” diye ifade etmek mümkündür. Bu noktada ilk akla gelen bir sistem modelleyebilen UML'in ontoloji oluşturmak için de kullanılabilirliği. UML'in yaratıcıları Object Management Group (OMG) bu konudaki yaklaşım ve standartlarını, 2001 Eylül itibarı ile Meta – Object Facility (MOF) başlığı altında toplamaya başlamıştır[73,74].

Alternatiflerine rağmen ontoloji oluşturmak için UML'i tercih etmek için varolan sebepleri şu biçimde sıralamak mümkündür:

1. Nesneye yönelik yazılım geliştirmede; farklı iş alanlarında gerçekleştirilen analiz ve tasarım süreçlerinden kazanılan tecrübe UML'e yansıtılmıştır.
2. Bakımı OMG tarafından yapılmakla birlikte açık standarttır.
3. Özel bir uygulama ortamı için UML'e uzantı ekleme standartları tanımlanmıştır.
4. Sadece yapay us alanında varolan bilgi gösterim dillerinin aksine UML bugün birçok üniversitede ders konusudur. Tanınma ve kullanılma kriterlerine göre diğer dillerden üstündür.
5. Birçok tasarım aracı tarafından desteklenmektedir.

Sıralanan bu avantajların yanında şu hatırlatmayı da yapmak yerinde olacaktır. Ontoloji oluşturmak için gerekli olan sınıf / alt sınıf hiyerarşileri, sınıflar arası bağıntılar, sınıflara ait nitelik tanımlamaları ve kısıtlara ait aksiyomları UML ile belirtmek mümkündür. Ancak iki noktada UML OCL'den (Object Constraint Language) destek almaktadır:

1. Kısıtları belirleyen aksiyomları ifade etmek

## 2. Tamamen görsel olan UML çizeneklerini metin tabanlı ifadelere çevirmek.

Bu noktada aslında UML'in bir ontoloji için var olan iki dezavantajı da belirtilmiş oluyor. Bu dezavantajlardan kaynaklı olarak, UML ile bir ontoloji geliştirildiğinde, ontolojiyi sadece UML ile ifade edilmiş hali ile bırakmak mümkün değildir. Genelde tercih edilen OCL olmakla birlikte RDF ya da DAML; UML ile birlikte kullanılıyor. Dolayısıyla UML'in ontolojiyi ifade etmek için doğrudan kullanımında, bir başka gösterim dilinde –RDF, OCL, DAML+OIL- ifade etmek için görsel arayüz görevi gördüğünü söylemek yanlış olmaz. Bu rapor kapsamında UML sadece ontoloji oluşturma ve gösterme yönünden ele alınmıştır. Ancak diğer ontoloji ilintili çalışmalarda da (ontoloji eşleştirme, tutarlılık kontrolü vb gibi) kullanıldığını da vurgulamak gereklidir.

UML ile ontoloji gösterimi üzerine var olan çalışmalardan gerçekleştirim tarihleri ve konularına göre seçilen bir kaç tanesi şöyledir[73]:

- Cranefield ve Pruvit (2000) "UML Data Binding Tool", UML'in sınıf çizenekleri ile ontoloji gösterimi ve nesne çizenekleri ile olgu gösterimini gerçekleştirmiştir. Ayrıca XML ile kodlanan sınıf çizenekleri için RDF şemaları ve Java sınıflarına dönüştürüm de gerçekleştirilmiştir.
- Bergenti ve Pogii(2000), ajan tabanlı yazılım mühendisliği için UML tabanlı bir model önermişlerdir. Bu çalışmada önerdikleri "ontoloji çizeneği" dikkat çekicidir. Ontoloji çizeneği çoklu ajanlı sistemler için ajanları, alanda yer alan varlıkları, ilişkileri kodlar.
- Backlawski(2001), UBOT, DAML için metin etiketlerini doğal dil işleme yöntemlerince ele alan bir ontoloji oluşturma aracı geliştirdi. UML, DAML için görsel arayüz misyonunu üstlenmiştir.
- Sandpiper Yazılım (2001), UML ile ontoloji gösterimi üzerine bir çalışma gerçekleştirdi ve UML'e uzantılar ekledi.

UML'de sistemler toplam 12 adet çizenekle modellenir. Bunlardan dördü durağan, beşi dinamik, üçü de modül yönetimi olmak üzere üç temel noktadan hareket eder. Durağan modelin önde gelen çizeneği sınıf çizenekleri (class diagrams)'dir. UML'in sınıf çizeneklerinde; sınıflar varlıklara karşılık gelir ve bu çizenekle sınıfları, sınıflara ait nitelikleri ve sınıflar arası bağıntıları tanımlamak mümkündür. Sınıflar

arası bağıntı türleri şunlardır: kavramsal bağıntı (association), kümeleme (aggregation), oluşum (composition), genelleme (generalization), bağımlılık (dependency) ve gerçekleştirim (realization). Bunlardan özellikle ilk dördü, alan modellemede kullanışlıdır. Ayrıca çokluk bezemeleri ve adlandırma UML'de mevcuttur. Tüm bunlar ontoloji gösterimi için gereklidir. Ancak iki noktada sorun mevcuttur:

- Nesneye yönelik analiz ve tasarımda varolan nesnelere öne çıkarma alışkanlığı UML'de de mevcuttur. Örneğin bir bağıntı, bağıntıyı taşıyan sınıflar arasında anlamlıdır. Başka bir deyişle kaynak ve hedef sınıflar arasında bağıntı oluşturulur. Tek yönlü ya da çift yönlü olması bu noktada önemli değildir. Her ikisi için de iki sınıf ve aralarında mevcut bağıntı olmalıdır. Ancak ontoloji gösteriminde bağıntıların nesnelere önüne geçtiği görülmektedir. Yani bir bağıntı ve bu bağıntıyı taşıyan birçok ikili sınıflar olabilir. Ama aralarında taşıdıkları bağıntı aynı olduğunda, bağıntı merkezli bir tarza yönelim başlar. Örneğin "sorumlusu olmak" bağıntısı bir "araba" ve bir "şöför" varlığı arasında mevcut olabileceği gibi bir "ders" ve bir "eğitmen" arasında da olabilir ve aslında aralarında fark yoktur. İki sınıfı birbirine bağlayan "sahip olma" bağıntısı söz konusudur. UML de her iki bağıntı ayrı ayrı ele alınır. Yani bağıntı sınıfların bir özelliğidir. Oysa bu noktada bağıntıyı öne çıkarıp, "sahip olma" bağıntısını taşıyan varlıkları, bağıntının özelliği olarak görmek bir diğer yaklaşım olabilir. Ancak bu UML'de doğrudan desteklenmemiştir.
- Ayrıca UML kesin bir şekilde metin tabanlı bir gösterim diline ihtiyaç duymaktadır. Hem kısıtları, ön ve ardıl koşulları tanımlamak hem de varolan çizenekleri ifade etmek için metin tabanlı bir dil desteği gereklidir. Bu sebeple tek başına bilgi erişiminde doğrudan kullanımı olası değildir. Mutlaka çizeneklerin metin tabanlı bir dile çevrimi gerekmektedir.

Ancak görsel ifadenin model okuyucusunu rahatlattığı ve okunabilirliği artırdığı kabul edilebilir. Bu sebeple tez kapsamında geliştirilen modeller raporda UML ile gösterilmiş ve karşılığı olan RDFS ve OWL gösterimleri raporun eklerine yerleştirilmiştir.



### 3.6. Ontoloji ve ERW (Entity and Relations on the Web)

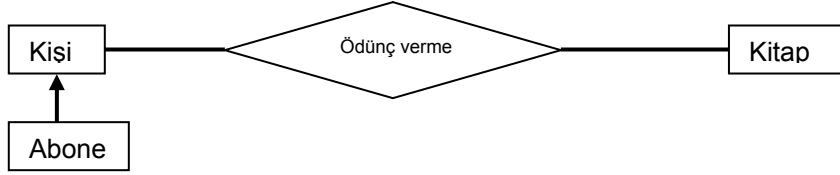
Nesneye yönelik yaklaşımdan önce de varlıkları ve bağıntılarını modellemek söz konusuydu. Varlık bağıntı çizimeleri (Entity Relational Diagrams - ERD), ilişkisel veri tabanı modelinin gelişmesi ile birlikte bu amaca hizmet ettiler. Yani gerçek dünyayı modelleyerek sistemin ihtiyaç duyacağı veri tabanı tasarımında kullanıldılar. Yani modellemeye hizmet ederler. ERW de bu temel hedeften çok uzaklaşmaz. ERW ile büyük ve karmaşık veri tabanlarını web tarayıcısı kullanarak modellemek ve yönetmek mümkündür. Modelleme üzerine halen varolan yaklaşımların örneklendiği bu bölümde, ERW yaklaşımının da yer alması uygun bulunmuştur.

ERW, web üzerinde veritabanı uygulaması gerçekleştirirken kullanılan, geleneksel varlık bağıntı ilişkisini ele alan bir yaklaşımdır. XML tabanlı Varlık Bağıntı Dili (Entity Relational Language – ERL ) kullanılır. Hatta ERL ile ifade edilen modeller ERTTool adı verilen, Java ile gerçekleştirilmiş bir araçla, PHP ile ilgili model için gerekli tabloları yaratan kodları otomatik üretilebilir. Böylece SQL ya da PHP bilmeksizin ya da kullanmaksızın ilgili modele ait kod kesimi elde edilmiş olunur. Bu otomatik kod üretimi sayesinde kullanıcılar modele daha yakın ve hakim olabilirler. Zira onları ilgilendiren sadece varlıklar ve bir varlığın diğer varlıklar ile kurduğu bağıntılardır[92].

ERW'nin temel özellikleri şunlardır[92]:

- ERW uluslararası standartlara uygun ve açık kodludur. Yazılımı parasız elde edilebilir.
- ERW mantıksal ve referans bütünlüklerini destekler.
- ERW, grafiksel arayüzü için W3C DOM'u kullanır

Örnek bir E-R çizeneği şekil 3.13.a'da ve şekle uyumlu ERL kütüğü şekil 3.13.b'de verilmiştir[92]:



Şekil 3.13.a: Basit bit ER çizeneği

Şekil 3.13.a'da verilen örnekte ifade edilen anlamsal bilginin ERL ile HTML sayfaları içinde ifade edilmesi Şekil 3.13.b ile gösterilmiştir.

```

<? Xml version = 1.0?>
<! DOCTYPE erl OUBLIC "-//DSI//DTD ERL V1.0 //EN" ""erl.dtd>
<erl id="library" title "A Library">

<enum id = "dönemuzunluğu" type = "char" size = "1">
    <enumval value="U" label="UzunDönem"/>
    <enumval value="K" label="KısaDönem"/>
</enum>
<ent id="kişi" label= "Kişi">
    <attr id = "adı" label="Adı" size = "30" mand ="true"/>
    <attr id = "sadı" label="Soyadı" size = "30" mand ="true"/>
</ent>
<ent id="abone" label= "Abone" is = "kişi">
    <attr id = "kart" label="KartNo" type = "int" mand ="true"/>
    <attr id = "adres" label="Adres" mand ="true"/>
</ent>
<ent id="kitap" label= "Kitap" >
    <attr id = "başlık" label="Başlık" mand ="true"/>
    <attr id = "yazar" label="Yazar" mand ="true"/>
    <attr id = "yayınevi" label="Yayınevi" mand ="true"/>
    <attr id = "isbn" label="ISBN" mand ="true"/>
</ent>
<rel id="ödünçverme" label= "ÖdünçVerme" >
    <attr id = "başlangıçtarihi" label="Başlangıç Tarihi" type ="date" mand ="true"/>
    <attr id = "bitiştarihi" label="Bitiş tarihi" type = "date" mand ="true"/>
    <attr id = "süre" label="Tür" type="enum" enumref="termlen" mand ="true"/>
    <leg entref="kişi" label="ÖdünçAlındı..."/>
    <leg entref="kitap" label="eVerildi..."/>
</rel> </erl>

```

Şekil 3.13.b: Şekil 3.13.a ile uyumlu ERL kütüğü

Örnekte genelleme bağıntısı gösterilmiştir. Ancak ERW, içirme bağıntısını da desteklemektedir. Bu iki özel tür bağıntının dışında kalan her tür bağıntı, ödünç verme örneğinde görüldüğü üzere desteklenmektedir. Ancak sadece tekil ve ikili ilişkiler tanımlamak mümkündür. Daha fazla sayıda çoklu söz konusu ise, ikililere bölerek ele almak gerekmektedir.

ERW Netscape 6.0 ve üsü Internet Explorer 5.5 ve üstü tarafından desteklenmektedir.

Şimdiye kadar anlamsal web üzerinde ontoloji gösteriminde ERW kullanılmamıştır. Zaten ERW yaklaşımın anlamsal web bağlamında kullanılma hedefi yoktur. Ancak ERW öz itibarı ile bir web veri tabanı için model ifade etmektedir ve yazar için fikir verici olmuştur.

## **BÖLÜM 4. WEB SAYFALARI İÇİN ANLAMSAL ERİŞİM SİSTEMİ**

Bu bölümde tez kapsamında HTML ile gösterimi yapılmış web sayfaları için geliştirilen anlamsal erişim sisteminin genel yapısı, işleyişi, kullandığı ontoloji modeli, ontoloji kullanım biçimi, dizinleme ve sorgulama aşamaları ayrıntılı açıklanacaktır. Bölüm 5'te sistemin başarımını ölçmek için düzenlenen deneyler ve sonuçları açıklanıp, yorumlanacaktır.

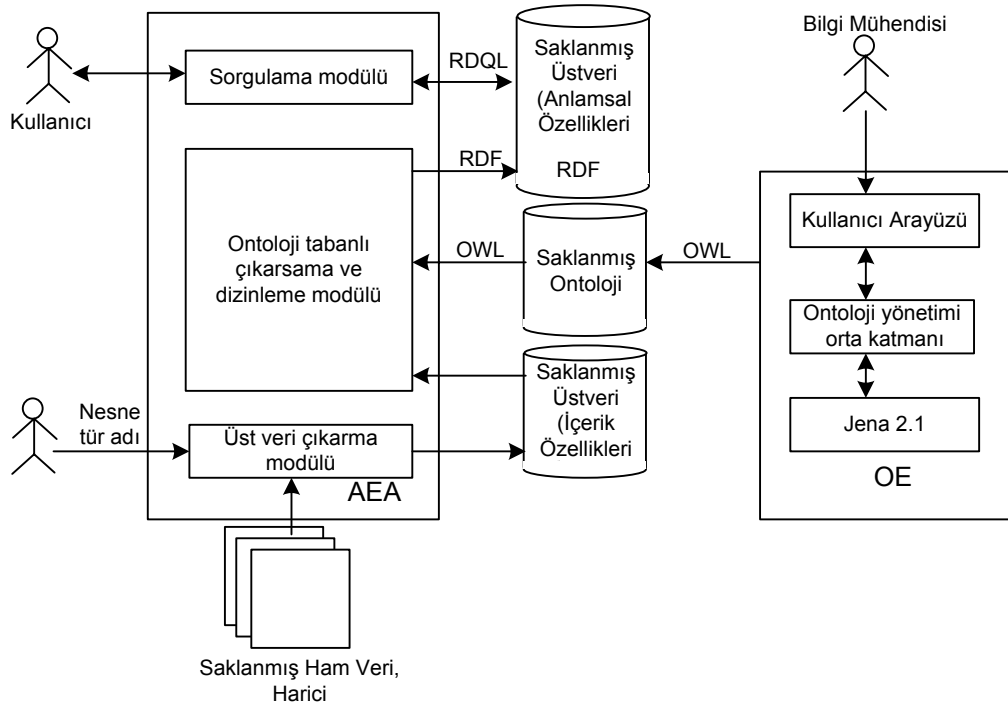
### **4.1. Sisteme Genel Bakış**

Geliştirilen anlamsal erişim sistemi, en genel hatları ile web sayfaları için anlamsal üst verileri oluşturur ve sorgulamaya açar. Anlamsal üst veri üretiminde bir alan ontolojisi kullanılmıştır. Yani sistemin iki ana ve ayrı parçası ontoloji yönetimi ve sayfa dizinlemedir. Her ikisi için iki ayrı araç geliştirilmiştir.

Ontoloji yönetimi, Ontoloji Editörü (OE) adlı bir araç ile gerçekleştirilir. Bu araç, önerilen NEK (Nesne, Eylem, Kavram) Modeli ile seçilen bir alanın modellenmesi, modelin OWL ile gösterimi ve saklanması, saklanmış ontolojinin yüklenmesi ve her düzeyde değiştirilebilmesi yeteneklerini içerir. Görüleceği üzere bu aracın girdisi uzman bilgisi ve çıktısı, ontolojinin OWL ile gösterimidir. OE, ontoloji yönetiminde kütüphane yazılımı Jena 2.1'i [151] kullanır.

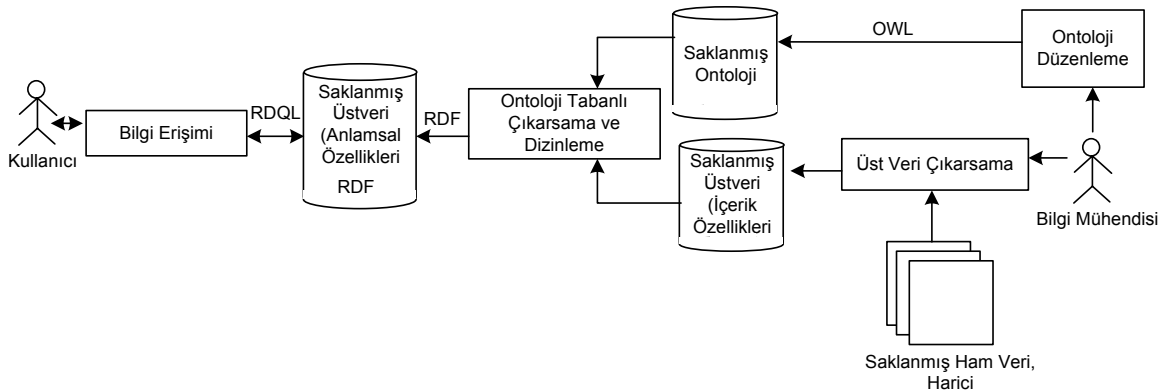
Dizinleme ve erişim birlikte düşünülmüş, Anlamsal Erişim Aracı (AEA) adlı bir araç ile gerçekleştirilmiştir. AEA, üç ana modülden oluşmaktadır: üst veri üretimi, ontoloji kullanarak dizinleme ve sorgulama. AEA'nın ilk girdisi web sayfalarıdır. AEA ile web sayfalarının görüntü ve metin parçaları ayrıştırılıp, her iki parça için içerik özelliklerinden oluşan üst verileri üretilir. Bu noktada görüntü üst verilerinin üretiminde, kullanıcının görüntü içindeki nesnelere tür adları ile adlandırması gerekir. Yani otomatik nesne tanıma gerçekleştirilmez. AEA, görüntüler için nesnelere ve konumsal ilişkiler; metinler için terimlerden oluşan üst verileri üretir. Üretilen üst veriler, alan ontolojisi ile ilişkilendirilip anlamsal çıkarım yapılır. Elde edilen anlamsal özellikler üst veriye eklenir. Yani son aşamada, sistemin ürettiği anlamsal üst verilerden bir derlem oluşturulur. Üst verilerin gösterimi RDF ile yapılır. Sorgular, üst veri derlemi üzerinden cevaplanır. Sorgulama dili RDQL'dir.

AEA ve OE araçlarının birbirleri ile ilişkileri ve modülleri şekil 4.1’de gösterilmiştir.



Şekil 4.1: Anlamsal erişim sistemi araçları, modülleri ve ilişkileri

Şekil 4.1’de ki gösterimde araçlar ve modüller esas alınmıştır. Aslında şekil dolaylı olarak sistemdeki bilgi akışını da göstermektedir. Ancak okuyucu için bilgi akışını yatayda gösteren ve sistemin Anlamsal Web’e uygun bilgi yönetimi mimarisine sahip olduğunu vurgulayan çizime şekil 4.2 de yer verilmiştir. Şekil 4.2 bilginin akış sıralamasını ve düzenini ifade etmektedir.



Şekil 4.2: Sistemin bilgi yönetim mimarisi

Şekil 4.2’de görüleceği üzere, Anlamsal Web katmanlarına [bkz. Bölüm 3.1] uygun olarak öncelikle web sayfaları için üst veri üretilmiştir. Daha sonra var olan alan ontolojisi ile üst veriler ilişkilendirilmiştir. Bu noktada elde edilen anlamsal özellikler de üst veriye eklenmiştir. Sorgular anlamsal özellikleri de içeren üst veriler üzerinde işlenir.

Paylaşılabilirlik özelliği Anlamsal Web dilleri [bkz. Bölüm 3.3] kullanımı ile sağlanmıştır. Üst verilerin gösterimi RDF ile, ontoloji gösterimi OWL ile gerçekleştirilmiştir. Tüm bu özellikleri sebebiyle çalışmanın Anlamsal Web ile uyumlu olduğu tespiti yapılabilir.

İlerleyen bölümlerde sırası ile açıklamalar boyunca kullanılacak örnekler, Ontoloji Editörü (OE) ve Anlamsal Erişim Aracı (AEA) ayrıntılı tanıtılacaktır.

## **4.2. Örnek Uygulama Alanlarının Tanıtımı ve Bulanıklığa Giriş**

Tez kapsamında önerilen NEK Ontoloji modeli ile iki ayrı alanın ontolojileri üretilmiştir: “Gövde”, “Voleybol”. Her iki alan da NEK Modelinin açıklamalarında, anlamsal çıkarımın gerçekleştiriminde örnek olarak ve sistemin başarımlarında deneylerinde konu olarak kullanılmıştır. Bu sebeple her iki alanın tanıtımı yapılarak, gerek bulanıklık kavramının gerekse ilerleyen kesimlerin anlaşılabilirliğini artırmak amaçlanmıştır.

Gövde tıp bilimi bağlamında ele alınmıştır. Bu bölge bir çok organ ve organ kesimini içine almaktadır. Hatta işlevsel olarak birbiri ile ilgili organların birlikteliği ile bazı sistemleri örneğin sindirim sistemi gibi tamamen kapsamaktadır. Vucudun gövde bölümü toraks ve abdomen olmak üzere ikiye ayrılır. Toraks boğaz altından başlayıp akciğerlerin bittiği, diyaframın olduğu kesimi ve abdomende akciğerlerin bittiği kesimden gövdenin bittiği kesimi hedefler. Şu durumda gövde, toraks ve abdomenin birleşimidir denilebilir. Toraks içinde başlıca sağ akciğer, sol akciğer, kalp, abdominal aort, vertebra korpus ve medulla spindins yer almaktadır. Abdomen içinde başlıca karaciğer (toplam 9 adet karaciğer kesimi), mide (mide gövdesi ve mide bağlantısı), pankreas (pankreas gövdesi ve pankreas başı), dalak, özofagus, çölyek, sol hepatik ven, sağ hepatik ven, orta hepatik ven, vera

cava inf., splaenik arter, falsiform ligament, superior mezanterik arter, protal ven, sağ adrenal, sağ böbrek (sağ böbrek korteks ve sağ böbrek pelvis), sol böbrek (sol böbrek korteks ve sol böbrek pelvis), safra kesesi (safra kesesi fundus ve safra kesesi korpus), koledok, duodonum, abdominal aort, vertabra korpus ve medulla spindins bulunmaktadır. Her iki bölümde adı geçen tüm organ ve organ kesim adları, abdomen bölgenin CT görüntülerinde yer almaktadır.

Gövde, yukarıda adı geçen organlara ilişkin tetkik işlemlerinde CT ile görüntülenmektedir. Gövdenin CT ile görüntülenmesinde, gövdenin başından sonuna dek sık alınan kesitler ile birbiri ardına kesit görüntüleri üretilir. Her bir görüntü, dcom ya da benzeri bir biçim ile görüntüleme sonuçlarının yer aldığı özel sunucularda saklanır, arşivlenir. Ayrıca doktorların görüntüleri inceleme esnasında da dikkat ettikleri husus görüntüde yer alan organ ve organ kesimleridir.

Saklanmış görüntülerin erişimi, sorgulanması için abdomen, toraks, üst abdomen gibi vucuda ait bölge adları; dalak, sol hepatik ven... vs gibi organ/kesim adları, sindirim sistemi gibi organ birliktelikleri ya da karaciğer gibi birçok kesimden oluşan organ adlarının kullanılmasının gerekliliği düşünülmektedir. Bu gereklilikten hareketle bölüm 4.3'ten itibaren görülecek uygulama örnekleri, bahsi geçen tür sorgulamaları desteklemeye çalışmaktadır.

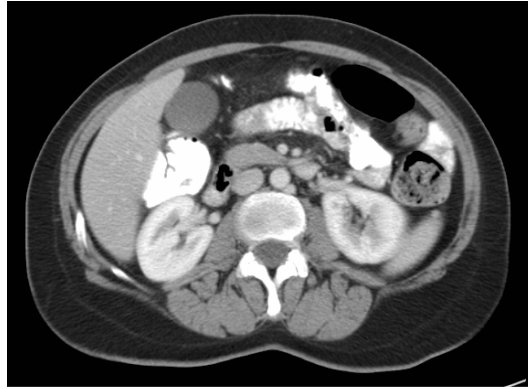
CT görüntüleri üzerinde yapılacak sorgulamaların işlenmesi bulanık olmalıdır. Örneğin üst abdomene ait görüntülerin erişimi istendiğinde, aslında istenen organların üst abdomen bağlamında yer alan organ ve kesimleri olduğu çıkarsanmalıdır. Dolayısı ile görüntü üst abdomene ait organları içerdiği oranda üst abdomen bağlamını ifade etmektedir. Hatta bir görüntünün Şekil 4.3'te görüleceği üzere hem üst abdomen hem de toraks bölgeyi içermesi mümkündür.



(a)



(b)



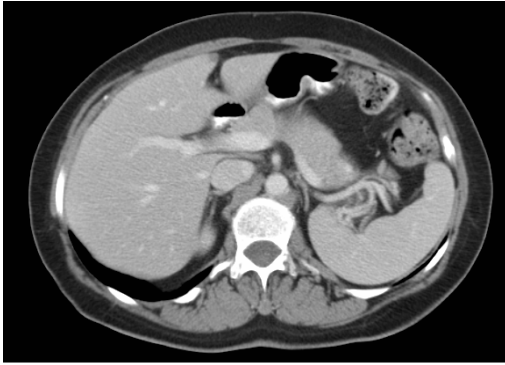
(c)

Şekil 4.3: Abdomene ait üç farklı CT görüntüsü

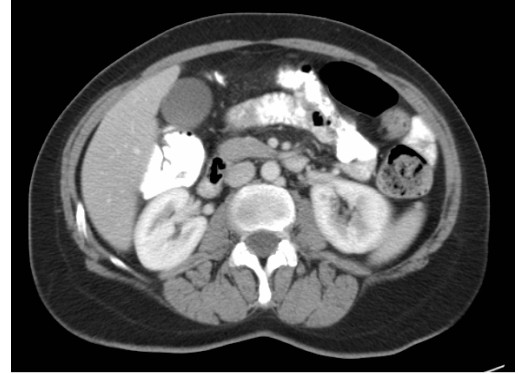
Şekil 4.3.a'da yer alan görüntü aslında bir toraks görüntüsüdür ancak üstabdomen bölgeyi de içermektedir. Aynı biçimde şekil 4.3.b'de aslında bir üst abdomen görüntüsüdür fakat aynı zamanda kısmen toraksı da içermektedir. Şekil 4.3.c ise sadece üst abdomene aittir. Bu üç şekilde görüldüğü üzere toraks görüntülerine erişim hedeflendiğinde ilgili erişim çıktısının şekil 4.3.a'yı yüksek ilgililikle, şekil 4.3.b'yi daha düşük ilgililikle içermesi ve şekil 4.3.c'yi içermemesi gerekmektedir. Erişim çıktısında gerçekleşmesi gereken bu sıralama bulanık yaklaşım ile gerçekleştirilebilir. Gövde için bulanıklığın gerekliliği bir grup organın ifade edildiği sistem adları ile gerçekleştirilen sorgulamalarda ya da bir grup kesimin birlikte oluşturduğu organların sorgulamasında da kendini hissettirir. Örneğin sindirim sistemine ait görüntülerin erişiminde, sindirim sisteminde yer alan tüm organların (karaciğer, dalak, safra kesesi, mide, pankreas) görüntüde yer alma oranlarına göre erişim çıktısı sıralanmalıdır. Tüm organları içeren görüntü elbette en ilgili görüntülerden biri olacaktır. Ancak bir organ tam yer almıyorsa bile görüntü yine sindirim sistemine aittir. Ancak belli bir eksiklikle yani ilgi derecesinde düşme ile görüntünün aitliği çıkarılmalıdır. Bir başka örnek karaciğer gibi birçok kesimden



bir araya gelen organların erişimi üzerine verilebilir. Karaciğer toplam 9 kesimden oluşur ve her bir kesimin birbiri ile konumsal ilişkisi vardır. CT görüntülerinin karaciğer ile ilgiliği, barındırdıkları kesim ve doğru konumsal ilişki sayısı ile doğru orantılı olmalıdır. Yani aslında karaciğerin, toplam 9 kesimden ve konumsal ilişkilerinden oluşan bir tanımı vardır. Ancak karaciğer organını içeren CT görüntülerinde hepsi yer almayabilir. Şekil 4.4 karaciğer organına ait örnek CT görüntüleridir.



(a)



(b)

Şekil 4.4: Karaciğere ait iki farklı CT görüntüsü

Şekil 4.4.a'da karaciğer 6 ayrı kesim ile görüntü üzerinde önemli bir yere yani vurguya sahiptir. Şekil 4.4.b'de ise karaciğer, 2 ayrı kesimi ile yer almaktadır. Her iki görüntüde de karaciğerin varlığı açıktır ama varolma dereceleri farklıdır. Karaciğer üzerine gelen bir sorgunun her iki görüntüye erişilmesi, ancak a şıklı görüntünün b'den daha ilgili olduğunu tespit edilmesinin gerekli olduğu düşünülmektedir.

Buraya kadar yapılan açıklamaları özetlemek gerekirse, gövde iki ayrı alt bölgeden oluşmakta, her bölgede ayrı ayrı yer alan basit (tek parça) ya da karmaşık (kesimlerden oluşan) organlar bulunmaktadır. Her organ ya da kesimi tanımlanabilir konumsal ilişkiler taşımaktadır. Ayrıca abdominal aort, vertebra korpus ve medulla spindins gibi yapılar tüm gövde boyunca yer almaktadır. Gövdeye görüntülerin erişiminde terimlerin birbiri arasında var olan parça-bütün, konumsal ilişkileri ve bağlam ilişkilerinin önemli olduğu düşünülmektedir. Örneğin sol böbrek iki ayrı kesim olan sol böbrek pelvis ve sol böbrek korteks'ten oluşur.

Dolayısı ile sol böbrek pelvis ile sol böbrek korteks arasında anlamsal bir ilişki vardır: her ikisinde aynı organın parçasıdır. Ayrıca her ikisi ile sol böbrek arasında parça bütün ilişkisi vardır. Bir başka örnekle; abdomen bir bölgeye verilen addır ve abdomenin üst kesimi olan karaciğer, mide gibi organlar ile abdomenin alt kesiminde yer alan uterus, mesane gibi organlar birbiri ile anlamsal ilgilidir. Abdomen, anlamında tüm bu organları barındırır. Yani abdomen terimini kullanmak tek başına tüm bölgeyi ifade eder. Aynı şekilde örneğin mide teriminin de abdomeni ifade etme gücü vardır çünkü bağlamında yer alır. Bu tür anlamsal ilişkilerin ele alınması, görüntünün içerdiği kesimlerden organlara, organlardan sistemlere ve bölgelere ulaşabilmeyi yani görüntüyü yorumlayabilmeyi olanaklı kılar. Böylece bu anlamsal çıkarım zincirinde yer alan her bir aşama ayrı ayrı sorgulanabilir.

Sistem için diğer bir örnek konu da voleybol alanıdır. Bu kesimde voleybol alanının bilinirliğinin yüksek olması sebebiyle voleybol sporu ile ilgili ek açıklama verilmeyecektir. Ancak abdomen alanda varolan bulanıklık gereksiniminin aynı biçimde voleybol alanı içinde geçerli olduğu kanaati açıklanacaktır. Voleybolda sıkça karşılaşılan somut terimler top, saha, voleybolcu, file'dir. Ancak bu somut terimler voleybol sporuna ait anlık görüntülerin yorumlanmasına yetmez. Voleybola varolan iki ayrı topa vuruş biçimi olan manşet ve smaç için neredeyse aynı nesnelere ile iki ayrı tanım yapılabilir. Tanım farklılığı nesnelere değil konumsal ilişkilerden kaynaklanmaktadır. Ancak bulanıklığın kendini daha çok hissettireceği asıl örnek şu biçimde verilebilir: smaç voleybol bağlamında bir topa vurma biçimidir ve tam topa vurma anının bir tanımı yapılabilir. Ancak topa vurduktan çok kısa bir süre sonra konumsal ilişkiler değişir. Fakat olayın smaç ile anlamsal ilgisi bitmemiştir. Topa vurma anındakine göre daha düşük derece ile de olsa topa vuruş anı sonrası da smaç ile ilgilidir. Şekil 4.5.'te smaç eylemine ait üç ayrı görüntü yer almaktadır.



(a)



(b)



(c)

Şekil 4.5: Smaç anına ait örnek görüntüler

Şekil 4.5.a'da yer alan görüntüde smaç biçiminde topa vurma anı tam yer alırken, şekil 4.5.b'de smaç biçiminde topa vuruş gerçekleştirilmiştir. Resimdeki anlam halen bir smaç olmakla birlikte, smaç sonrasını ifade etmektedir. Şekil 4.5.c ise sporcunun duruşu itibarı ile bir smaç görüntüsü olabilir. Bu üç ayrı yorumun erişim çıktısına yansımaları için resimdeki her bir somut nesnenin, smaç eyleminin tanımı ile bulanık karşılaştırılması ve bulanık derecelendirilmesi gerekir ve beklenen erişim çıktısında da ilgi sıralamasının azalan biçimde (a), (b) ve (c) düzeninde olmasının gerekliliği düşünülmektedir. Görüleceği üzere voleybol alanında öyle terimler vardır ki (örn. smaç) görüntüdeki varlıkları, alandaki somut nesnelere ve nesnelere konumsal ilişkilerine göre tespit edilebilir. Ayrıca smaç tanımında kullanılan nesnelere aynı zamanda smaç ile anlamsal ilgiye de sahiptir. Voleybol alanında verilen smaç örneği ile devam edilirse, topa bu biçimde vurmaya karşı takımın hücum etmek anlamına gelir. Yani voleybol bağlamındaki hücum terimi, kendi bağlamında smaç terimini içerir. Hücum sadece smaçtan oluşmayabilir, hücum bağlamında smaç vardır. Yani iki terim arasında yine bir anlamsal ilgi vardır. Dolayısıyla ile şekil 4.5'te yer alan görüntüler sadece smaç biçimi topa vuruşu değil, aynı zamanda hücum yorumunu da ifade ederler. Şekil 4.5'te yer alan nesnelere smacın çıkarsanması ve smaçtan da hücum yorumuna erişilebilmesi için, smaç anının nesnelere bağlı tanımı ve smacın hücum ile bağlam ilişkisi belirlenmelidir. Bağlam ilişkisi doğası gereği bulanıktır. Örneğin smaç serviste bir hücum mudur ya da smaç ile smaç servis aynı oranda hücum bağlamına girer mi? Bu sorunun doğal dille ifade edilecek cevabı da bulanık ifadeler içerecektir.

Bu kesimde her iki örnek alan tanıtılmıştır. Her iki örnek ile anlamsal çıkarsamada yer alması gereken bulanıklıktan ve anlamsal çıkarsamanın düzeylerinden giriş seviyesinde bahsedilmiştir. İlerleyen kesimde sistemin anlamsal çıkarım tabanı olan NEK Ontoloji modeli, bu modele dayalı ontolojiler ile gerçekleştirilen anlamsal çıkarım ve erişim biçimi ayrıntılandırılmıştır.

### **4.3. NEK (Nesne, Eylem, Kavram) Ontoloji Modeli<sup>15</sup>**

Geliştirilen sistemde, anlamsal çıkarıma hizmet edecek ontoloji için temel sınıf ve ilişki tanımları yapılmıştır. Bu tanımlara ontoloji modeli adı verilmektedir. NEK (Nesne, Eylem, Kavram) kısaltması ise ontoloji modelinde yer alan üç temel sınıftan esinlenilerek kullanılmaktadır. NEK Ontoloji modeli üzerine yapılan çalışmanın temel iki sebebi vardır:

1. Alan ontolojilerini üretirken genel yaklaşım bir uzmanın tüm kavramsal modeli ontolojiye yerleştirmesidir. Aynı zamanda ontolojiyi kullanarak üretilen olgular da ontolojideki modellemeye bağlı kalınarak etiketlenilmektedir. Ontolojiyi elle üretmek zaten zaman alıcı zahmetli bir iş iken ontoloji olgularını üretmenin otomatik olması gerektiği düşünülmektedir. Bu sebeple NEK Ontolojisinin kaygılarından biri olguların otomatik üretilebilmesidir.
2. NEK ontoloji modeli ile alana bağlı ontolojiler üretilir. NEK Ontolojisinin bir alanda yer alan terimleri ele alma biçimi anlamsal katmanlar bazındadır ve alana ait bağlamı en kapsayıcı ve tanımı en geniş terimlerinin yanı sıra bağlamı en dar ve tanımı en net terimler de ele alınmıştır. Örneğin “abdomen” alanda en dar anlamlı terimler “dalak” gibi basit organ adlarından biri olabilirken, “toraks” tüm alt gövdeyi tanımlayan ve bir çok organı içine alan bir terimdir. Aynı şekilde “voleybol” alanı için en dar anlamlı terim “voleybol topu” iken en geniş anlamlı terim “takım sporu” olabilir. Ayrıca NEK Ontoloji modeli ile üretilecek ontolojinin uygulama alanı ele alınmamış ya da bir uygulama alanı için model önerilmemiştir. Şöyle ki; eğer müze eserlerinin sayısal

---

<sup>15</sup> Ontoloji için verilen örnekler “Voleybol” ve “Tıpta Gövde” alanlarından seçilmiştir. Sistemin başarımını ölçmek için düzenlenen deneyler bu iki alana aittir ve ilgili alanların ontolojileri ek-5'te ve ek-6'da OWL ile gösterilmiştir.

kayıtları için özel bir model önerilecekse sanatçının adı, eserin adı, eserin dönemi ve yılı gibi birçok alana has varlık ve nitelik modele girecektir. Elde edilen ontoloji müze eserlerine özeldir yani uygulama alanına bağlıdır. Ancak NEK burada temel modelleme biçimiyle tanımlanmıştır ve bu model ile alana bağımlı ontoloji üretilir, bununla birlikte aynı temel kullanılarak farklı alanlar için modelleme yapılabilir.

Özellikle, madde 2 ile açıklanan karar stratejik bir karardır. Şöyle ki; eğer uygulama alanına ait bir ontoloji modeli geliştirilirse elbette elde edilen ontoloji son derece ayrıntılı nitelik ve bağıntılara sahip olabilecektir. Çünkü alan için özelleşecektir. Ancak NEK web sayfaları için anlamsal erişimin temelini oluşturmaktadır, bu sebeple NEK bir alana özel değildir, NEK ile oluşturulan ontolojiler alana bağımlıdır.

NEK Ontoloji modelinde terimlerin sınıflandırılmasından hareket edilmiştir. Bir alanda olabilecek terim türlerinden ontolojiye girmesi gereken türler tespit edilerek modelleme başlamıştır. Çünkü NEK Ontoloji modeli ile bir alan için anlamlı web sayfalarının dizinlenmesi önerilmiştir ama x alanı olmalıdır gibi bir zorlama yapılmamıştır. Bu karar da ontolojide kullanılacak niteliklerde ve bağıntılarda bir azalma yaratmıştır. Ancak bu azalma dezavantaj gibi ele alınmamaktadır. Çünkü bu sayede ontoloji olguları otomatik üretilmektedir.

NEK Ontolojisinin bir diğer özelliği metin ve görüntü veri türlerinin anlamsal özelliklerini keşfetmeye hizmet edecek ortak bilgi tabanı olmasıdır. Yani model önerilirken modelin hizmet edeceği veri türleri de dikkate alınmıştır. NEK Ontoloji olgularının otomatik üretilmesi için web sayfasından beklenen girdiler metinlerin terimleri ve görüntülerin içerdiği nesne adları ve konumsal ilişkileridir. Sadece bu iki girdi türüne bağlı kalınmıştır.

NEK Ontolojisi bir alanda yer alan terimleri 3 temel sınıfta ele alır:

- Nesnelere (Objects): Bir alanda yer alan somut nesne adlarıdır ve ontolojideki en dar anlamsal genişliğe sahip terimlerdir.

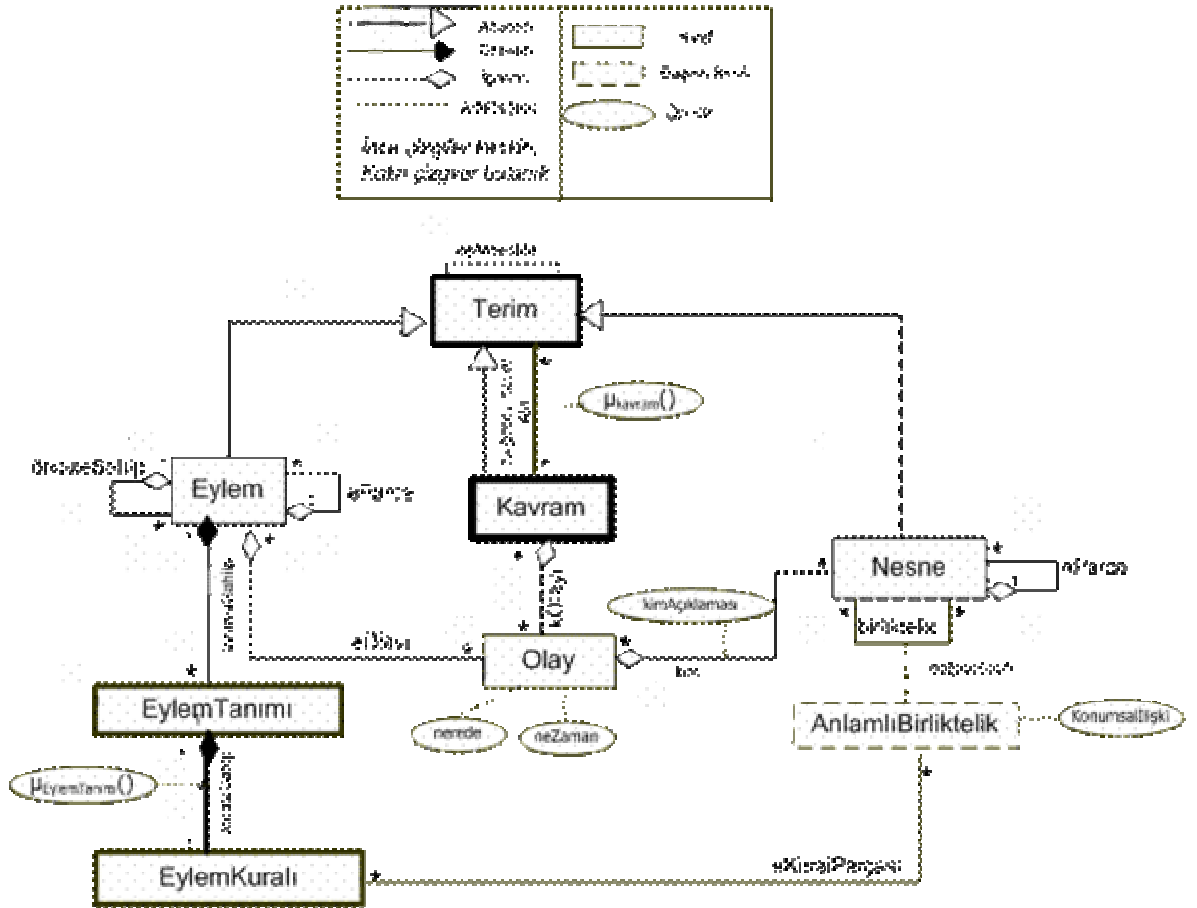
- Eylemler (Actions): Sadece nesnelere ve konumsal ilişkiler kullanılarak tarif edilebilen alan terimleridir. Bir başka deyişle an adlarıdır (instant name).
- Kavramlar (Concepts): Alan için nesne ve eylemlerin dışında kalan ve daha geniş anlamsallığa sahip olan terimlerdir. Bu terimler kendi başlarına bir anlamı (meaning) ifade ederler ve başka kavram, eylem ve nesnelere bu anlama dahildirler.

NEK Ontolojisinde 3 temel sınıfı tamamlayıcı olarak 2 sınıf daha yer almaktadır:

- Konumsal İlişkiler (Spatial Relations): Konumsal ilişkiler üstünde, altında, yanında, içinde, yakın, uzak vb. gibi iki nesnenin birbirine göreli konumunu ifade etmek için kullanılabilen sözel değişken kümesidir. NEK Ontolojisinde kullanılacak konumsal ilişkiler kümesi modelin içinde önceden tanımlanmıştır. NEK Ontolojisinde nesnelere ve konumsal ilişkiler kullanılarak eylem tanımları yapılır. Konumsal ilişkilerin ayrıntıları bölüm 4.4'te bulunabilir.
- Olaylar (Events): Alandaki eylem ve kavramlara ilişkin olaylardır. Her bir olay bir zamanda ve yerde, bir nesne ile icra edilmiştir. Örneğin "voleybol" alanı için "maç" bir kavramsa; "Beşiktaş – Güneş Sigorta Maçı" kavramın bir olayı olmakta ve sırası ile örneklenen yer ve zaman niteliklerini içermektedir: "Selim Sırrı Tarcan Spor Salonunda", "23.12.2004". Ya da abdomen için "üst abdomen" bir kavramsa sırası ile örneklenen yer ve zaman niteliklerine sahip olabilir "Rıdvan Ege Hastanesi", "12.10. 2005". Olayın içinde kullanılan nesne, ontolojide tanımlı olmalıdır. Ayrıca olay bağlamında nesnelere rolleri de ele alınmıştır. Örneğin yine "voleybol" alanı için "smaç" eylemdir. "Smaç" eyleminin nesne ve konumsal ilişkilere göre yapılacak tanımında "insan", "top" ve "file" nesnelere kullanılacaktır. "Smaç" eyleminde "insan" nesnesi "smaçör" rolünü üstlenecektir. Dolayısı ile olay sınıfı kavram ya da eyleme ait bir olayı yer, zaman ve aktör bilgisi ile verirken aktöre ait rol tanımını da içerir.

Görüleceği üzere NEK modelinde alanın terimleri, üç sınıfa göre bölümlenir. Her üç sınıfın da birbiri ile ve kendi içinde ilişkileri bulunmaktadır. Nesne ve eylem

sınıflarının ilişkisi: eylemler nesnelere aracılığıyla tanımlanır. Ayrıca nesnelere, eylemlerin olaylarında kullanılır. Nesne ve kavram sınıflarının ilişkisi: nesnelere kavramların anlam kümesine ya da başka deyişle anlam bağlamına girer. Örneğin, “voleybol” kavramının anlamsal kümesinde “top”, “insan” gibi nesnelere ve “abdomen” kavramında “mide” ve “pankreas” gibi nesnelere yer alır. Bu eylemler ve diğer kavramlar için de geçerlidir. Yani “sindirim sistemi” nesne ve konumsal ilişkilerle tanımlanabilir için eylemdir ve “abdomen” kavramının bağlamında yer alırken, “abdomen” kavramı da “gövde” kavramının bağlamında yer alır. Ayrıca nesnelere kavramların olaylarını tanımlarken de kullanılır. Tüm bu ilişkiler ve sınıf tanımları Şekil 4.6’da görsel olarak ifade edilmiştir.



Şekil 4.6: NEK Ontoloji modeli

Şekil 4.6’da görüleceği üzere alanın ontolojiye girebilecek olan her terimi “Terim” sınıfı içinde yer alır. Bu terimler anlamsal özelliklerine göre 3’e ayrılır. Şekilde koyu çizgilerle ifade edilen bağıntı ve sınıflar bulanıktır. Aslında bağıntıların bulanık olması sınıfları da bulanıklaştırmıştır. İlerleyen kesimde Şekil 4.6’da görülen tüm

bağıntı ve sınıflar çizelge 4.1. ve çizelge 4.2'de açıklanmıştır. Daha sonra ontolojinin bulanıklık özelliği üzerinde durulmuştur.



Çizelge 4.1: NEK Ontoloji modeli sınıfları ve açıklamaları

Adı	Ata Sınıf	Tanımı
Terim	Yok	Alanda modellemeye girebilecek her türlü terim bu sınıfın olgusudur.
Nesne	Terim	Alanda yer alan somut nesne adlarıdır. Örneğin voleybol alanı için insan, file, top, saha vb. gibi ya da abdomen yapılar için mide, sağ böbrek, medulla spindis vb. gibi.
Eylem	Terim	Sadece tanımlı nesne ve konumsal ilişkiler ile tarif edilebilen eylemlerdir. Voleybol örneği için smaç eylem'dir ve insan, el, gövde, file, top nesnelere; yakın, içinde, üstünde konumsal ilişkileri ile ifade edilir. Gövde örneği için sindirim sistemi eylemi; mide gövdesi, mide bağlantısı, pankreas, karaciğer ve dalak nesnelere uygun konumsal ilişkileri ile ifade edilir.
EylemTanımı	Yok	Bir eylem birden çok biçimde ifade edilebilir. Bu gereksinim hem eşseslilik özelliği hem de aynı eylemin farklı tarif edilebildiği gerçeği ile ortaya çıkmıştır. Her bir eylemin tanımı bir EylemKuralı olgusu ile yapılır. Burada yapılan tanım bulanık bir tanımdır. Çünkü amaç eylemi tarif etmektir. Bu noktada birçok farklı tarif ve belirlilik düzeyi ortaya çıkar.
EylemKuralı	Yok	KuralParça'larından oluşan ve tüm parçalar bir araya geldiğinde ait olduğu EylemTanım'ını oluşturan kural kümesidir.

Çizelge 4.1: NEK Ontoloji modeli sınıfları ve açıklamaları (devam)

Adı	Ata Sınıf	Tanımı
AnlamlıBirliktelik	Yok	EylemKuralı'nı oluşturan <nesne, konumsal ilişki, nesne> üçlüsüdür. Yani belli konumsal özellikle bir araya gelen nesnelerin bir anlamı ifade etmesi söz konusudur. Örneğin <top, üstünde, file> üçlüsü "smaç" eylem tanımının bir kural parçasıdır. Tek başına yeterli değildir. Smaç eylemini çıkarsamak için gerekli olan nesne yerleşimlerinden sadece bir tanesidir. Ya da gövde alanında <pankreasgövdesi, üstünde, dalak> ifadesi, sindirim sistemi eylemini tarif ederken gereklidir ancak yeterli değildir.
Kavram	Terim	Alan içinde genelde soyut terimlere kaşılık gelmekle birlikte, anlamsal bağlamı olan terimlerdir. Eylem ve Nesne sınıflarından tamamen ayrıktır. Kavramın anlamsallığında diğer kavram, eylem ve nesnelere yer alabilir. Örneğin voleybol kavramının anlamsal genişliği, smaç eylemini, insan nesnesini ve maç kavramını kapsar.
Olay	Yok	Eylem ve Kavram sınıfı olgularına ait olaylardır. Yer ve zaman niteliği mevcuttur. Ayrıca olayın içinde yer alan nesne rolü ile tanımlanır.
Konumsal İlişki Şekil 4.6'da nitelik olarak gösterilir. Bu nitelik, değerini numarlandırılmış sınıftan alır.	Yok	EylemKural'larında kullanılabilinecek konumsal ilişkiler kümesidir. İki nesnenin birbirine göre konumsal ilişkisini ifade eder. Örneğin a nesnesi üstünde b nesnesi ifadesi b'nin referans alındığı durumda a'nın b üstünde olduğunu ifade eder. Kullanılabilinecek konumsal ilişkiler listesi: üstünde, üzerinde, altında, yanında, sağında, solunda, hizasında, yakın, uzak, içinde, kısmenİçinde

Çizelge 4.2: NEK Ontoloji modeli bağıntıları

Adı	Kaynak	Hedef	Tanım
eşAnlamlılık	Terim	Terim	Birbirinin eşanlamlısı olan iki terim arası ilişkidir.
nParça	Nesne	Nesne	Bir nesne diğerinin parçası (part-of) olabilir. Örneğin <i>gövde nesnesi</i> parçasıdır <i>insan nesnesinin</i> ya da <i>Lk</i> parçasıdır <i>sol böbrek nesnesinin</i> gibi.
birliktelik	Nesne	Nesne	Birlikte bulunmaları anlamlı olan iki nesne arasında, hangi konumsal ilişki ile birlikte bulduklarını gösteren nitelikli bağıntıdır. Birçoğa birçok özelliğinden dolayı bağıntı “AnlamlıBirliktelik” adlı bağıntı sınıfı ortaya çıkar.
eParça	Eylem	Eylem	Bir eylem diğerinin parçası olabilir. Örneğin “smaç” eyleminde yerden “sıçrama” eylemi parça eylemdir ya da “çift böbrek” eyleminin parçası “sol böbrek” eylemidir gibi.
öncüleSahip	Eylem	Eylem	Bir eylem diğerinin öncülüdür. Örneğin manşet eyleminin öncülü servis eylemi olabilir.

Çizelge 4.2: NEK Ontoloji modeli bağıntıları (devam)

Adı	Kaynak	Hedef	Tanım
tanımaSahip	Eylem	EylemTanımı	Eylemin adı ortak olmakla birlikte farklı tanımları hatta her tanımında farklı adı olabilir. Örneğin “servis” eylemi “smaç servis” ve “düz servis” olmak üzere iki ayrı ada ve tanıma sahiptir. Bir başka örnekle “karaciğer” sahip olduğu 9 kesimin, 6 tanesi ile bir tanıma sahip olurken 3 tanesi ile de bir başka tanıma sahip olabilir. Yani Eylem genel, EylemTanımı özel seviyeye karşılık gelir. Bu bağıntı ile aynı adlı eylem için birçok özel tanım geliştirilebilir.
kuralaSahip	EylemTanımı	EylemKuralı	Bu bağıntı, EylemTanımı ile özelleştirilen eylemi, çıkarsamasında kullanılacak kural kümesine bağlar.
eKuralParçası	Anlamlı Birliktelik	EylemKuralı	Eylem Kuralında yer alan her bir kural parçası, AnlamlıBirliktelik olgusudur. Olguyu kurala bağlar.
bağlamındaYer Alır	Kavram	Terim	Kavramın anlamsal bağlamında yer alan terimleri kavrama bağlayan bulanık ilişki.
kimAçıklaması	Olay	Nesne	Olayın içinde etken olan nesneyi gösteren bağıntıdır. Nesnenin rolü ya da açıklamasını nitelik olarak alır.
eOlayı	Eylem	Olay	Eylem’e ait zamanı, yeri belli olay
kOlayı	Kavram	Olay	Kavram’a ait zamanı yeri belli olay

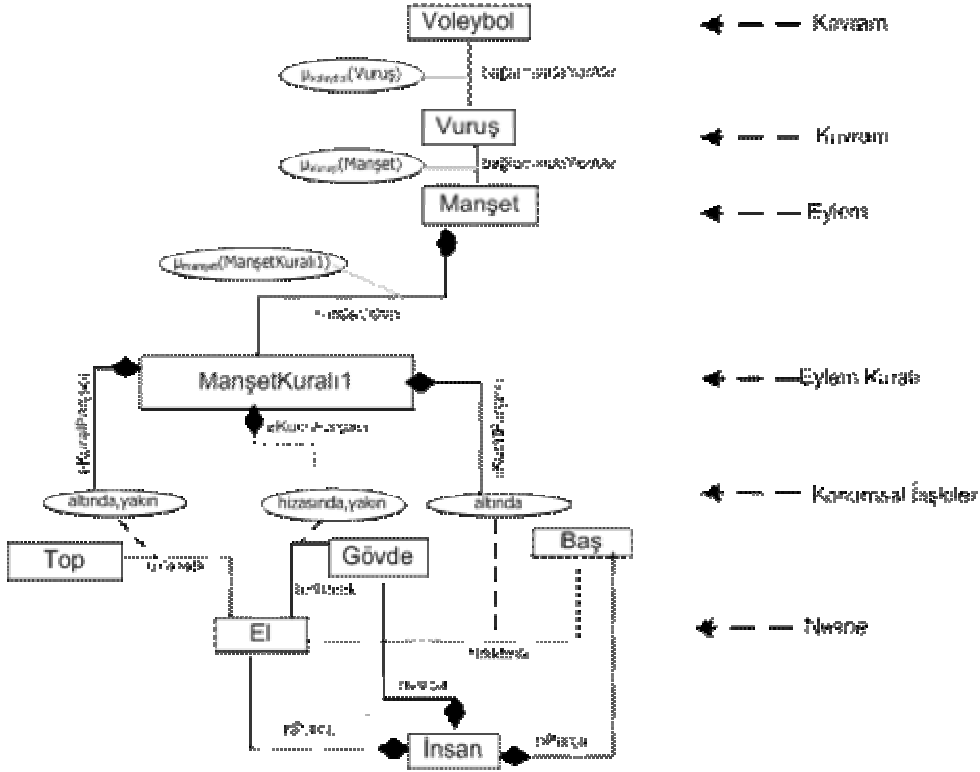
NEK Ontoloji modelinin biçimsel ifadesi şu biçimde oluşturulabilir: D modellenecek alan ve t alan terimi olmak üzere  $D = \{t_0, t_1, \dots, t_s\}$  biçimindedir. Alandaki nesnelere N, eylemler E ve kavramlar K ile gösterilirse  $N = \{t_0, t_1, \dots, t_n\}$ ,  $E = \{t_{n+1}, t_{n+2}, \dots, t_m\}$  ve  $K = \{t_{m+1}, t_{m+2}, \dots, t_s\}$  biçiminde gösterilebilir. Görüleceği üzere  $N \cap E = \emptyset$ ,  $N \cap K = \emptyset$  ve  $K \cap E = \emptyset$ 'dir. Ayrıca konumsal ilişkiler önceden tanımlıdır ve SR konumsal ilişkileri göstermek üzere  $SP = \{\text{içinde, dışında, kısmen içinde, üstünde, altında, sağında, solunda, sağ yanında, sol yanında, hizasında, üzerinde, uzak, yakın}\}$  değerlerini içermektedir.  $t_i$ 'nin nesne, eylem ya da kavram olması haline göre taşıyacağı nitelik ve ilişkilerin biçimsel ifadesi aşağıda verilmiştir.

$t_i \in N$  olmak üzere  $t_i = \{t_{i\text{adı}}, NP_i\}$ 'dir.  $NP_i$ , i. nesnenin parçalarında oluşan küme olmak üzere eğer  $t_j \in NP_i$  ise  $t_j \in N$  ve  $i \neq j$ 'dir.

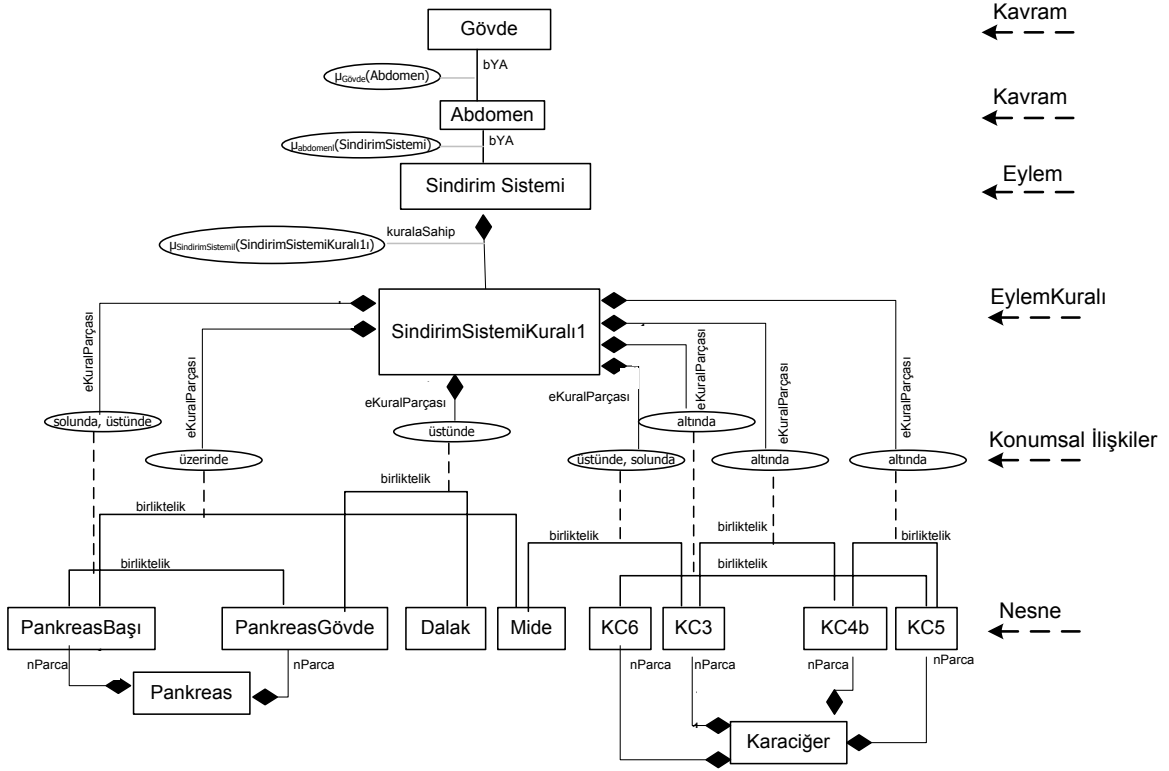
$t_i \in E$  olmak üzere  $t_i = \{t_{i\text{adı}}, ET_i, EP_i, E\ddot{O}_i, O_i\}$ 'dir.  $ET_i$ , i. eyleme ait tanımlar kümesi olmak üzere  $ET_i = \{ET_{i0}, ET_{i1} \dots ET_{im}\}$  biçiminde gösterilebilir. Ancak  $ET_i \neq \emptyset$ 'dir.  $ET_{im}$ , i. eyleme ait m. tanım ise  $ET_{im} = \{EK_{im}: \mu_{ET_{im}}(EK_{im})\}$  biçiminde gösterilebilir.  $EK_{im}$ , eylem tanımını yapan kurallar kümesidir ve her tanım sadece bir kural kümesi içerir.  $\mu_{ET_{im}}(EK_{im})$ , i. terimin m. tanımına ait eylem kuralının m. tanımını tarif edebilme derecesidir.  $EK_{im} = \{EK_{im0}, \dots, EK_{imt}\}$  biçimde kural parçaları ile ifade edilir ve  $EK_{im} = \emptyset$ 'dir.  $EK_{imt}$ , i. eylemin m. tanımında yer alan t. kural parçası olmak üzere  $EK_{imt} = \{t_h:r_k:t_y\}$  biçimde bir üçlüdür ve  $t_h \in N$ ,  $r_k \in SR$  ve  $t_y \in N$ 'dir.  $EP_i$ , i. eylemin parçalarından oluşan küme olmak üzere eğer  $t_j \in EP_i$  ise  $t_j \in E$  ve  $i \neq j$ 'dir.  $E\ddot{O}_i$ , i. eylemin öncül eylemlerinden oluşan küme olmak üzere  $t_j \in E\ddot{O}_i$  ise  $t_j \in E$  ve  $i \neq j$ 'dir.  $O_i$ , i. eyleme ait olay, Y olayın yeri, Z olayın zamanı ve  $t_n$  de olayın aktörü olmak üzere  $O_i = \{Y, Z, t_n:\text{rol adı}\}$  ve  $t_n \in N$ 'dir.

$t_i \in K$  olmak üzere  $t_i = \{t_{i\text{adı}}, KK_i, O_i\}$ 'dir.  $KK_i$ , i. kavramın bağlamında yer alan terimler kümesidir ve  $KK_i = \{t_0: \mu_{ti}(t_0), t_1: \mu_{ti}(t_1), \dots, t_n: \mu_{ti}(t_n)\}$  biçiminde ifade edilir. Bu ifadede  $t_n$ ,  $t_i$  kavramının bağlamında yer alan bir terimdir ve  $\mu_{ti}(t_n)$  ile bağlamda yer alma derecesi gösterilir. Ayrıca  $KK_i \neq \emptyset$  ve  $t_n \in D$ 'dir.  $O_i$ , i. kavrama ait olay, Y olayın yeri, Z olayın zamanı ve  $t_m$  de olayın aktörü olmak üzere  $O_i = \{Y, Z, t_m:\text{rol adı}\}$  ve  $t_m \in N$ 'dir.

NEK Ontoloji modeline uygun olarak voleybol alanının küçük bir kesiti şekil 4.7.a ve abdomen alanının küçük bir kesiti de şekil 4.7.b ile örneklenmiştir. Şekil 4.7.a ve şekil 4.7.b modelin bulanık bağıntı ve sınıfları için düzenlenmiştir. Çizelge 4.2’de gösterilen bağıntıların büyük kesimini örneklemeyiz. Ancak alandaki anlamsallığın nasıl nesne düzeyinden kavram düzeyine dek yükseltildiğinin küçük bir örneği olarak incelenmelidir.



Şekil 4.7.a: NEK Ontoloji modelinin bulanık sınıf ve bağıntıları: “voleybol”



Şekil 4.7.b: NEK Ontoloji modelinin bulanık sınıf ve bağıntıları: “abdomen”

NEK ontoloji modelinin amacı terimlerin ve görüntülerin anlamsal özelliklerini çıkarsamaktır. Aynı model her iki veri türüne de hizmet edecektir. Bu sebeple şekil 4.7.a ve b’de gösterilen kesitlerin her iki veri türü için anlamsal çıkarsamada ayrı ayrı nasıl kullanıldığı genel olarak ve NEK Ontoloji Modelinde yer alan üç temel sınıfın alandaki karşılıkları bu kesimde açıklanacaktır. Ayrıntılı bilgi bölüm 4.5’de yer almaktadır.

### Görüntüler için NEK Ontoloji Modelinin İşlevselliği:

Şekil 4.7.a’da “voleybol” alanı için NEK Ontoloji modelindeki nesne sınıfının olguları “top” ve “insan” nesnesinin parçaları olan “gövde”, “baş” ve “el”dir. Zira bu terimler görüntüde yer alan alanlarda karşımıza çıkabilirler. Yani alanın somut terimleridirler. “Voleybol” alanı ile ilgili bir görüntü için bu nesnelerin birlikte var olması elbette anlamlıdır. Ancak sadece var olan nesnelere bilmek görüntünün “smaç”, “servis”, “plase”, “manşet”, “maç”, “set”, “mola” gibi anlamsal özelliklerden hangisinin içerebildiğini belirleme noktasında yetersizdir. “Voleybol” alanındaki bazı terimlere yakından bakıldığında, nesne ve konumsal ilişkiler ile ifadeleri

mümkündür (örn. manşet, smaç, servis gibi). Çünkü bu terimlerin tarifi mümkün icra biçimleri vardır. Yani nesnelere ne tür bir yerleşimle varıldığı önem kazanır. Dolayısıyla ile NEK Ontoloji modelindeki Eylem sınıfının olguları olarak ele alınırlar. Manşet terimi de nesne ve konumsal ilişkiler ile tarif edilebilmesi sebebiyle, bir eylem olgusu olarak ele alınmış ve Şekil 4.7.a ile “manşet” eylemi gösterilen konumsal ilişkiler ve nesnelere ile tarif edilmiştir. Burada manşet eyleminin bulanıklığına dikkat çekmek gereklidir. Zira manşet eylemi bir tarife sahiptir. Ancak bu tarifte sözel değişkenler olan konumsal ilişkiler ve nesnelere kullanılmıştır. Yani tarifin kendisi bulanıktır. Dolayısıyla ile görüntülerden elde edilen nesne ve konumsal bilgilerin karşılaştırılacağı tarif bulanıktır denilebilir. Ayrıca görüntülerden çıkarılan konumsal ilişkiler de bulanıktır. Örneğin “üstünde” konumsal ilişkisi [0..1] aralığında bir değerle çıkarılacaktır. Dolayısıyla ile bulanık tarifin sınanacağı verilerde bulanık üretilmektedir. Sonuç olarak gerçekleşmesi gereken sınama da bulanık olacaktır.

Aynı biçimde Şekil 4.7.b’de “gövde” için örneklenen nesnelere “pankreas” ve parçaları, “karaciğer kesimleri”, “dalak” ve “mide” dir. Görüleceği üzere nesnelere organlara ve organ kesimlerine karşılık gelmektedir. Bu terimlerin nesne olmasına sebep olan faktör alanın somut terimleri olup CT görüntülerinde doğrudan yer alabilmeleridir. Sindirim sistemi bir eylemdir, çünkü abdomen alandaki nesnelere ve konumsal ilişkiler ile tarif edilebilir ve Şekil 4.7.b’de sindirim sistemi eylemine ait tanım görülmektedir. Ancak bu eylem bulanıktır. Çünkü yapılan tarif nesnelere ve sözel değişkenlerden oluşan konumsal ilişkileri içerir. Tanım bulanık olduğu gibi, bu eylemin çıkarılması da bulanıktır. Çünkü yapılan tanımın sağlanma derecesi söz konusudur. Yani yapılan tanım tam sağlandığında bile konumsal ilişkileri sağlama derecesi bulanık gelecektir. Ayrıca tanımın tam sağlanamaması da söz konusudur. Dolayısıyla ile her iki durumdaki belirsizliği ele almanın, ancak sindirim sisteminin bulanık bir eylem olarak işlenmesi ile mümkün olabileceği düşünülmektedir. Örneğin toraksın alt kısmına ait görüntülerde de sindirim sistemi eylemini tarif eden organlar mevcuttur. Fakat hem değişen konumsal ilişkiler hem de eklenen yeni organlar söz konusudur. Bu durumda alt toraksa ait görüntülerde de sindirim sistemi eylemi vardır ancak eylem tarifini sağladığı ölçüde ilgi derecesine sahip olacaktır. Yani uzmanlar için toraksın alt kesiminde başlayan “sindirim sistemi” eylemi önce artarak üst abdomende meydana gelmekte daha



sonra azalarak alt abdomende devam edip, sonlanmaktadır. Dolayısı ile uzmanın kendi doğalında görüntüyü ifade ederken kullanabileceği “toraks görüntüsü ama kısmen sindirim sistemi de var” ifadesi NEK Ontoloji Modelini kullanarak gerçekleştirilecek anlamsal erişimlerde desteklenmiştir.

Eylemi “tarif etmek” aynı zamanda tarifin netlik derecesi hususunu da gündeme getirir. Dolayısı ile tarifi taşıyan “EylemKuralı” türündeki olgulara örnek olarak şekil 4.7.a’da “ManşetKuralı1” ve şekil 4.7.b’de “SindirimSistemiKuralı1” olgularının da tarif edebilme derecesi vardır. Bu dereceler “ $\mu_{\text{manşet}}(\text{ManşetKuralı1})$ ” ve “ $\mu_{\text{SindirimSistemi}}(\text{SindirimSistemiKuralı1})$ ” ile gösterilmiştir. NEK modelinde bu tür tarif edebilme derecelerinin varsayılan değerleri 1’dir. Modelleyici “EylemKuralı”nda belirtilen tüm kuralların sağlanması halinde bile eylemin tam tarif edilemeyeceğini düşünür ise bu dereceyi [0..1] kapalı aralığında değiştirebilir. Çizelge 4.3.a’da manşet eylemini ve çizelge 4.3.b’de sindirim sistemi eylemini tarif eden kurallar bir çizelge ile gösterilmiştir:

Çizelge 4.3.a: “Manşet” eylemi EylemKuralı örneği

Nesne	Konumsal İlişki	Nesne
baş	üstünde	gövde
gövde	hizasında, yakın	el
Top	altında, yakın	el

Çizelge 4.3.b: “SindirimSistemi” eylemi EylemKuralı örneği

Nesne	Konumsal İlişki	Nesne
PankreasBaşı	solunda, üstünde	PankreasGövde
KC6	altında	KC5
KC4b	altında	KC3
KC5	altında	KC4b
Mide	üzerinde	PankreasBaşı
PankreasGövde	üstünde	Dalak
KC3	üstünde, solunda	Mide

Şekil 4.7.a'da “manşet” eylemi “vuruş” kavramına “bağlamındaYerAlır” bağıntısı ile bağlanmıştır. “Vuruş” bir eylem değildir. Çünkü “vuruş”, nesne ve konumsal ilişkiler ile tarif edilemez. “Vuruş” eylem türleri grubunun genel adıdır. Yani “voleybol”da “vuruş” türleri vardır. Bu türlerin her biri eylemdir. “Vuruş” kavramı tüm türleri anlamında barındıran bir kavramdır. “Manşet” te, “vuruş” türlerinden biri olduğu için ilgili bağıntıya sahiptir. “bağlamındaYerAlır” bağıntısı bulanıktır. Çünkü amaç kavramı, anlamında içerdiği terimler ile ilişkilendirmektir. Anlamında içerme tespiti ise ancak bulanık yapılabilir. Dolayısı ile  $\mu_{\text{vuruş}}(\text{manşet})$  ile “manşet” eyleminin “vuruş” kavramı tarafından ne kadar ifade edilebildiği; başka bir deyişle “vuruş” kavramının anlamı düşünüldüğünde “manşet”in bu anlama girebilme derecesi ifade edilir. Dolayısı ile vuruş kavramı, “bağlamındaYerAlır” bulanık ilişkisi sebebiyle bulanıktır. Yani vuruş kavramı ile ontolojide ilişkilendirilmiş olan diğer terimler, ilişkinin gücünü ifade eden derecelere sahiptir. Manşet eylemi ile vuruş kavramı arasındaki bağlamındaYerAlır ilişkisinin derecesi 1 bile olsa, manşet eyleminin görüntüden çıkarılması yukarıda açıklandığı üzere bulanıktır. Dolayısı ile bu durumda bile görüntüden çıkarılabilecek vuruş kavramı bulanıklaşacaktır.

Şekil 4.7.b'de yer alan “sindirim sistemi” eylemi “abdomen” kavramının bağlamında yer alır. Abdomen bir kavramdır. Yani nesnelere gibi somut değildir ve eylemler gibi nesne ve konumsal ilişkiler ile makul tarifi söz konusu değildir. Çünkü birçok farklı nesne ve konumsal ilişki barındırabilir. Aslında gövde üzerinde bir bölgeyi hedeflediği için somut ya da tarif edilebileceği gibi bir intiba uyandırabilir. Ama dikkat edilmelidir ki; abdomen bir bölgede yer alan birçok nesneden oluşur. Değişik anamolilere göre abdomen içinde değişik nesne ve konumsal ilişkiler söz konusu olur ve her biri başka anlamlara gelebilir. Bunların hepsi abdomen kavramına bağlıdır. Zira ait oldukları bölgedir. Şu durumda alt gövde üzerindeki tüm eylemler “abdomen” terimi ile ilişkilidir diye düşünmek, abdomenin anlamsal genişliğine daha uygun bir yaklaşımdır. Zira alan uzmanı yani doktorlarda bir görüntüye bakıp, kavramsal anlamını “üst abdomen ama kısmen toraksta var” biçiminde ifade edebilir. Yani görüntüler abdomen kavramını 0 ya da 1 derecesi ile değil, [0..1] aralık derecesi ile taşıyabilirler. Örneğin abdomen CT görüntüleri, gövdenin üst kesiminden başlayıp en alta kadar sıra ile çekilmiş kesit görüntüleridir. Dolayısı ile akciğerlerin olduğu kesim olan “toraks” görüntüleri, toraksın bitişi esnasında aslında gövdenin üst abdomen kısmını da içermeye

başlamıştır. Dolayısı ile görüntünün içinde yer alan üst abdomen ifadesinin, görüntüde yer aldığı kadarı ile çıkarılmasının gerekliliği düşünülmüştür. Dolayısı ile NEK Ontoloji modelinde abdomen, kavram sınıfının bir olgusudur. Bulanık eylem sindirim sisteminin, bulanık kavram olan abdomen ile “bağlamındaYerAlır” ilişkisi de bulanıktır ve  $\mu_{\text{Abdomen}}(\text{SindirimSistemi})$  ile derecelendirilir. Bu derece alan uzmanınca verilir. Sindirim Sistemi terimi ile abdomen arasında varolan bağlamındaYerAlır ilişkisinin derecesi 1 bile olsa görüntünün abdomen terimi ile ilişkisi yine bulanık kalacaktır. Zira Sindirim Sistemi de bulanık olup görüntüden çıkarsanma düzeyi  $[0..1]$  aralığında olacaktır.

Sonuç olarak eylemlerin bulanıklaşmasının temel sebebi, taşıdıkları tarifiñ sözel değişkenler kullanılarak bulanıklaşmasıdır. Ayrıca görüntüden çıkarsanacak verilerinde bulanık olduğu unutulmamalıdır. Kavramların bulanıklaşmasının temel sebebi taşıdıkları bağlamındaYerAlır ilişkisinin doğal yaşamda da bulanık olmasıdır. Ayrıca görüntülerden çıkarsanacak eylemlerin bulanıklığı, eylemlerin bağlamına girdikleri kavramları da bulanıklaştırır. Özetle gerek eylem ve gerekse kavram sınıflarının bulanıklaşması için iki temel faktör oluşmaktadır: taşıdıkları tanım ve ilişkilerin doğal yaşam da da bulanık olması ve görüntülerden çıkarılan verinin bulanık üretimesi.

Görüntüler için, anlamsal çıkarımda, görüntülerin taşıdığı nesne ve konumsal ilişkiler bulanık üretilir. Üretilen veri, NEK Ontoloji Modelinde tanımlı “EylemKurallar”ı ile sınanır ve görüntünün ifade ettiği eylemler bulanık çıkarsanır. Zira görüntüden çıkarılan içerik özellikleri ve sınamada kullanılan bağıntı bulanıktır. Daha sonra görüntünün içerdiği nesne ve eylemler, ontolojide tanımlı “bağlamındaYerAlır” bağıntısı ile ilişkilendirilmiş terimler için sınanır. Bu sınama da gerek bağıntının doğası ve gerekse görüntüden çıkarılan eylemlerin bulanık olması sebebiyle bulanıktır.

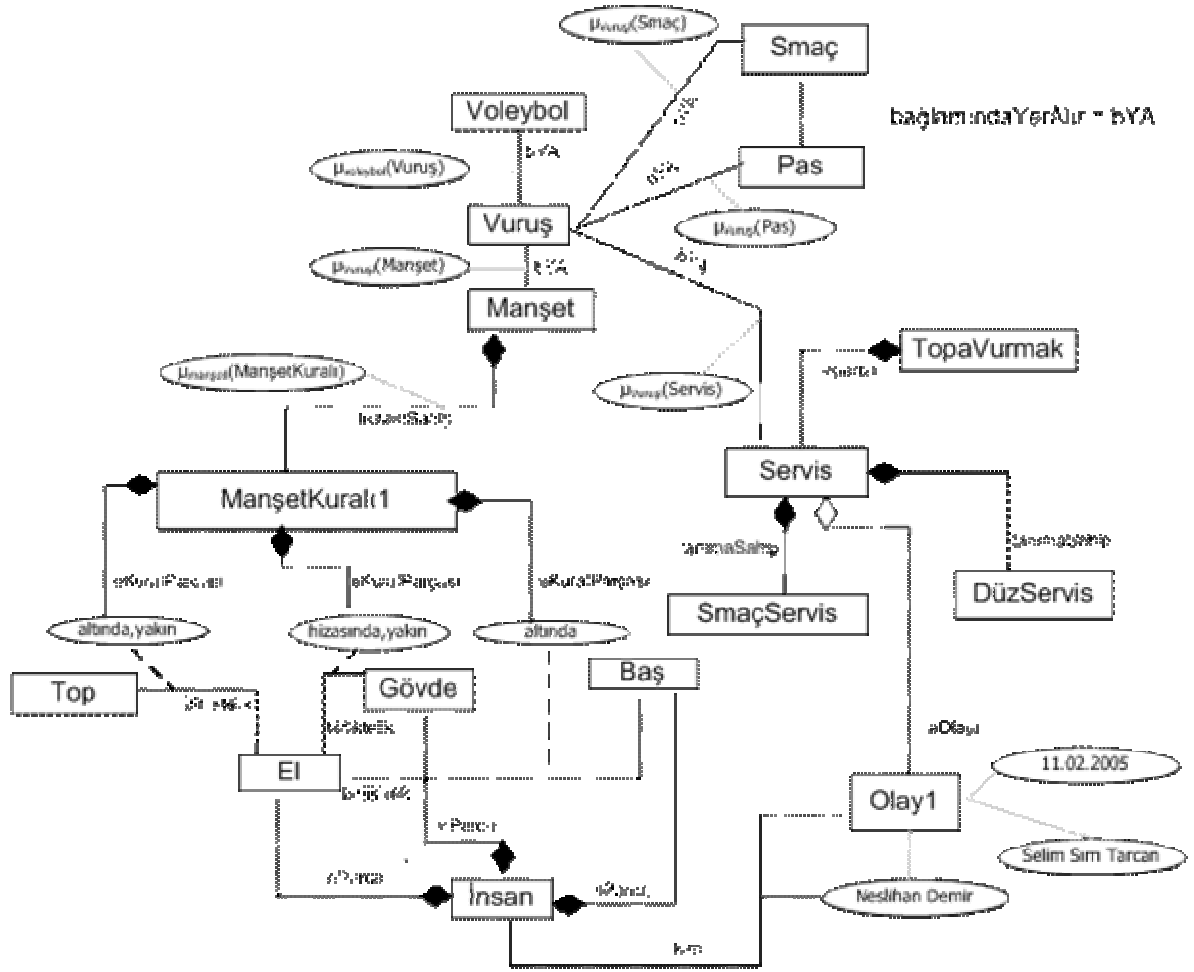
### **Metinler için NEK Ontoloji Modelinin İşlevselliği:**

NEK Ontoloji Modelinin metinler için kullanımı şu biçimdedir: Eylemler “EylemKuralı”ndaki nesnelere ile anlamsal ilişkilidir. Doğrudan o nesnelere ile tarif edilebilmektedir. Dolayısı ile her eylem kendisi için anlamlı nesne terimlerini bir kümede taşımaktadır denilebilir. Aynı şekilde her kavram da “bağlamındaYerAlır”

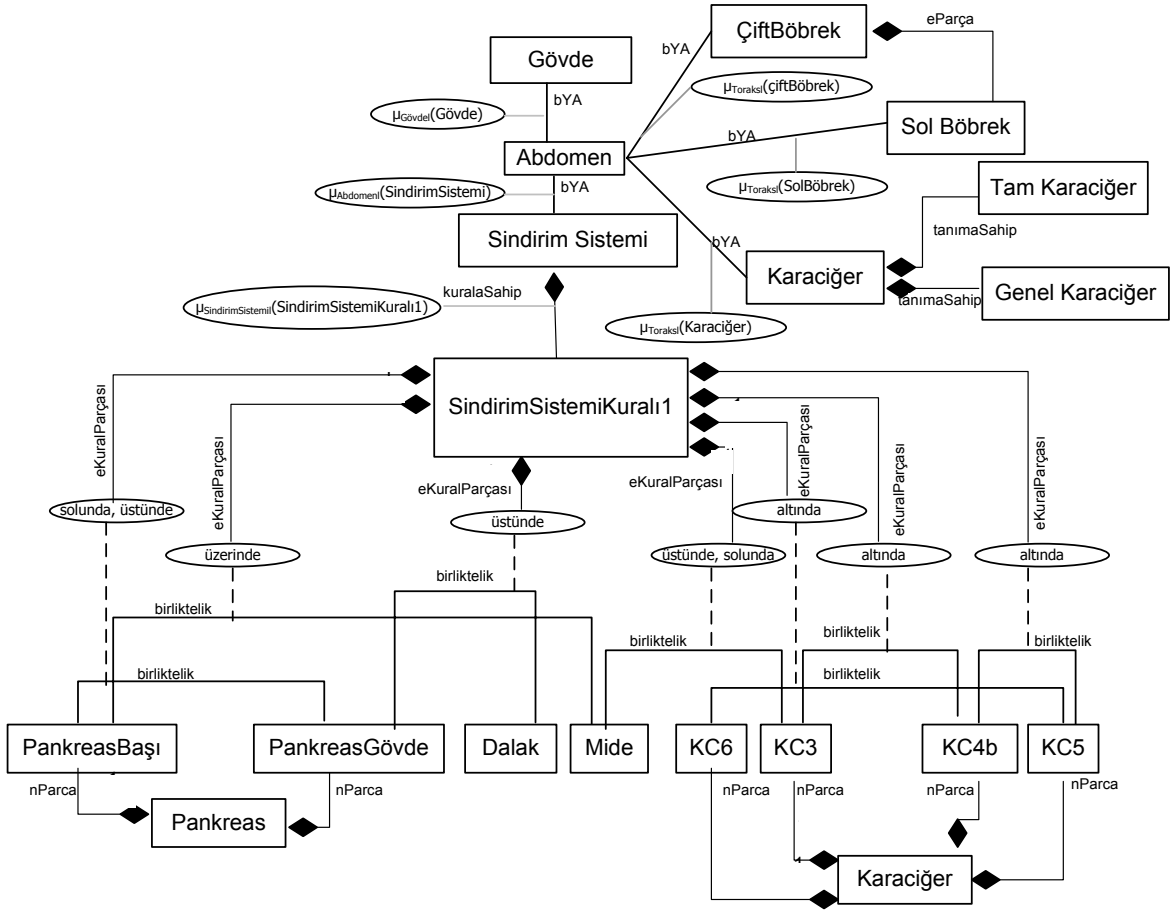
bağıntısı sebebi ile kendi bağlam kümesine sahiptir. Dolayısı ile metni oluşturan terimler için yayılmış ağırlık (propogated weights) yaklaşımının kullanılacağı uygun bir ortam söz konusudur. Yani “manşet” teriminin metinde varlığı “vuruş” teriminin ağırlığını, “sindirim sistemi” teriminin varlığı “abdomen” teriminin ağırlığını olumlu yönde etkiler. Aynı şekilde “insan” teriminin varlığı da “el”, “gövde” ve “baş” terimlerinin ağırlığını, “PankreasBaşı” ve “PankreasGövdesi” terimlerinin varlığı da “Pankreas” terimin ağırlığını olumlu yönde etkiler.

Şekil 4.7.a ve şekil 4.7.b üzerinde buraya kadar yapılan açıklamalar, NEK Ontoloji modelinin görüntü ve metne ait anlamsal çıkarsamalara nasıl hizmet edebileceği ile ilgili genel çerçeveyi oluşturmaktadır.

NEK Ontoloji modelinde var olan tüm bağıntıların yer aldığı ontoloji kesitleri şekil 4.8.a’da “voleybol” ve şekil 4.8.b’de “gövde” alanlarının kesitleri ile örneklenmiştir. Şekil 4.8, şekil 4.7’ye eklentileri içermektedir. Yani var olan bulanık omurganın üzerine Olay sınıfı ve bağıntıları ile yeni eylem ve kavramları içermektedir.



Şekil 4.8.a : Genişletilmiş “voleybol” ontoloji kesiti



Şekil 4.8.b : Genişletilmiş "abdomen" ontoloji kesiti

Şekil 4.8.a'da vuruş kavramı bağlamında yer alan manşet, smaç ve servis eylemleri görülmektedir. Servis eylemi iki ayrı tanım sahiptir. Çünkü iki ayrı vuruş tekniğine göre değişen konumsal ilişkiler, tarifleri farklılaştırmıştır. Ayrıca şekil 4.8.b'de ise abdomen bağlamında yer alan çift böbrek, karaciğer ve sindirim sistemi eylemleri görülmektedir. Karaciğer eyleminde, organ kesimlerinin görüntülerde yer alma farklılıklarına göre iki farklı tanım yapılmıştır. Sonuç olarak bir eylemin farklı biçimlerde aynı ya da ayrı adla tarif edilebilmesi gereksinimi söz konusudur ve NEK Ontoloji Modelinde desteklenmiştir. Ayrıca şekil 4.8.a ve b'de görüleceği üzere eylemler başka eylemler ile parça bütün ilişkisine sahip olabilmektedir.

Sonuç olarak NEK ontoloji modeli, ontoloji geliştirirken kullanılacak bir yaklaşımı, tanımladığı sınıf ve bağıntılar ile önermektedir. Modelleyiciye yol göstermektedir.

Bu yaklaşımın tasarımındaki en önemli iki etken yaklaşımı alandan bağımsız kılmak ve aynı model ile hem metin hem de görüntülere hizmet verebilmektir. Modelin bulanıklaşmasına sebep olan iki etken şu biçimde özetlenebilir:

- Hizmet edilecek veri türlerinden biri görüntüdür. Görüntüden otomatik çıkarsanacak anlamın bulanık olmasının türün doğasının bir gerekliliği olduğu düşünülmektedir.
- Ontoloji modelinde tarif etmek ve bağlamsal bağıntı kurmak gibi iki ifade biçimi bulunmaktadır. Her iki ifade doğal yaşamda da bulanıktır.

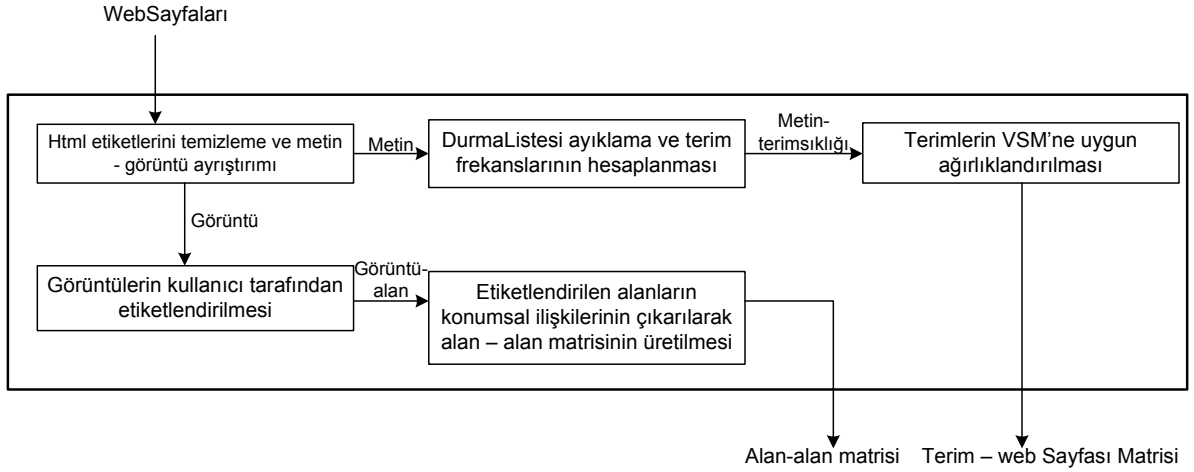
Bu kesimde açıklanan tüm bağıntı ve sınıf tanımlarını kullanarak ontoloji üretiminde kullanılan Ontoloji Editörü (OE) Aracının arayüzleri Ek-1'de ve OE'nin kullandığı sınıf ve bağıntı tanımlarının OWL ile gösterimi Ek-2'de yer almaktadır.

İlerleyen bölümde alan ontolojisi ile ilişkilendirilmek üzere görüntü ve metinlerin içerik özelliklerinin çıkarılması açıklanmıştır.

#### **4.4. Web Sayfasının İçerik Özelliklerinin Çıkarımı**

Bu kesimde tüm sayfanın içerik özelliklerinin nasıl çıkarıldığı anlatılacak, içerik özellikleri ile alan ontolojisinin ilişkilendirilmesi bölüm 4.5'de tartışılacaktır.

Sistemin girdisi, içinde görüntü ve/veya metin içeren HTML biçiminde web sayfalarıdır. Her bir sayfa, veri türlerine ayrıştırılır ve ayrı ayrı içerik özellikleri çıkarılır. Bu sebeple metin ve görüntülerin içerik özellikleri çıkarımı ayrı ayrı açıklanacaktır. Ancak işlemin ana hatları ile akışı Şekil 4.9'da gösterilmiştir. Şekil 4.9 AEA aracının, Şekil 4.1'de gösterilen, "Üst veri çıkarma modülü"nü ayrıntılı görünümüdür.



Şekil 4.9: Web Sayfasının içerik özelliklerini çıkarma aşamaları

Şekil 4.9’da her bir kutu bir işleme karşılık gelmektedir. Bu işlemlerin ayrıntıları aşağıda maddelenerek açıklanmıştır.

**a. HTML etiketlerini temizleme ve metin-görüntü ayrıştırımı:** Bu işlem adımında öncelikle HTML’e ait sayfanın gösterimini belirleyen tüm etiketler (tags) ayıklanmıştır. Ayıklama esnasında, sayfada terim olarak kullanılabilinecek her türlü bilgi korunmuştur. <img> ve </img> Etiketleri arasında kalan adres ve ad bilgileri ile görüntülere erişilmiştir. İşlem adımının çıktısı, terimleri ve görüntü adreslerini içeren iki adet listedir.

**b. Durma listesi (stop list) ayıklama ve terim sıklıklarının hesaplanması:** Bu işlem adımında “a.” işlem adımının çıktısı olan terim listesi kullanılarak, terimlerin web sayfasına özel sıklıkları sayılmıştır. Öncelikle, Türkçe dilinin durma listesi terim listesinden çıkarılmıştır. Kullanılan durma listesi 1995 yılında Ege Üniversitesi’nde yapılan “Turkish Stop List” adlı çalışmanın sonucunda üretilen listedir[152]. Bu liste halen Türkçe arama motorları için kullanılmaktadır. Sıklık saymada her bir terim gövdesi ile ele alınmıştır. Türkçe terimlerin gövdelenmesi için [156]’daki gövdeleme algoritması kullanılmıştır. Bu algoritma ile terimler çekim eklerinden ayıklanmış ancak yapım ekleri korunmuştur. Bu sebeple elde edilen terimlerin kök değil gövde olduğu söylenir. Bu işlem adımının girdisi her bir sayfanın terim listesi, çıktısı ise her bir sayfa için terim gövdeleri ve sıklıklarıdır.

**c. Terimlerin Vektör Uzayı Modeline (Vector Space Model–VSM)’ uygun ağırlıklandırılması:** Bu işlem adımından önce derlemdeki tüm web sayfaları için



“b.” işlem adımının tamamlanması gerekmektedir. Çünkü bir terimin VSM’ne uygun ağırlıklandırılabilmesi için, terimin sayfa ve derlem sıklıklarının tespit edilmesi gerekir. Dolayısı ile “c.” adım tüm derlem için uygulanır. VSM’de kullanılan ağırlıklandırma formülü şu biçimdedir:

$$W_{im} = t_{im} * (\log N/d_m)$$

Formülde kullanılan  $W_{im}$ , i. sayfada m. terimin ağırlığını;  $t_{im}$ , i. sayfada m. terimin geçme sıklığını; N, derlemdeki web sayfası sayısını ve  $d_m$ , m. terimin geçtiği web sayfası sayısını göstermektedir. Dolayısı ile bir terim geçtiği web sayfasında ne kadar çok ve derlemin web sayfalarında ne kadar az geçiyor ise o sayfa için ağırlığı da o oranda yüksek olacaktır.

Sonuç olarak “c.” işlem adımının tüm metinlere ait terim sıklık listelerini girdi olarak alıp, ağırlıklandırıp, web sayfası-terim matrisini ürettiği söylenebilir. Bu matrisin görünümü çizelge 4.4’te örneklenmiştir.

Çizelge 4.4: Örnek web sayfası – terim vektörü

	WebSayfası <sub>1</sub>	WebSayfası <sub>2</sub>	WebSayfası <sub>3</sub>
Terim <sub>1</sub>	0.33	0	0.99
Terim <sub>2</sub>	0	0	0.67
Terim <sub>3</sub>	0	0.78	0.89

Çizelge 4.4’te örneklenen ağırlıklar, derlemden istatistiksel yöntemle elde edilen ağırlıklardır. Henüz alan ontolojisi ile ilişkilendirilmemiştir. Bu sebeple ham ağırlık olarak adlandırılır. Asıl amaç, bu ham ağırlıkları alan ontolojisindeki bağıntılara göre yeniden hesaplayarak, sayfanın anlamsal özelliği olabilecek terimlerin ağırlıklarını öne çıkarmaktır. Görüleceği üzere “b” ve “c” maddeleri metinler için uygulanan iki işlem adımıdır. İlerleyen maddeler görüntüler için var olan işlem adımlarını içermektedir.

**d. Görüntülerin kullanıcı tarafından etiketlenilmesi:** Bu işlem adımında girdi olarak “a.” işlemin çıktısı olan görüntü adresleri listesi kullanılır. Her bir web sayfasının görüntü listesinden kullanıcının seçtiği görüntüler kullanıcı seçimine göre görüntülenir. Kullanıcı bu arayüzdeki görüntü üzerinde, etiketlendireceği alanı

işaretler ve adını yazar. İşaretlenen bölgenin koordinatları görüntüye göreli biçimde hesaplanır ve kayıt edilir. Bu işlem adımının çıktısında etiketlenilmiş her bir görüntünün, etiketlenilmiş alan adı ve koordinatlarını içeren bir liste üretilir. Bu işlem adımında kullanıcının etiketlendirme yapabileceği AEA aracının arayüzü Ek-3'te yer almaktadır.

**e. Etiketlendirilen alanların konumsal ilişkilerinin çıkarılması ve alan-alan matrisinin üretilmesi:** Bu işlem adımı "d." işlem adımının çıktısı olan listeyi kullanarak, her bir görüntü için alan-alan vektörünü üretir. Alan-alan vektörü satır ve sütunlarında alanları ve hücrelerinde satır ve sütunundaki alanlar arasındaki konumsal ilişkileri içeren bir vektördür. Vektörün genel görünümü Çizelge 4.5'te verilmiştir.

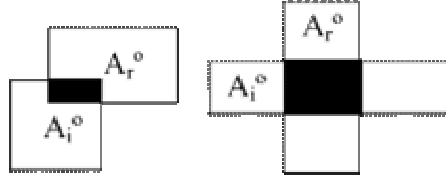
Çizelge 4.5: Alan-alan vektörü görünümü

	Alan <sub>1</sub>	Alan <sub>2</sub>
Alan <sub>1</sub>	—————	{ $\mu_{\text{topolojik}}(\text{alan}_1, \text{alan}_2)$ , $\mu_{\text{yön}}(\text{alan}_1, \text{alan}_2)$ , ( $\mu_{\text{uzaklık}}(\text{alan}_1, \text{alan}_2)$ )}
Alan <sub>2</sub>	{ $\mu_{\text{topolojik}}(\text{alan}_2, \text{alan}_1)$ , $\mu_{\text{yön}}(\text{alan}_2, \text{alan}_1)$ , ( $\mu_{\text{uzaklık}}(\text{alan}_2, \text{alan}_1)$ )}	—————

Alan-alan matrisinde görüleceği üzere her bir hücre üç tür konumsal ilişki değerine sahiptir [153]: topolojik ilişki (içinde, kısmen içinde, dışında), yön ilişkisi (sağında, solunda, üstünde, altında) ve uzaklık ilişkisi (uzak, yakın). Bu üç tür ilişkinin her biri için üyelik derecesi olan  $\mu$ 'yü belirleme biçimi aşağıda açıklanmıştır.

Bir G görüntüsünün m adet A ile gösterilen alanı olduğu kabul edilirse, görüntü  $G = \{A_0 \dots A_m\}$  biçiminde ifade edilebilir. Her bir A alanı için alanın üst sağ, orta ve sol alt noktaları tutulmaktadır. Bu sebeple i. A<sub>i</sub> alanı  $A_i = \{i, \text{alanAdı}, P_{\text{sağÜst}}, P_{\text{orta}}, P_{\text{solAlt}}\}$  biçiminde ifade edilir ve P nokta olmak üzere  $P = \{x, y\}$ 'dir. Ayrıca A<sub>i</sub> alanı için alanın içinde kalan bölgeyi  $A_i^\circ$  ve alanı çerçeveleyen kenarları da  $A_i^\delta$  gösterebilir. İki alan arasındaki üç tür konumsal ilişki derecesi  $\mu_{\text{topolojik}}$ ,  $\mu_{\text{yön}}$ , ve  $\mu_{\text{uzaklık}}$  değerlerinin hesaplanmasında bu gösterimler kullanılacaktır.

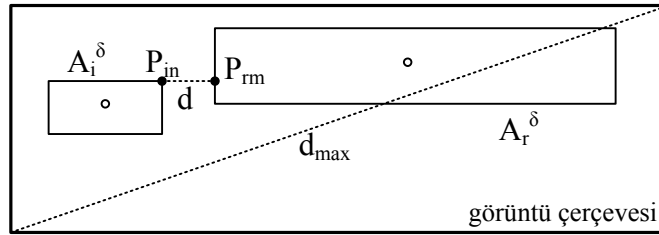
Topolojik ilişkiler; içinde, kısmen içinde ve dışında olmak üzere üç türdür. Şekil 4.10'da birbiri ile farklı konumlanışa sahip A<sub>i</sub> ve A<sub>r</sub> alanları görülmektedir.



Şekil 4.10:  $A_i$  ve  $A_r$  alanları konumlanış örnekleri: Topolojik

Şekil 4.10'da gösterilen  $A_i$  ve  $A_r$  alanları aynı görüntünün iki alanı olmak üzere ve  $A_i^o$  ve  $A_r^o$  alanların içini ifade ederse  $\mu_{\text{topolojik}}(A_i, A_r) = (A_i^o \cap A_r^o) / A_r^o$  'dir[153]. Formülden de anlaşılacağı üzere eğer  $\mu_{\text{topolojik}}(A_i, A_r) = 1$  ise  $A_r$  alanı  $A_i$  alanının tamamen içindedir. Eğer  $\mu_{\text{topolojik}}(A_i, A_r) = 0$  ise  $A_r$  alanı  $A_i$  alanının tamamen dışındadır. Eğer  $0 < \mu_{\text{topolojik}}(A_i, A_r) < 1$  ise  $A_r$  alanı  $A_i$ 'nin kısmen içindedir.

Uzaklık ilişkisi uzak ve yakın olmak üzere iki türdür. Uzaklık ilişkisi, topolojik ilişki ile birlikte ele alınmıştır. Şöyle ki;  $A_r$  ile  $A_i$  alanları arasındaki  $\mu_{\text{topolojik}}(A_i, A_r) > 0$  ise bu iki alan kısmen içinde ya da içinde ilişkisine sahiptir. Dolayısı ile uzak olamazlar. Yani  $\mu_{\text{uzaklık}}(A_i, A_r) = 0$  biçiminde düşünülmüştür. Dolayısı ile aralarında dışında ilişkisi olan  $A_i$  ve  $A_r$  alanları için uzaklık araştırması yapılmıştır. Uzaklık ilişkisinin şekillerin büyüklüğünden etkilendiği incelenen örnek görüntülerde gözlemlenmiştir. Bu sebeple varolan orta noktaların uzaklığı yaklaşımının[153] yerine kenarlar üzerinde bir birine en yakın iki noktanın uzaklığı kullanılmıştır. Şekil 4.11'de aynı görüntünün  $A_i$  ve  $A_r$  alanlarının konumlanışı uzaklık ilişkisi için örneklenmiştir.



Şekil 4.11:  $A_i$  ve  $A_r$  alanları konumlanış örnekleri: Uzaklık

Şekil 4.11'de görülen  $A_i$  ve  $A_r$  alanları birbirinden çok uzakta olmasa bile  $A_r$ 'nin yatayda uzun yerleşimi, alanın orta noktasının  $A_i$ 'den uzaklaşmasına sebep olmuştur. Yani orta noktaların uzaklığına asıl sebep  $A_r$  alanının yerleşimi ve büyüklüğüdür. Bu sebeple uzaklık ilişkisi iki alanın kenarları üzerinde yer alan en

yakın iki noktanın uzaklığı ile ölçülmüştür. Şekil 4.11’de yer alan “d” simgesi  $A_i$  ve  $A_r$ ’nin birbirine en yakın iki noktasının uzaklığını ve “ $d_{max}$ ” simgesi görüntüde birbirine en uzak olabilecek iki noktanın uzaklığını göstermektedir. Bu bilgiler ışığında  $A_i$  ve  $A_r$  alanlarının uzaklık derecesi  $\mu_{uzaklık}(A_i, A_r) = 1 - \mu_{yakınlık}(A_i, A_r)$ ’dir. Yakınlık derecesi olan  $\mu_{yakınlık}(A_i, A_r)$  içinse hesaplamalar şu biçimde gösterilir:

$$\mu_{yakınlık}(A_i, A_r) = \begin{cases} 1 & \mu_{topolojik}(A_i, A_r) > 0 \\ \max(\mu_{yakınlık}(P_{in}, P_{rm}) \mid P_{in} \in A_i^{\delta}, P_{rm} \in A_r^{\delta}) & \mu_{topolojik}(A_i, A_r) = 0 \end{cases}$$

Yön ilişkisi üstünde, altında, sağında ve solunda olmak üzere dört türdür. Dolayısıyla hesaplanması gereken  $\mu_{sağ}$ ,  $\mu_{sol}$ ,  $\mu_{alt}$  ve  $\mu_{üst}$  değerleri sözkonusudur. Bu ilişkilerin çıkarımında da orta noktaların oldukça sık kullanıldığı gözlemlenmiştir[153,154,155]. Yani  $A_i$  ve  $A_r$  aynı görüntünün iki alanı olmak üzere yön ilişkileri hesaplanırken şekillerin orta noktalarından hareket edilerek,  $A_i$  alanının orta noktası (0,0) orijini olarak kabul edilmiş ve  $A_r$ ’nin orta noktasının +x eksenini ile yaptığı açının sinüsü kullanılmıştır. Bir başka deyişle,  $A_i$  ve  $A_r$  alanlarının orta noktaları sırası ile  $P_{iorta}$  ve  $P_{rorta}$  ise,  $P_{iorta}$  (0,0) noktası olmak üzere, merkezden başlayıp  $P_{rorta}$ ’dan geçen doğru ile +x ekseninin açısı  $\Phi$  olmak üzere  $\mu_{sağ}(A_i, A_r)$ ,  $\mu_{sol}(A_i, A_r)$ ,  $\mu_{alt}(A_i, A_r)$  ve  $\mu_{üst}(A_i, A_r)$  değerlerinin hesaplanması şu biçimdedir:

Eğer ( $0 < \Phi < 90$ ) ve ya ( $270 < \Phi < 360$ ) ise  $\mu_{sağ}(A_i, A_r) = \sin(\Phi + 90)$ ’dir.

Diğer  $\Phi$  değerleri için  $\mu_{sağ}(A_i, A_r) = 0$ ’dir.

Eğer ( $90 < \Phi < 270$ ) ise  $\mu_{sol}(A_i, A_r) = \sin(\Phi - 90)$ ’dir.

Diğer  $\Phi$  değerleri için  $\mu_{sol}(A_i, A_r) = 0$ ’dir

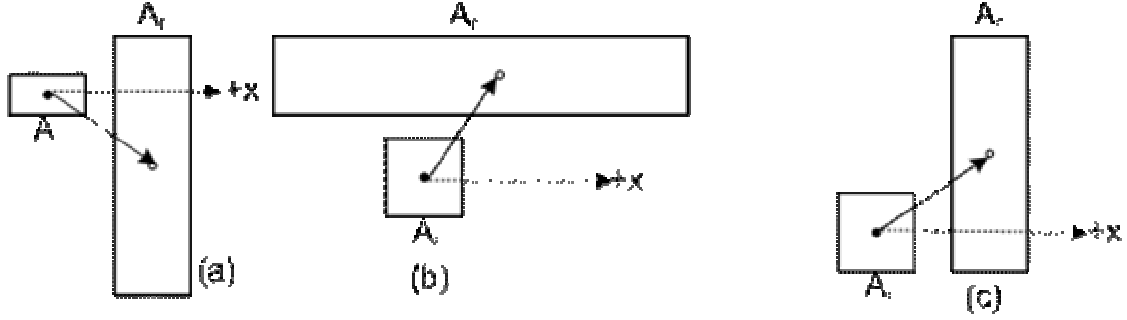
Eğer ( $0 < \Phi < 180$ ) ise  $\mu_{üst}(A_i, A_r) = \sin(\Phi)$ ’dir.

Diğer  $\Phi$  değerleri için  $\mu_{üst}(A_i, A_r) = 0$ ’dir

Eğer ( $180 < \Phi < 360$ ) ise  $\mu_{alt}(A_i, A_r) = \sin(\Phi - 180)$ ’dir.

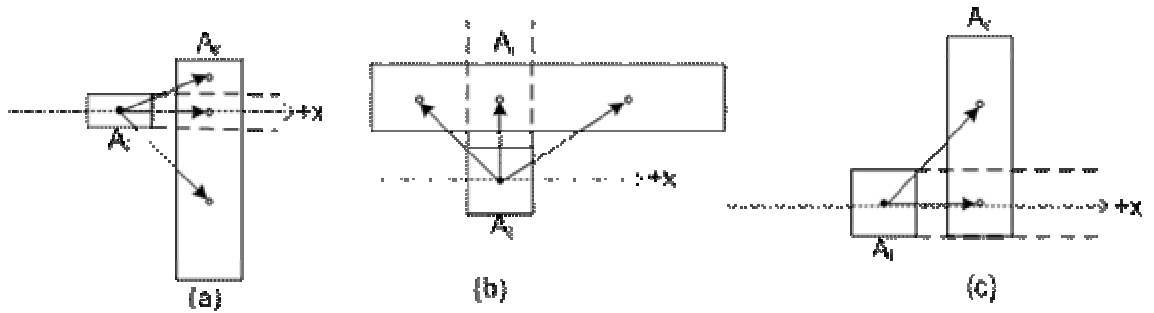
Diğer  $\Phi$  değerleri için  $\mu_{alt}(A_i, A_r) = 0$ ’dir.

Şekil 4.12 ile aynı görüntünün  $A_i$  ve  $A_r$  alanları için örnek yerleşimler ve orta nokta yaklaşımı uygulanması gösterilmiştir.



Şekil 4.12:  $A_i$  ve  $A_r$  alanları konumlanış örnekleri: Yön (orta nokta)

Şekil 4.12'de yer alan  $A_i$  ve  $A_r$  alanlarının yön ilişkisi için sadece orta noktalar kullanıldığında çeşitli yorum farklılıkları oluşması söz konusudur. Örneğin "b" şikkında sadece orta noktalara bakıldığında  $A_r$  alanı  $A_i$  alanının tam üstünde değil sağ ve üstünde çıkacaktır. Ancak  $A_r$  alanı,  $A_i$  alanının tüm üst bölgesinde boylu boyunca uzanmaktadır. Aynı biçimde "c" ve "a" şıklarında da  $A_r$  alanı,  $A_i$  alanının sağ tarafında boylu boyunca uzanmaktadır. Ancak şekillerin uzun boyu aynı zamanda orta noktalarının aşağıya kaymasına ve sağ ilişkisinin düşük derece ile hesaplanmasına sebep olmaktadır. Bu sebeple yön ilişkisi çıkarırken kaynak alana göre hedef alan bölümlere ayrılmış. Kaynak alanının orta noktasından hedef alanın bölümlerinin orta noktaları arasındaki yön ilişkileri araştırılmıştır. Şekil 4.13'te  $A_i$  kaynak alan, ve  $A_r$  hedef alan olmak üzere  $A_r$ 'nin  $A_i$ 'ye göre bölümlenmesi örneklenmiştir.



Şekil 4.13:  $A_i$  ve  $A_r$  alanları konumlanış örnekleri: Yön (bölümleme)

Şekil 4.13'te  $A_i$  kaynak alanına göre  $A_r$  hedef alanı bölümlenmiştir. Her bir bölümün orta noktasına kaynak alanın orta noktasından bakılmıştır. Yani iki orta nokta arasında kullanılan trigonometrik hesaplamalar, kaynak alanın orta noktası ile

hedef alanının her bir bölümünün orta noktası arasında yapılmıştır. Dolayısı ile hedef alana ait her bir bölümün orta noktası ile kaynak alanın orta noktası arasındaki  $\mu_{sağ}$ ,  $\mu_{sol}$ ,  $\mu_{alt}$  ve  $\mu_{üst}$  değerleri ortaya çıkar. Bu değerlerin seçiminde kullanılan yaklaşım şu biçimdedir:  $A_i$  kaynak alanın orta noktası  $P_{i_orta}$ ,  $n$  adet bölüme ayrılan  $A_r$  alanın  $m$ . bölümünün orta noktası  $P_{rm_orta}$  ise

$$\mu_{sağ}(A_i, A_r) = \max(\mu_{sağ}(P_{i_orta}, P_{rm_orta}) | 0 \leq m < n)$$

$$\mu_{sol}(A_i, A_r) = \max(\mu_{sol}(P_{i_orta}, P_{rm_orta}) | 0 \leq m < n)$$

$$\mu_{üst}(A_i, A_r) = \max(\mu_{üst}(P_{i_orta}, P_{rm_orta}) | 0 \leq m < n)$$

$$\mu_{alt}(A_i, A_r) = \max(\mu_{alt}(P_{i_orta}, P_{rm_orta}) | 0 \leq m < n)$$

Görüleceği üzere alt, üst, sağ, sol ilişkileri için bölümlerden elde edilen en yüksek dereceler kaynak ve hedef alanın alt, üst, sağ ve sol ilişkilerinin dereceleri olarak kullanılmıştır. Bu yaklaşım ile alanların uzunluk ve genişlik farklılıklarından dolayı orta noktalarının kayması halinde ortaya çıkan dezavantajlı durumun azaltılması amaçlanmıştır.

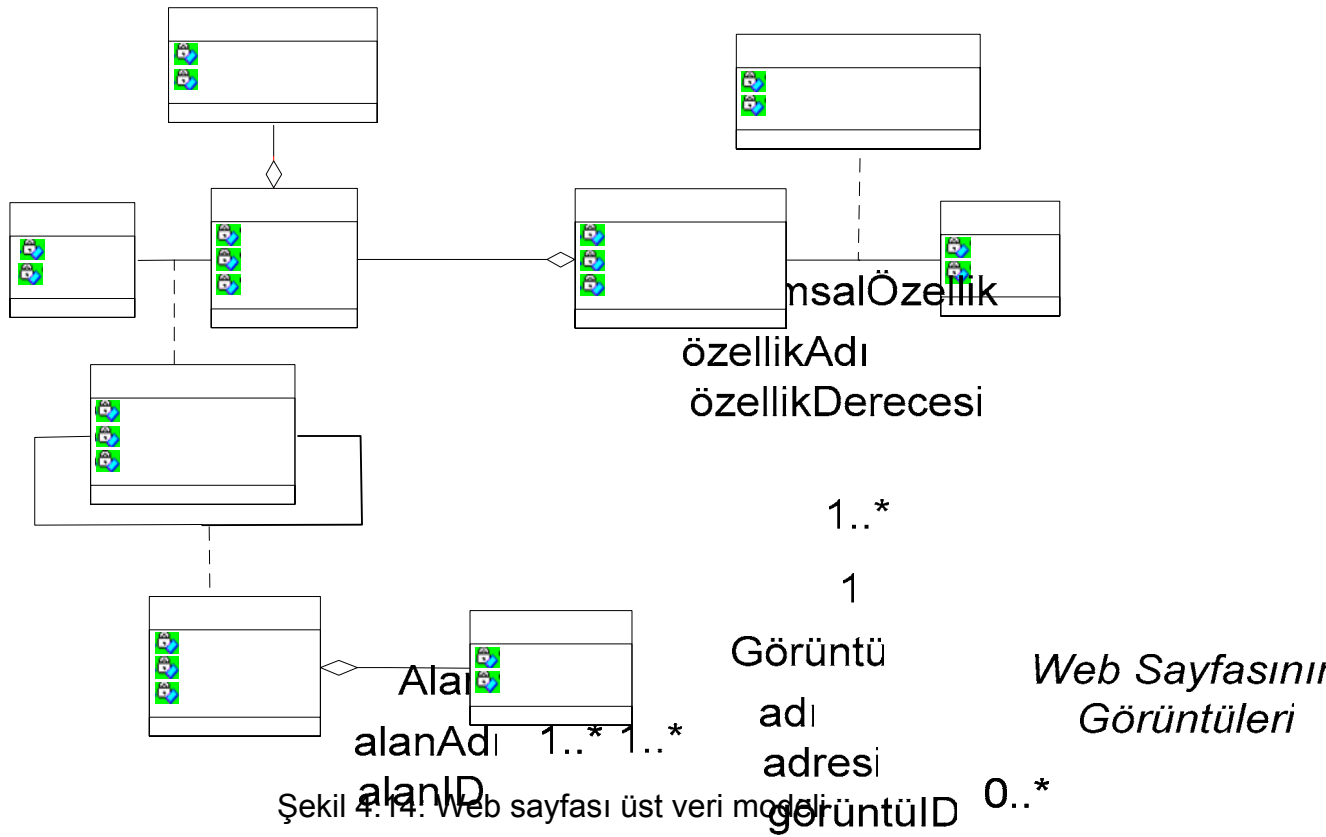
Bu noktaya kadar yapılan açıklamaların hepsi metin ve görüntüler için sayfanın içerik özelliklerinin çıkarımına yani içerik özelliklerinden oluşan üst verilerin üretimine ilişkindir. İlerleyen bölümde üst verilerin ontoloji ile ilişkilendirilip, anlamsal özelliklerin çıkarımı, yani ontoloji kullanımı açıklanacaktır.

## **4.5. Web Sayfasının Anlamsal Özelliklerinin Çıkarımı**

Bu bölümde web sayfasının üst veri modeli, üst veride yer alan içerik özellikleri ve alan ontolojisinden çıkarsanan anlamsal özelliklerin çıkarılma biçimi ve üst veriye eklenmesi açıklanacaktır.

### **4.5.1. Web Sayfası Üst Veri Modeli**

Web sayfalarının üst veri modeli, içerdiği görüntü ve metinlerin hem içerik hem de anlamsal özelliklerini tutabilecek biçimde önerilmiştir. İlk olarak içerik özellikleri ile ilgili kesimleri doldurulur. Daha sonra alan ontolojisi ile çıkarsanan anlamsal özellikler eklenir. Üst veri modelinin UML sınıf çizeneği ile gösterimi şekil 4.14'te yer almaktadır. Ayrıca üst veri modelinin RDF ile ifadesi Ek-4'te yer almaktadır.



Şekil 4.14. Web sayfası üst veri modeli

Şekil 4.14'te görüleceği üzere web sayfaları metin ve görüntülerden oluşmaktadır. Web sayfasının metinleri, metni oluşturan terimler ile birçoğa birçok özelliğinde bağıntıya sahiptir. Bağıntıdan çıkan bağıntı sınıfı "hamAğırlık" ve "çıkarsanmışAğırlık" olmak üzere iki ayrı niteliğe sahiptir. "hamAğırlık" niteliği terimin VSM'ye uygun ağırlıklandırılması sonucu elde edilen değeri göstermektedir. "çıkarsanmışAğırlık" Niteliği ise terimlerin alan ontolojisinde taşıdıkları ilişkilerin ve "hamAğırlık"larının birlikte kullanılması ile elde edilen ağırlıktır. Yani sayfanın anlamsallığını ifade eden terimlerin "çıkarsanmışAğırlık" değerleri "hamAğırlık"larından fazla olacaktır. "çıkarsanmışAğırlık" Niteliğinin hesaplanması bölüm 4.5.2'de açıklanmıştır.

### AlanMatrisi

Görüntüler için yapılan modellemede bir görüntünün alanlardan oluşacağı düşünülmüş ve her bir alanın adı ve görüntü için kullanılmıştır. Buradaki "alanAdı" niteliği kullanıcının AEA'nın etiketler arayüzünde sunduğu etiketlerdir. Alanlar da birlikte varoluşları sebebiyle birçoğa birçok özellikte bağıntıya sahiptir. Bu bağıntıdan çıkan bağıntı sınıfının adı "alanMatrisi" dir.

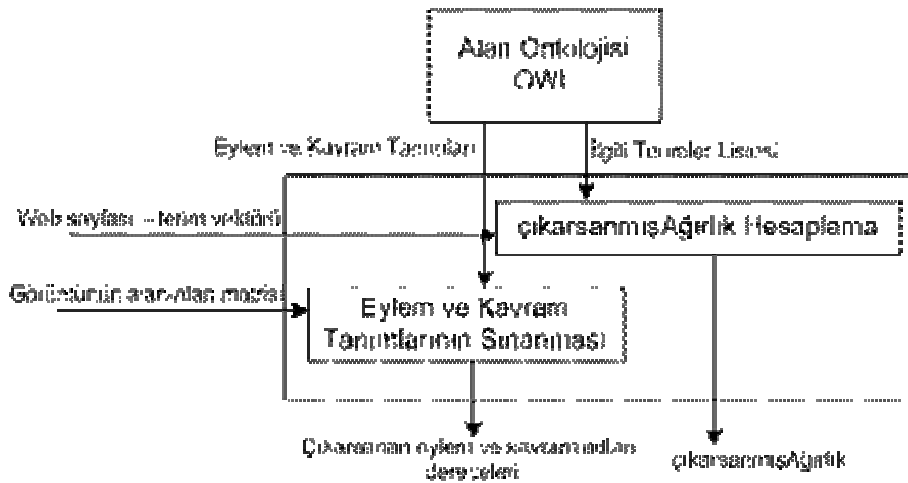
“alanMatrisi”nde iki alan arasında çıkarılan konumsal ilişkiler mevcuttur. “Alan” ve “alanMatrisi” sınıfları görüntünün içerik özelliklerini tutar. Görüntünün içerdiği “üstAnlamsalÖzellik” sınıfı ise alan ontolojisinde yapılan tanımlar ile görüntünün “alan” ve “alanMatrisi” bilgilerinin karşılaştırılması ile elde edilen anlamsal özellikleri içerir. Yani ontolojinin yönlendiriciliği ile görüntüde var olduğu çıkarılan eylem ve kavram adları, çıkarılma dereceleri ile birlikte “üstAnlamsalÖzellik” sınıfının olgusunu oluşturur.

Sonuç olarak üst veri modelinde yer alan terimler için “çıkarılmışAğırlık” niteliği ve görüntüler için “üstAnlamsalÖzellik” sınıfı, anlamsal eklentilerin yapılabileceği iki uç olarak tanımlanmıştır.

#### 4.5.2. Alan Ontolojisi İle Üst Veri'nin Anlamsal Genişletimi

Bu bölümde, alan ontolojisi ile web sayfası üst veri modelinde yer alan içerik özelliklerinin ilişkilendirilmesi açıklanacaktır.

Aynı alan ontolojisi web sayfasının görüntü ve metinlerine hizmet verecektir. Görüntüler için beklenti, görüntüde tespit edilen nesne ve konumsal ilişkilerin yani alan matrislerinin, ontolojide tanımlanmış karşılıklarının tespit edilmesidir. Metinler için beklenti ise birlikte yer alan terimlerin, ontolojide tanımlanmış ilişkileri var ise, bu tür terimlerin birlikteliklerinin öne çıkarılmasıdır. Şekil 4.15 burada yapılan çok genel ve sözel ifadeyi çizim ile göstermektedir.



Şekil 4.15: Üst veri ve alan ontolojisi ilişkisi



Şekil 4.15'te görüleceği üzere alan ontolojisi ile üretilecek üç tür bilgi vardır: görüntünün içerdiği eylem ve/veya kavram adları ve dereceleri, terimler için beklenen çıkarsanmış ağırlık. Aşağıda bu üç bilginin çıkarsanması maddelenerek açıklanmıştır.

### 1. Görüntünün içerdiği eylemlerin çıkarsanması:

Ontolojinin içindeki eylem tanımları nesne ve konumsal ilişkilerden oluşan eylem kuralları ile tarif edilmektedir. Görüntünün içerik özelliklerindeki alan matrisi de alanların birbiri ile konumsal ilişkisini göstermektedir ve her alan etiketlenmiş bir nesneye karşılık gelmektedir. Şu durumda eylem kurallarının alan matrisleri üzerinde sınırlanabileceği bir ortam oluşmuş demektir. Eylem tanımları bulanık biçimde oluşturulan tanımlardır ve alan matrisi de bulanık değerler içerir. Şöyle ki; "a", "b", "c", "d" eylem kuralında kullanılan nesne adları ve "e" de eylemin adı olmak üzere tipik bir eylem kuralı şu biçimde görünür:

Kural : if (a üstünde b) && (c altında, hizasında d) then  $\mu_{\text{eylem}}(e) = \text{eylemDerecesi}$

Daha açık bir ifade ile

Kural:if (a üstünde b) && ((c altında d) || (c hizasında d)) then  $\mu_{\text{eylem}}(e) = \text{eylemDerecesi}$

Kuralda yer alan "eylemDerecesi" ifadesi eylemin yapılan tanımının eylemi çıkarsamada ne kadar anlamlı olduğunu göstermektedir. Yani tanım 1 derecesi ile sağlansa bile eylem 1 derecesinde çıkarsanmayabilir. Yani bu derece yapılan tanımın ne kadar keskin olabildiğini göstermektedir. Ontolojideki bu eylem kuralına karşılık görüntünün alan matrisinde "a", "b", "c" ve "d" nesnelere ve çıkarılan konumsal ilişkileri içeren matrisin görünümü çizelge 4.6'da gösterilmiştir:

Çizelge 4.6: Eylem çıkarsama için örnek alan matrisi

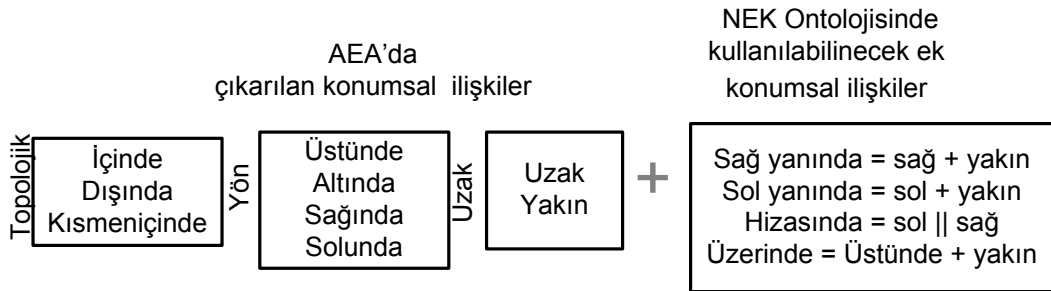
	a	b	c	d
a	————	$\{\mu_{\text{topolojik}}(a,b), \mu_{\text{yön}}(a,b), (\mu_{\text{uzaklık}}(a,b))\}$	$\{\mu_{\text{topolojik}}(a,c), \mu_{\text{yön}}(a,c), (\mu_{\text{uzaklık}}(a,c))\}$	$\{\mu_{\text{topolojik}}(a,d), \mu_{\text{yön}}(a,d), (\mu_{\text{uzaklık}}(a,d))\}$
b	$\{\mu_{\text{topolojik}}(b,a), \mu_{\text{yön}}(b,a), (\mu_{\text{uzaklık}}(b,a))\}$	————	$\{\mu_{\text{topolojik}}(b,c), \mu_{\text{yön}}(b,c), (\mu_{\text{uzaklık}}(b,c))\}$	$\{\mu_{\text{topolojik}}(b,d), \mu_{\text{yön}}(b,d), (\mu_{\text{uzaklık}}(b,d))\}$
c	$\{\mu_{\text{topolojik}}(c,a), \mu_{\text{yön}}(c,a), (\mu_{\text{uzaklık}}(c,a))\}$	$\{\mu_{\text{topolojik}}(c,b), \mu_{\text{yön}}(c,b), (\mu_{\text{uzaklık}}(c,b))\}$	————	$\{\mu_{\text{topolojik}}(c,d), \mu_{\text{yön}}(c,d), (\mu_{\text{uzaklık}}(c,d))\}$
d	$\{\mu_{\text{topolojik}}(d,a), \mu_{\text{yön}}(d,a), (\mu_{\text{uzaklık}}(d,a))\}$	$\{\mu_{\text{topolojik}}(d,b), \mu_{\text{yön}}(d,b), (\mu_{\text{uzaklık}}(d,b))\}$	$\{\mu_{\text{topolojik}}(d,c), \mu_{\text{yön}}(d,c), (\mu_{\text{uzaklık}}(d,c))\}$	————

“a”, “b”, “c” ve “d” alanlarını içeren bir G görüntüsünün üst anlamsal özelliği “e” adlı eylem olabilir mi ya da ne derecede olabilir sorusunun cevabı, kuraldaki yapı kullanılarak, bulanıklık dereceleri ile şu biçimde oluşturulur:

Kural: if  $(\mu_{\text{üst}}(a,b)) \ \&\& \ ((\mu_{\text{alt}}(c,d)) \ || \ ((\mu_{\text{hizasında}}(c,d)))$  then  $\mu_{\text{eylem}}(e) = \text{eylemDerecesi}$   
 $\&\& \Rightarrow \min$  ve  $|| \Rightarrow \max$  ile ele alınır

$$\mu_e(G) = (\min(\mu_{\text{üst}}(a,b), \max((\mu_{\text{alt}}(c,d)), (\mu_{\text{hizasında}}(c,d)))) * \text{eylemDerecesi}$$

Yukarıda görüleceği üzere kural tanımında kullanılabilinecek konumsal ilişkiler otomatik çıkarsananlardan daha fazladır. Şekil 4.16’da NEK ontolojisinde kullanılabilinecek tüm konumsal ilişkilerin listesi ve kaynakları görülmektedir:



Şekil 4.16: NEK Ontolojisinde kullanılan konumsal ilişkiler

## 2. Görüntünün içerdiği kavramların çıkarsanması:

Kavram çıkarsama, eylem çıkarsama işleminden sonra gerçekleştirilir. Ontolojide ter alan “bağlamındaYerAlır” ilişkisi bir kavramın anlamını oluşturan ya da kavramın kendi başına kullanıldığında da ifade edebileceği nesne, eylem ve diğer kavramları, seçilen kavrama bağlar. Bu ilişki bulanıktır. Çünkü ilişkinin

gerçek dünyadaki yorumu da bulanıktır. Alan modellenirken uzman bu bağıntıyı uygun biçimde derecelendirir. Yani “e” eylemi, “k” kavramının bağlamında yer alıyorsa bu bulanık ilişki şu biçimde ifade edilir:  $\mu_{kr}(e) =$  bağlamındaYerAlmaDerecesi. Bu ifadede “bağlamındaYerAlmaDerecesi” modelleyicinin atadığı [0..1] aralığında bir değerdir. Örneğin “voleybol” alanında “blok” eylemi “savunma” kavramının bağlamında yer alma derecesi 1.0 ise  $\mu_{savunma}(blok) = 1.0$ 'dır. Bir başka örnekle “gövde” alanında “Akciğer” eyleminin, “toraks” kavramının bağlamında yer alma derecesi  $\mu_{Toraks}(Akciğer) = 1.0$  iken “abdomen” kavramının bağlamında yer alma derecesi  $\mu_{Abdomen}(Akciğer) = 0.2$  olabilir. Eğer bir G görüntüsünde “e” eyleminin çıkarsanma derecesi  $\mu_e(G)$  ise ve ontolojide tanımlanmış  $\mu_{k\_bağlamındaYerAlır}(e)$  değeri var ise, görüntünün kavramı ifade etme derecesi olan  $\mu_k(G)$  şu biçimde hesaplanır:

$$\mu_k(G) = \mu_{k\_bağlamındaYerAlır}(e) * \mu_e(G)'dir$$

Aynı yaklaşım kavramın bağlamında yer alan nesnelere ve diğer kavramlar için de geçerlidir. Örneğin  $k_i$  kavramın bağlamında alan  $k_j$  kavramı tanımlanmış ise ve G görüntüsünde  $k_j$  kavramı çıkarsanmış ise

$$\mu_{k_i}(G) = \mu_{k_i\_bağlamındaYerAlır}(k_j) * \mu_{k_j}(G) olacaktır.$$

**3. Terimler için “çıkarsanmışAğırlık”ların hesaplanması:** Terimler için eylem ya da kavramların çıkarsanması olarak adlandırmak yerine “çıkarsanmışAğırlık” değerinin hesaplanması denilmesinin sebebi şudur: “çıkarsanmışAğırlık” hesaplanırken eylem ve kavram adları da kullanılmaktadır. Yani bu ifade, eylem ve kavram çıkarsamayı da içermektedir. Şöyle ki; eğer ontolojide bir kavramın bağlamında yer aldığı belirtilen terimler, metin içinde kullanılıyorsa; o kavram metin içinde hiç geçmese bile bağlamında yer alan terimlerin metindeki varlıkları sebebiyle “çıkarsanmışAğırlık” değerine sahip olacak ve metnin terim listesine eklenecektir. Örneğin “gövde” ontolojisinde varolan “abdomen” kavramının bağlamında “sindirim sistemi” ve “çift böbrek” eylemleri yer alacaktır. Çünkü her iki eylem de gövdenin abdomen bölgesindedir. Eğer web sayfasında “çift böbrek” ve/veya “sindirim sistemi” terimleri ağırlıklı biçimde yer alıyorsa, sayfanın terim listesine “abdomen” terimi de eklenecektir. Dolayısı ile kavram çıkarsanmış olacaktır. Ancak burada uygulanan yaklaşımın adı

kavram ya da eylem çıkarsama değildir. Çünkü ontoloji birbiri ile ilişkili terimleri eylem ya da kavram adları altında toplamıştır. Amaç bu terim ilişkilerini kullanarak, birlikte kullanılan terimlerin ağırlıklarını artırmak ve birbiri ile ilişkili terimlerden hareketle yeni terimleri çıkarsamaktır. Bu amaçla “çıkarsanmışAğırlık” değerini elde etmek için kullanılan hesaplama biçimi aşağıda formülle ifade edilmiştir:

$$W_{im} = w_{im} + \sum_{p=0}^{o} (w_{ip} \mid p \neq m \ \&\& \text{Ontolojikİlişki}(t_m, t_p) \neq 0) / \text{ont.terim sayısı}$$

o = ontolojideki terim sayısı

$W_{im}$ , i. sayfadaki m. terimin çıkarsanmış ağırlığı ve  $w_{im}$  m. terimin i. sayfadaki ham ağırlığıdır.  $\text{Ontolojikİlişki}(t_m, t_p)$ , m. terimle ontolojide ilişkilendirilmiş p. terimi göstermektedir. Sözel bir ifade ile yapılan hesaplama terimin ham ağırlığına, terimle ontolojik ilişkisi olan terimlerin o web sayfasındaki ağırlıklarının toplamını, ontolojideki terim sayısına bölerek ekler. Böylece terim ontolojide ilişkilendirildiği terimler ile ne kadar birlikte kullanılıyorsa o oranda ağırlığı artar. Ham ağırlık değer 0 bile olsa, ilişkili olduğu terimlerin sayfada geçmesi sebebiyle çıkarsanmış ağırlık kazanır.

Görüleceği üzere ontoloji ile ilişkilendirmede, web sayfasının ontolojiden edinebileceği her tür anlamsal çıkarsama sonucu, sayfanın üst verisine eklenir. Yani ontolojiden çıkarsanabilecek tüm yeni anlamsal özellikler bütünüyle çıkarsanır ve üst veri en geniş içeriğine kavuşur. İlerleyen bölümde sayfaların üst verileri üzerinden sorgulanması anlatılmıştır.

#### 4.6. Sorgulama

Sistemde anlamsal çıkarsama sonucu her sayfa için elde edilen üst veriler RDF ile ifade edilir ve saklanır. Yani sistemde anlamsal çıkarsama sonucu elde edilen bir RDF havuzu (RDF repository) oluşur. Yani bu noktada sistem, Anlamsal Web'in bilgi yönetimi anlayışına uyumlu davranmıştır. Dolayısıyla, sistemin sorgulama biçimi de bu çizgi biçimiyle baştan belirlenmiştir. Yani RDF belgeleri RDQL ile sorgulanmıştır. Ancak elbette kullanıcıdan alınan sorgu kullanıcı dostu arayüzü ile kullanılan dilden bağımsızdır.

Kullanıcı sorgusunda, alanda yer alan nesne, eylem, kavram adları ya da olay niteliklerini yani alan terimlerini kullanabilir. Sorgulamayı sadece “metin”, sadece “görüntü”ler üzerinden yapabileceği gibi “metin ve görüntüler” veya “metin ya da görüntüler” üzerinde de yapabilir. Sorgu terimleri ve hangi veri türüne ait olduğu bilgisi arayüzden alındıktan sonra sistem amaca uygun sorguyu RDQL ile ifade eder. Burada kullanılacak terimler arası “ve” veya “ya da” işleçleri de desteklenmiştir. Sorgu sonucu liste halinde kullanıcıya döndürülür. Kullanıcı istediği belgeye listedeki adı üzerinden erişir.

Burada ontoloji ile sorgu arasındaki ilişkinin açıklanması gerekmektedir. Burada iki yaklaşım söz konusu olabilir. İlk yaklaşımda sistem sorgulama esnasında web sayfası ve sorgunun anlamsal özelliklerini çıkarsamak için ontoloji ile ilişkilendirir. İkinci yaklaşımda web sayfası derlemi için anlamsal çıkarsamalar sayfaya işlenir. Sorgulamada ontoloji ile ilişki kurulmaz. Bu çalışmada ikinci yaklaşım uygulanmıştır. Bunun sebebi de sistemin RDF üst veri havuzunda çalışması ve paylaşılabilirlik gereği sistemin bağımsız RDF havuzu üretebilmesidir. Böylece başka arama motorlarının da ontolojye ihtiyaç duymaksızın derlemi sorgulayabilmesi standart erişim dili olan RDQL ile mümkün hale gelir. Bu seçimde sorgu ontoloji ile ilişkilendirilip genişletilmez. Genişleme web sayfalarının üst verilerinde gerçekleşir. Örneğin derlemde yer almayan bir terim ile sorgulama yapılıyorsa, bu terimin ilişkili olduğu terimleri sorguya eklemek yerine, terimin ilişkili olduğu diğer terimlerin varlığı sebebiyle ekleme web sayfasının üst verisine yapılır. Dolayısı ile terim web sayfasında yer almıyorsa da sayfanın üst verisinde yer alır.

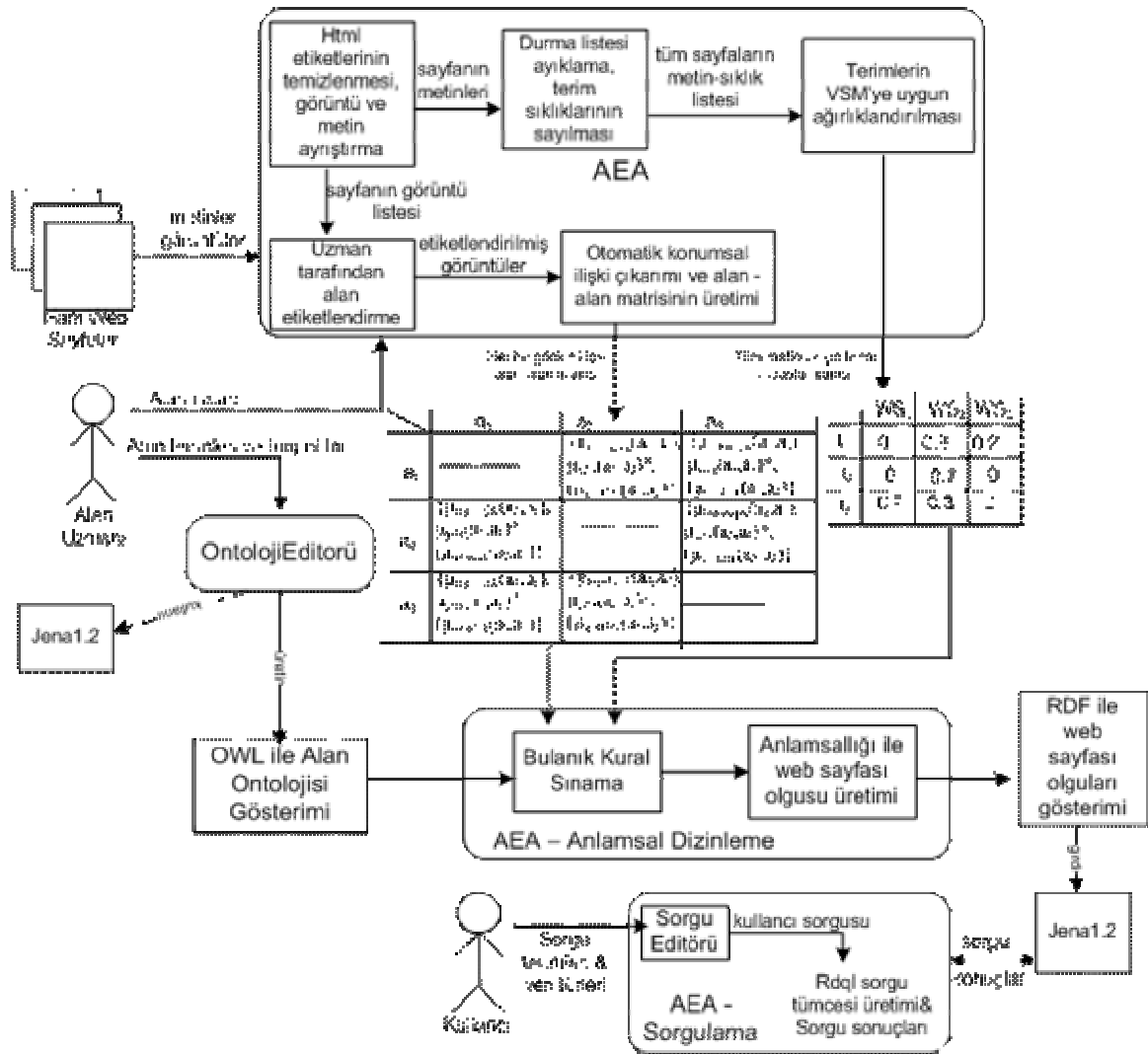
Sistemin hedef olarak tanımladığı anlamsal sorgulama, içerik ve anlamsal özelliklerin birlikte saklandığı üst verileri sorgulanması ile sağlanır. Örneği “voleybol” alanında bir görüntü için ancak anlamsal özellik olabilecek “servis” terimi, ya da voleybol oyuncusu “Neslihan”ın attığı “smaç servis”ler için metinde “Neslihan” ve görüntüde “smaç servis” terimleri ile sorgulama yapılabilir. Bir başka örnekle kullanıcı görüntü üzerinde “sindirim sistemi” biçiminde bir bilgi ihtiyacı bildirirse sadece görüntüler üzerinde anlamsal çıkarımla elde edilebilen bir eylemi sorgulamış olacaktır.

Örnek sorgular ve sonuçları bölüm 5’te deneyler kısmında yer almaktadır.

## 4.7. Sistem Mimarisi

Bu bölümde, Bölüm 4 kapsamında açıklanan sistem işleyişi ve bileşenleri bir bütün olarak ele alınıp, sonuçlandırılacaktır.

Sistemin varolan iki temel bileşeni Ontoloji Editörü (OE) ve Anlamsal Erişim Aracı (AEA)'dır. Bu iki aracın girdi ve çıktılarının birlikte ve ayrıntılı görünümü şekil 4.17'de verilmiştir.



Şekil 4.17: En genel sistem bileşenleri ve ilişkileri

Şekil 4.17 öncelikle bileşenler ve ilişkileri için çizilmiş olmakla birlikte yukarıdan aşağıya doğru işlem sırasını da ifade etmektedir. Bölüm 4'ü özetleyen işlem adımları şu biçimdedir: öncelikle OE ile alan ontolojisi oluşturulur. Daha sonra alana ait web sayfalarının içerik özellikleri çıkarılır. Kullanıcı bu esnada görüntü

etiketlendirir. İçerik özellikleri ve alan ontolojisi ile anlamsal özellikler üretilir. Tüm bu veriler sayfanın üst verisini oluşturur. Tüm sayfaların üst verileri ile RDF havuzu oluşur. Sorgulama havuz üzerinden gerçekleştirilir.

Geliştirilen sistem ontoloji yükleme, saklama gibi yetenekleri için Jena 2.1 adlı kütüphane yazılımını kullanmaktadır. Jena 2000 yılında HP Laboratuvar'ının Anlamsal Web grubunca geliştirilen, açık kaynak kodlu bir ontoloji kütüphane yazılımıdır[151]. Jena, RDF ve OWL dillerini destekler. Yani bu diller ile ifade edilen ontolojilerin çizgelerinin oluşturulması, değiştirilmesi, sorgulanması ve yeniden dille ifade edilmesini sağlar. Geliştirilen sistemden ontolojinin kullanıldığı her işlem adımında modelin bellekteki çizgesi Jena ile oluşturulmuş ve yine uygun Jena komutları ile çizge üzerinde işlem yapılmıştır.

İlerleyen bölümde sistemin başarımı tartışılmıştır.

## BÖLÜM 5. SİSTEM BAŞARIMI ÖLÇME DENEYLERİ VE SONUÇLARI

Geliştirilen tezin hedeflediği problem aynı sorgu cümlecığı ile farklı veri türlerini anlamsal sorgulayabilmektir. Bu sebeple farklı veri türü içeren ve sorgulanma ihtiyacı daha çok göz önünde olan web sayfaları tercih edilmiştir. Web sayfalarındaki metin ve görüntü türü veriler birlikte ele alınmıştır. Çünkü temel öngörü bu iki veri türünün birlikte kullanım amacının belli bir anlamı vurgulamak ya da zenginleştirmek olduğu biçimindedir. Dolayısı ile birlikte kullanıldıkları gibi birlikte sorgulanmalarının gerekliliği ana fikri ile hareket edilmektedir. Bu açıklamalara paralel olarak sistemden beklenen temel beklentiler:

1. Web sayfasının görüntüsünde ve metninde ortak olan anlamsallığın öne çıkması
2. Web sayfasının görüntü ve metnindeki anlamsallığın birbirini tamamlaması biçimindedir.

Birinci maddede “anlamsallığın öne çıkması” ile ifade edilen durum, anlamsallığın zaten sayfanın metninde tespit edildiği durumda, aynı anlamsallığın görüntüden de çıkarsanmasıdır. Bu durumda erişim başarımına pozitif katkı beklenir. Şöyle ki; bu katkı ilgililik derecesini değiştirdiği için erişim çıktısında yer alan sıralamayı değiştirmelidir.

İkinci maddede “birbirini tamamlaması” ile ifade edilen durumda, anlamsal özellik sadece tek veri türünden edinip, diğer veri türünün katkısını ihmal etmek yerine her iki veri türünün anlamsal özelliklerinin birleşiminin kullanılması kastedilmektedir. Bu katkı sayfanın anlamsal özelliklerini zenginleştirdiği için somut katkısı daha fazla sayfanın erişim çıktısında yer almasıdır.

Özetle birinci madde sayfanın anlamsallığını dikeyde derinleştirirken, ikinci madde de anlamsallığı yatayda genişletmektedir.

Sistemden sadece metin tabanlı yaklaşımda beklenen katkı yine anlamsallığın dikeyde ya da yatayda genişlemesidir. Terimlerin ağırlıklarının hesaplanmasında kullanılan ontolojik ilişkiler sebebi ile, eğer terim sayfada hiç geçmiyor ama



dizinlemeye giriyorsa, yatayda bir genişleme gerçekleşmiştir. Bu durumda erişim çıktısında uzama beklenir. Sayfada geçen bir terim, aynı sayfada ontolojik ilişki taşıdığı terimler ile birlikte yer alıyorsa, çıkarsanmış ağırlığı ham ağırlığında fazla olacaktır ve sayfanın anlamsal vurgusunu (dikeyde genişleme) yansıtacaktır. Bu etkinin erişim çıktısındaki sıralamayı değiştirmesi beklenir.

Sistemin başarımını ölçmekte kullanılan parametreler Anma (Recall), duyarlık (Precision) ve  $R_{norm}$  olmak üzere 3 tanedir. Anma değeri, derlemde sorgu ile ilgili olan belgelerin erişim oranını ifade eder. Yani

$$\text{Anma} = \frac{\text{Erişim çıktısındaki ilgili belge sayısı}}{\text{Belge derlemindeki ilgili belge sayısı}}$$

biçiminde hesaplanır[156].

Duyarlık değeri, erişim çıktısında yer alan belgelerin, gerçekte ne kadarının sorgu ile ilgili olduğunu ifade eder. Yani

$$\text{Duyarlık} = \frac{\text{Erişim çıktısındaki ilgili belge sayısı}}{\text{Erişim çıktısındaki belge sayısı}}$$

biçiminde hesaplanır[156].

Anma ve duyarlık parametreleri için en iyi durum her iki değer de 1 olmasıdır. Ancak yapılan çalışmalarda anma ile duyarlık arasında kısmen ters ilişki olduğu söylenebilir[156]. Yani daha çok ilgili belgeye erişmek için erişim çıktısının uzaması aynı zamanda daha çok ilgisiz belgenin de çıktıya girmesine sebep olabilir. Bu noktada ilgisiz belgelerin erişim çıktısında nasıl bir yerleşime sahip olduğu önem kazanır.

$R_{norm}$  değeri erişim çıktısındaki sıralamanın ne oranda doğru sıralama olduğunu ifade eder.  $S^+$ , erişim çıktısında ilgili belgelerin ilgisiz belgelerin önünde yer aldığı belge çifti sayısını;  $S^-$  ilgisiz belgelerin ilgili belgeler önünde yer aldığı belge çifti sayısını,  $S^+_{max}$  erişim çıktısında olabilecek en yüksek  $S^+$  değerini ve  $\Delta$ 'de erişim çıktısını gösterirse

$$R_{\text{norm}}(\Delta) = \frac{1}{2} \left( 1 + \frac{S^+ - S^-}{S^+_{\text{max}}} \right)$$

biçiminde hesaplanır[156].

Sistem başarımını ölçmek için öncelikle bir alan seçilmiş ve modellenmiştir. Daha sonra alan ile ilgili web sayfaları ile kapalı bir derlem oluşturulmuş ve dizinlenmiştir. Daha sonra çeşitli tür ve anlamsallık seviyesindeki alan terimleri ile sorgulama yapılmıştır. İlerleyen bölümde sistemin başarımlarını için kullanılan iki ayrı alan ve sonuçları açıklanmıştır.

### 5.1. Başarım Deneylerine Ait Ortamların Tanıtımı

Sistemin başarımını sınamak için iki ayrı alan seçilmiştir. İlk deneyler internet ortamından sağlanan veriler ile kolektif spor dallarından “voleybol” alanı için ve ikinci deneyler, verileri elle üretilen (sentetik), tıp bağlamında “gövde” için gerçekleştirilmiştir. Deneylerde, geliştirilen sistemin başarımlarını metin tabanlı sorgulama aracı olan Apache Lucene[157] ile karşılaştırılmıştır. Apache Lucene, tam metin dizinleme ve sorgulama yeteneğinde, java ile geliştirilmiş, açık yazılımdır. Dolayısıyla deneylerde üç tür sorgulama gerçekleştirilmiştir:

- Apache Lucene ile metin tabanlı sorgulama
- Tezde geliştirilen sistem ile sadece “metinde” anlamsal sorgulama: görüntü veri türünü ihmal eden yaklaşımdır. Ontoloji destekli olması sebebiyle ilişkili terimlerin birlikte kullanımından kaynaklanan anlamsal genişlemeyi dikkate alır.
- Tezde geliştirilen sistem ile “metin ya da görüntüde” anlamsal sorgulama: her iki veri türü aynı sorgu cümlecisi ile sorgulanır. Sistemin farklı veri türlerinin birlikte kullanımının anlamsallığa getirdiği yatay ve dikey genişlemeyi dikkate alan sorgulama biçimidir.

Her iki alan için deney ortamı aşağıda ayrıntılandırılmıştır:

- Deney 1: “Voleybol” alanı için gerçekleştirilen deneylerde modelleyici uzman tez yazarı olmak üzere “voleybol” spor dalı için NEK Ontoloji Modeli kullanılarak “Voleybol Alan Ontolojisi” oluşturulmuştur. Deney derlemi, internet ortamında bulunan 15 adet gerçek web sayfası olup, hem görüntü hem de metin içerirler. Bu sayfalar genelde voleybol sporu ile ilgili güncel haberleri içermektedir: turnuvalar, maç özetleri ve haberleri, yorumlar vb. gibi. Deneyde toplam 6 adet sorgu kullanılmıştır. Bu sorgular sırası ile “smaç”, “servis”, “blok”, “vuruş”, “hücum”, “maç” olmak üzere 3’ü eylem ve 3’ü de kavram düzeyindedir. Yani anlamsal çıkarım ile elde edilebilecek düzeydedirler.
- Deney 2: Tıbbi bağlamda “Gövde” alanı için gerçekleştirilen deneylerde modelleyici uzmanlar Ufuk Üniversitesi, Tıp Fakültesi’nden Dr. Kemal Arda ve radyoloji asistanı Nazan Çiledağ NEK Ontoloji Modelini kullanarak “Gövde Alan Ontolojisi”ni oluşturmuşlardır. Tez yazarı alan bilgisini geliştiren Ontoloji Editörü’ne aktarmıştır. Deney derlemi, gövdeye ait 40 adet elle hazırlanan sentetik web sayfasından oluşmaktadır. Web sayfaları üretiminde, 40 adet CT kesit görüntüsü kullanılmıştır. Her bir web sayfası en az 1 adet görüntü içermektedir. Web sayfalarının metin kısımları elle üretilmiştir. Alan uzmanının yönlendiriciliği ile üretilen metinlerde gövdede yer alan doku, organ kesimi ve organ adlarına yer verilmiştir. Her bir organ için de birer adet doğrudan adı ile yer almadığı, ancak gövde bağlamında ilgili olduğu diğer organ ve kesim adlarının yer aldığı web sayfası düzenlenmiştir. Sorguların niteliği alan uzmanları ile birlikte belirlenmiş ve toplam 10 adet sorgu kullanılmıştır. Sorgular sırası ile “Akciğer”, “Tam Karaciğer”, “Çift Böbrek”, “Tam Mide”, “Sindirim Sistemi”, “Sağ Böbrek”, “Tam Pankreas”, “Sol Böbrek”, “Toraks” ve “Üst Abdomen” biçimindedir. Görüleceği üzere sorgu terimlerinin anlamsallığı ilk 8’i için eylem düzeyinde ve son 2’si için kavram düzeyindedir. “Toraks” ve “Üst Abdomen” kavramlarının doğrudan yer aldığı bir metin düzenlemesi yapılmamıştır. Zira aslen amaçlanan her bir görüntünün içerdiği nesnelerin metin kesimine yansımalarıdır. Bu sebeple kavram adlarına metinlerde yer verilmemiştir. Derlemde her bir sorgu için ilgi seviyesine göre 4 tür sayfa bulunmaktadır:

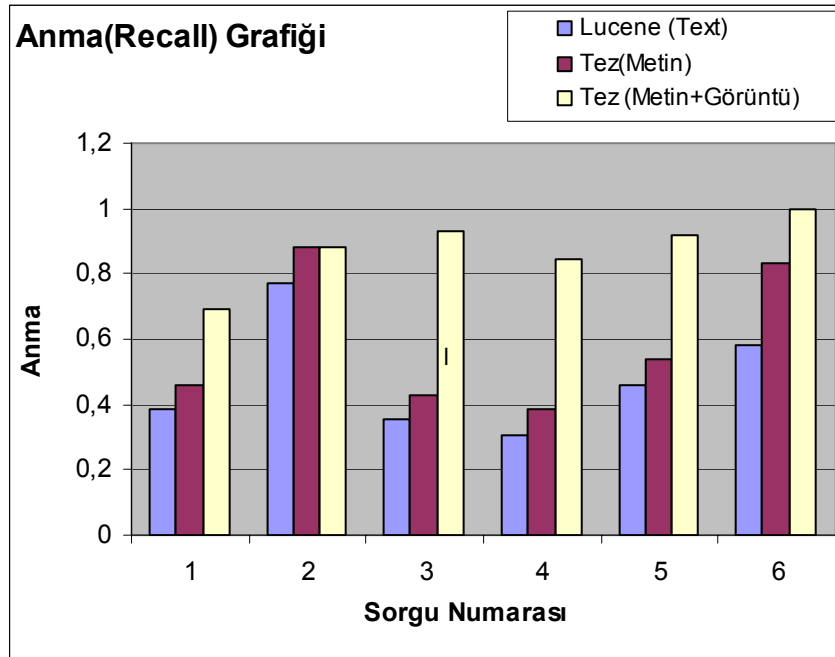
1. Sorgu ile hem metni hem de görüntüsü ilgili sayfalar
2. Sorgu ile sadece metni ilgili sayfalar
3. Sorgu ile sadece görüntüsü ilgili sayfalar
4. Sorgu ile hem metni hem de görüntüsü ilgisiz sayfalar.

## 5.2. Başarım Deneyleri Sonuçları

Bu kesimde her iki deney ortamında gerçekleştirilen üç tür erişim biçimine ait başarım değerlendirmesi anma, duyarlık ve  $R_{norm}$  parametrelerine göre grafikler ile ifade edilmiş ve değerlendirilmiştir.

### 5.2.1. Deney 1'e Ait Başarım Sonuçları

Voleybol alanı için 6 adet sorgu ile gerçekleştirilen erişimlerin anma, duyarlık ve  $R_{norm}$  grafikleri sırası ile şekil 5.1, şekil 5.3 ve şekil 5.2'de verilmiştir.

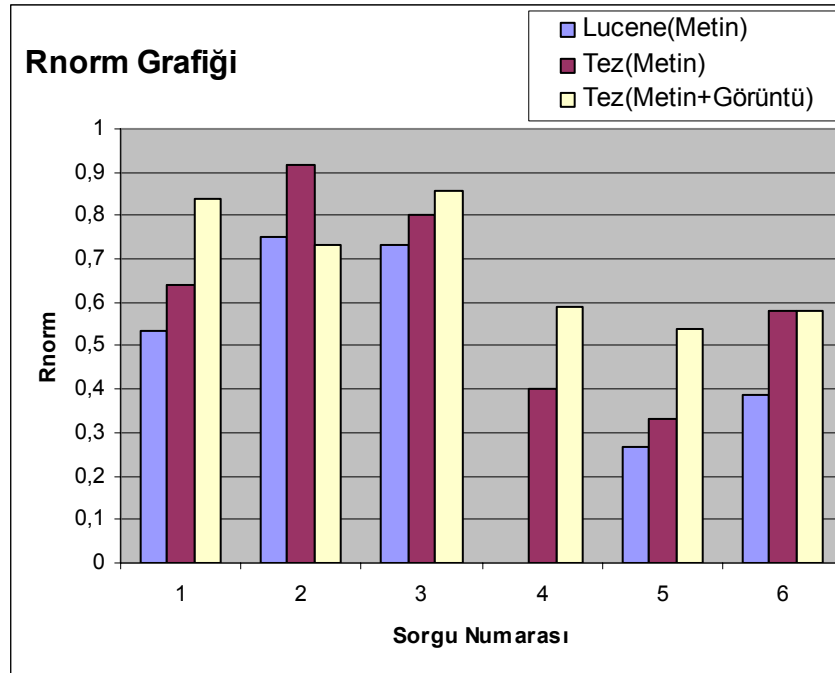


Şekil 5.1: Anma değeri değişim grafiği (Deney1)

Şekil 5.1'de görüleceği tezde önerilen yaklaşımla geliştirilen metin tabanlı anlamsal erişim, Lucene ile gerçekleştirilen metin tabanlı sorgulamaya göre daha iyi sonuç üretmiştir. Lucene'de kullanılan yaklaşımda bir terimin bir sayfa için ağırlık kazanabilmesi için terimin o sayfada en az bir kere geçiyor olması gerekir. Oysa tezde geliştirilen metin tabanlı anlamsal erişimde terim sayfada geçmiyor

olsa bile ontolojik olarak ilişkilendirildiği diğer terimlerin sayfadaki varlığı sebebiyle ağırlık değeri kazanabilir. Özetle var olan başarımların artışı sayfaların sadece içerdikleri terimler üzerinden değil, aynı zamanda ontolojide var olan terimler arası ilişkiler sebebiyle kazandıkları yeni terimler üzerinden dizinlenmesi ile ortaya çıkmıştır.

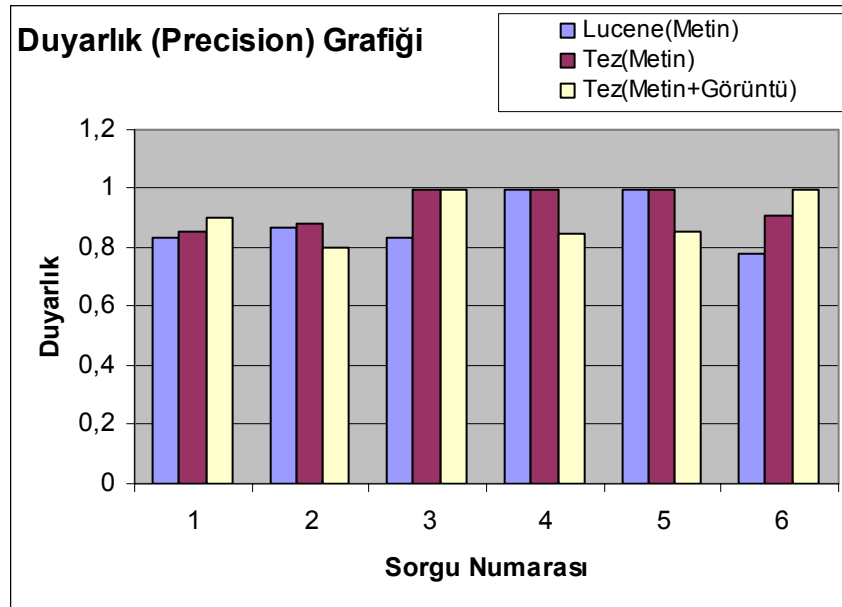
Şekil 5.1’de görüleceği üzere, tezde geliştirilen sistem ile gerçekleştirilen metin veya görüntü tabanlı anlamsal erişimde diğerlerine oranla anma değeri üzerinde yükselme gerçekleşmiştir. Bu değerlerin artışını sağlayan etken sayfaların görüntülerinden gelen anlamsal bilginin katkısıdır. Şöyle ki sadece metin sorgulandığında ve sorgu terimi en basiti ile “blok” olduğunda bu terim sayfada yer almıyorsa erişim çıktısında yer almaz. Ancak bir kere geçiyor bile olması anahtar tabanlı VSM’de az ya da çok bir ağırlığa sahip olur. Oysa o sayfada yer alan bir “blok” konulu görüntünün de en az metinde bir kere geçmek kadar anlam taşımalarının gerektiği düşünülmektedir. Elde edilen anma grafiğinin de bu fikri destekler görünümde olduğu görülmektedir. Özetle anma değerlerinde gözlenen ilerlemenin sebebi, sayfaların görüntülerden çıkarsanan anlamsal özellikler ile birlikte dizinlenmesidir. Aynı sorgulama biçimleri ile elde edilen  $R_{norm}$  değerlerinin grafiksel gösterimi şekil 5.2’de görülmektedir.



Şekil 5.2:  $R_{norm}$  değeri değişim grafiği (Deney1)

Şekil 5.2'de görüleceği üzere tezde geliştirilen metin tabanlı anlamsal erişim Lucene ile gerçekleştirilen erişimlere göre daha iyi sonuç üretmiştir. Bu ilerlemenin sebebi de anma değerinde oluşan ilerlemenin sebebi ile aynıdır. Yani ontolojide birbiri ile ilişkili terimlerin aynı sayfada geçiyor olması halinde sayfanın anlamsal vurgusu artmış, bu artış ontoloji desteğiyle ele alınmış ve erişim çıktısında daha üst sıralara yerleşmesine sebep olmuştur.

Tezde geliştirilen metin veya görüntü tabanlı anlamsal sorgulamada, diğer ikisine göre genelde bir ilerleme kaydedilmiş, ancak 2. sorguda gerileme elde edildiği gözlemlenmiştir. İkinci sorgu hariç  $R_{norm}$  değerleri üzerinde gözlemlenen ilerlemeyi sağlayan sebep, her iki veri türünden gelen ortak anlamsal özelliklerin yani anlamsal vurgunun doğru ele alınmış olmasıdır. Yani metni ve görüntüsü hem birbiri ile hemde sorgu ile ilgili belgeler erişim çıktısında üst sıralara yerleşmiştir. İkinci sorguda varolan gerilemenin sebebi, görüntülerden çıkarılan yanlış anlamsallıktır. Yani sorgu ile ilgisiz görüntüler sebebiyle aslen ilgisiz sayfalar da erişim çıktısında yer almıştır. Bu durum şekil 5.3'te yer alan duyarlık grafiği üzerinde de gözlemlenmiş ve tartışılmıştır.



Şekil 5.3:Duyarlık değeri değişim grafiği (Deney1)

Şekil 5.3'te görüleceği üzere genel olarak her 3 sorgulama biçiminde de gözlemlenen duyarlık değerleri oldukça yüksektir. Bu durumun temel sebebi

deneyin gerçekleştirildiği “voleybol” alanına ait derlemin kapalı olma özelliğidir. Yani derlemdeki tüm sayfalar “voleybol” alanı ile ilişkilidir. Dolayısı ile duyarlık değeri üzerinde olumsuz etki yapacak faktörler kapalı derlem özelliği sebebi ile elenmiştir. Örneğin duyarlık değerini olumsuz etkileyen temel faktörlerden biri olan eşseslilik durumu, oluşturulan kapalı derlemde mevcut değildir. Bir sorgu ile örneklemek gerekir ise “manşet” terimi, oluşturulan deney derleminde sadece voleybol oyunu içindeki anlamı ile yer alırken, açık bir derlemde haber manşetleri ile eşseslilik özelliği gösterir ve erişim çıktısına haber manşetlerinde girmesi söz konusu olur. Bu durumda erişilen sayfaların duyarlık derecesi olumsuz etkilenir.

Şekil 5.3'te görüleceği üzere tezde geliştirilen metin tabanlı anlamsal erişimde, Lucene ile gerçekleştirilen erişimlere göre 2 noktada ilerleme kaydederken, 4 noktada aynı kaldığı gözlemlenmiştir. Dolayısı ile anlamsal sorgulama ile erişilen belgelerin ilgililik oranında, terimler arası ontolojik ilişkilerin gözetimi sebebi ile artış sağlamıştır. Zira terimler arası ontolojik ilişkilerin dizinlemede kullanılması, anma analizinde açıklandığı üzere, erişim çıktısının uzamasına sebep olmuştur. Erişim çıktısına eklenen ilgili sayfa sayısının ilgisizlere göre daha yüksek olduğu gözlemlenmiştir. Ancak tezde geliştirilen metin veya görüntü tabanlı anlamsal erişimde değişik tepkiler elde edilmiştir. Yani 3 sorgu için gerileme, 1 sorgu için ilerleme ve 2 sorgu için de seviyeyi koruyan bir davranış gözlenmiştir. Bu durumu yaratan etkenin erişim çıktılarının uzaması olduğu düşünülmektedir. Şöyle ki; anma değerini artıran faktör görüntü verilerinden çıkarsanan anlamsallıkların da kullanılmasıdır. Ancak görüntülerin anlamsallığının çıkarılmasında, yanlış anlamsallıkların da çıkarsanarak erişim çıktısına eklendiği duyarlık grafiğinden görülmektedir. Geliştirilen tezde, görüntülerin anlamsallığının çıkarılmasında eşik değeri uygulaması kullanılmamıştır. Bu sebeple sorgu ile küçük bir benzerliği tespit edilen görüntüler de sayfanın erişim çıktısına yerleşmesine sebep olmuştur.

Her üç grafik birlikte yorumlandığında deney 1'e ait değerlendirmeler şu biçimde ifade edilebilir:

- Metin tabanlı anlamsal erişimler: Anma ve  $r_{norm}$  değerlerinin artmasının sebebi, ontolojide ilişkilendirilmiş terimlerin, derlemdeki birlikteliklerinin ele alınması ve her bir terimin ham ağırlığının yanında ontolojik ilişkilerinden kaynaklanan ek ağırlıklarını da kazanmasıdır. Bu yöntem ile sayfada yer

almayan terimler de dizinleme de kullanılmıştır. Bu anlamsal genişleme erişim çıktısını uzatırken, sayfada geçen bir terimin kazandığı ek ağırlık anlamsal vurguyu yansıtmış ve sırlamayı değiştirmiştir.

Duyarlık değerinde ise gözlenen kısmi ilerleme, anlamsal genişleme ile uzayan erişim çıktısında eklenen yeni belgelerin daha çok ilgili belgeler olduğunu göstermektedir.

- Metin veya görüntü tabanlı anlamsal erişimler: Sayfada yer alan görüntülerin de anlamsal dizinlemede kullanılması anma ve  $r_{norm}$  değerlerinin yükselmesini sağlamıştır. Zira metinde yer almayan yeni anlamsal özellikler erişim çıktısını uzatarak anmayı artırmış, metinde ve görüntüde birlikte yer alan anlamsal özellikler anlamsal vurguyu yansıtmış ve erişim çıktısının sıralamasını değiştirmiştir.

Duyarlık değerleri görüntülerden çıkarılan yanlış anlamsal özellikler sebebi ile kısmen düşmüştür. Bu düşüşün sebebi erişim çıktısının uzamasıdır. Erişim çıktısının uzaması anma değerlerinin artışını sağlamaktadır. Dolayısı ile anma da kazanılan toplam ilerlemenin, duyarlılıkta sağlanan toplam gerilemeden daha yüksek olduğu belirtilmelidir. Ayrıca  $r_{norm}$  grafiği ilgisiz belgelerin erişim çıktısının sonunda olduğunu göstermektedir.

Sistemin metin ve görüntüleri birlikte ele alabilmesi, metin ve görüntüler için aynı biçimde ele alınabilen üst veriler üretmesi mantığına dayanmaktadır. Yani görüntülerde terimler ile ifade edilir. Görüntülerin terimlere dönüştürülmesinde alan ontolojisi kullanılır. Ontolojide yapılan eylem ve kavram tanımları kullanılır. Be kesime kadar açıklanan erişimler web sayfasının bütün olarak kullanıldığı erişimlere aittir. İlerleyen kesimde ise sadece görüntüler için tezde geliştirilen anlamsal erişim sisteminin erişim sonuçları yer almaktadır.

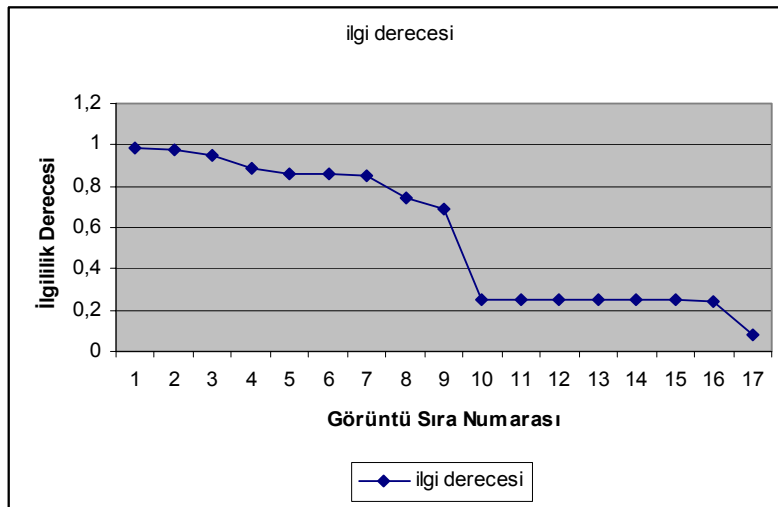
Deney ortamında toplam 45 tane görüntü vardır. Sadece görüntüler üzerinde “smaç” sorgu ile elde edilen erişim çıktısındaki görüntüler, soldan sağa doğru ilgi dereceleri azalan biçimde şekil 5.4’te yer almaktadır.





Şekil 5.4: “Smaç” eylemini içeren görüntüler kümesi

Şekil 5.4’te yer alan görüntülerin ilgi derecelerinin grafik ile gösterimi Şekil 5.5’te yer almaktadır.



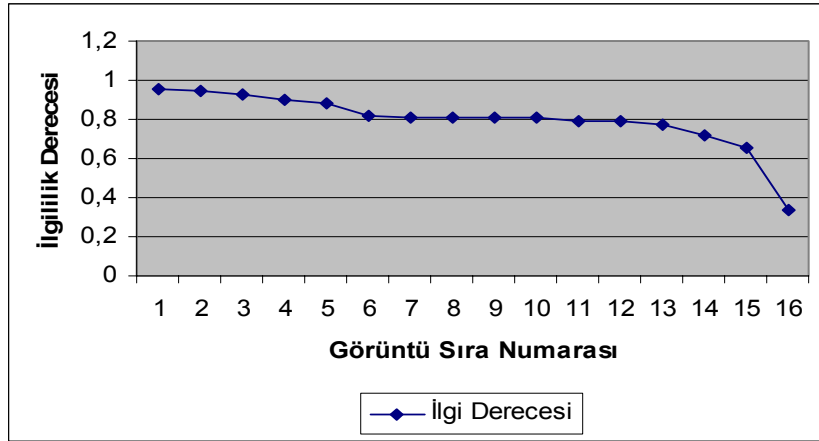
Şekil 5.5: “Smaç” eylemini içeren görüntülerin ilgi dereceleri

Görüleceği üzere 10. görüntüden itibaren ilgi derecelerinde belirgin bir düşme gerçekleşmiştir. Bu düşmenin sebebi görüntüler incelendiğinde fark edilecektir ki 10. görüntüden itibaren ya eylem gerçekleşmiştir. Yani “smaç” vuruşu icra edilmiştir. Ya da “smaç” eyleminin tanımında geçen nesnelere görüntüde yer almamaktadır.

Aynı biçimde “blok” sorgusu ile elde edilen erişim çıktısında yer alan görüntüler şekil 5.6’da ilgi dereceleri azalan sırada yer almaktadır.



Şekil 5.6: “Blok” eylemini içeren görüntüler kümesi



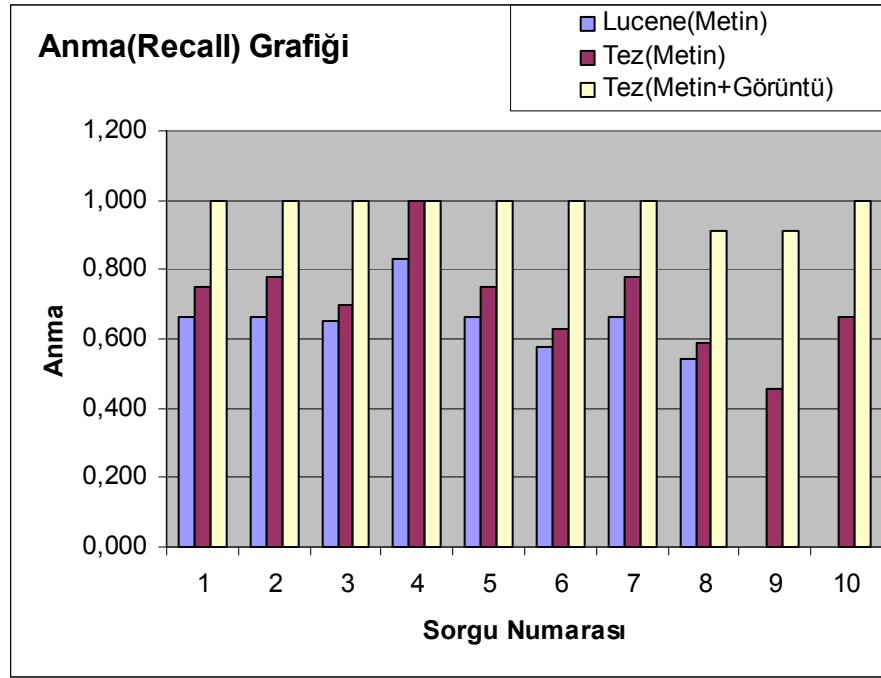
Şekil 5.7: “Blok” eylemini içeren görüntülerin ilgi dereceleri

Görüleceği üzere blok eyleminde üst sıralarda yer alan görüntüler topun blokta olduğu ana aittir. Her iki eylemin tanımı Ek-6’da yer alan “voleybol” alanı ontolojisinde incelenebilir.

Her iki örnekte ontoloji modelinde yer alan eylemlerin nesne ve konumsal ilişkiler ile ifade edilebilmesi fikrini destekler niteliktedir. Yani görüntülerde yer alan eylemlerin çıkarsanma başarısı, eylemlerin tarif edilebilmesi başarısı ile doğru orantılıdır ve eylemlerin bu biçimde çıkarsamanındaki olurluluğunu göstermektedir.

### 5.2.2. Deney 2’ye Ait Başarım Sonuçları

“Gövde” alanı ile ilgili 10 adet sorgu ile gerçekleştirilen erişimlerin anma, duyarlık ve  $r_{norm}$  grafikleri şekil 5.8, şekil 5.9 ve şekil 5.10’da yer almaktadır.

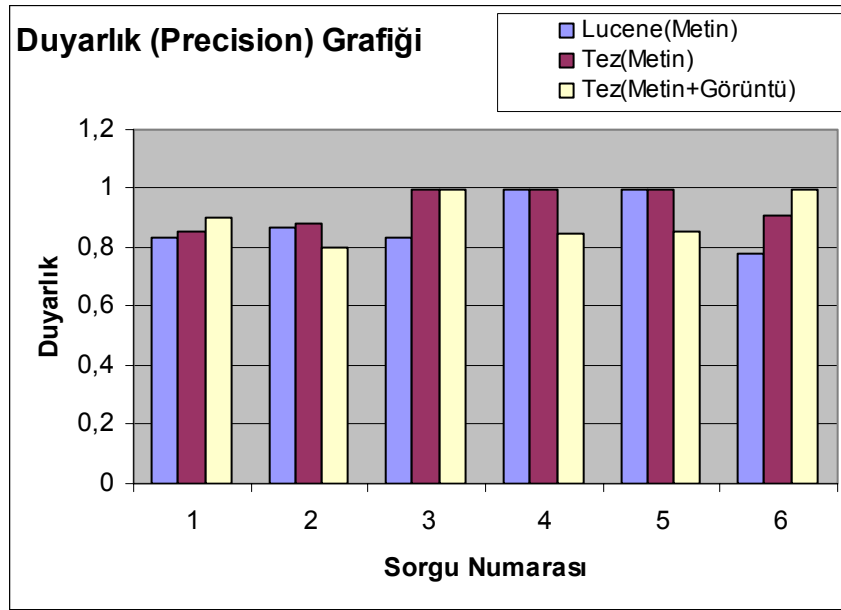


Şekil 5.8: Anma değeri değışim grafiđi (Deney2)

Şekil 5.8’de tezde geliştirilen metin tabanlı anlamsal erişim sisteminin, Lucene’ e göre daha iyi anma değeri ürettiđi görülmektedir. Bu ilerlemenin sebebi, deney 1’de gerçekteşen ilerleme sebebi ile aynı olarak, terimlerin ađırlıklarının ontolojik ilişkiler gözetilerek üretilmesidir. Yani sayfada yer almayan sorgu terimleri, ontolojik ilişkileri sebebiyle sayfa dizinlemesinde kullanılmıştır. Bu anlamsal genişleme erişim çıktısının uzamasını ve anma artışını sağlamıştır. Bu durumun etkisi özellikle 9. ve 10. sorgularda yani kavramlarda gözlemlenebilir. Zira her iki sorguda kavram düzeyinde anlamsallıđa sahip terimlerden oluşmaktadır ve hiçbir biçimde metin üretiminde kullanılmamışlardır. Böyle bir durumda Lucene erişim çıktısı üretemezken, ontolojik ilişkiler sebebi ile metin tabanlı anlamsal sorgulamada orta kalitede bir erişim çıktısı üretilmiştır.

Şekil 5.8’de görüleceđi üzere metin veya görüntü tabanlı anlamsal sorgulamada elde edilen anma değeri, diđer erişimlerden yüksektir. Bu ilerlemenin sebebi deney 1’deki açıklamalar ile aynı olarak, görüntülerden gelen anlamsal özellikler ile sayfanın dizinlenmesidir. Elde edilen anma değeri oldukça yüksektir. Çünkü metin kesiminde kullanılan terimlerin hepsinin bir ve aynı alan terimleri olmaları sebebi ile ontolojiden güçlü bir destek alınmıştır. Fakat asıl sebep görüntülerden çıkarılan anlamsal özelliklerin doğruluđunda yatmaktadır. Şöyle ki; NEK Ontoloji

modelinde eylemler, nesne ve konumsal ilişkiler kullanılarak tanımlanır. Hareket arzeden görüntülerde, konumsal ilişkiler hareketin gerçekleşme süreçlerinde değişimler ya da bir başka deyişle bir ilişkiden diğerine dönüşümlerin gerçekleşmesi mümkündür. Dolayısı ile hareket anına ait görüntüler de konumsal ilişkiler her zaman netlik arzemez. Oysa kullanılan görüntüler bir hastaya ait abdomenin CT kesit görüntüleridir. Dolayısı ile bu görüntülerdeki eylemlerde hareket yoktur. Yani konumsal ilişkiler neredeyse sabittir. Ayrıca görüntüler hep aynı uzaklıktan ve perspektiften çekilmiştir. Yani bakış açısına göre değişen görüntü çekimleri olmadığı için konumsal ilişkilerin değişmesini gerektiren bir başka sebepte ortan kalmıştır. Görüntü çekim mesafesi ve perspektif sabitliği ile zaten durağan olan nesnelere arası konumsal ilişkileri tespit ettiğimizde, eylemlerin doğru çıkarılma oranı en iyiye yaklaşmıştır.

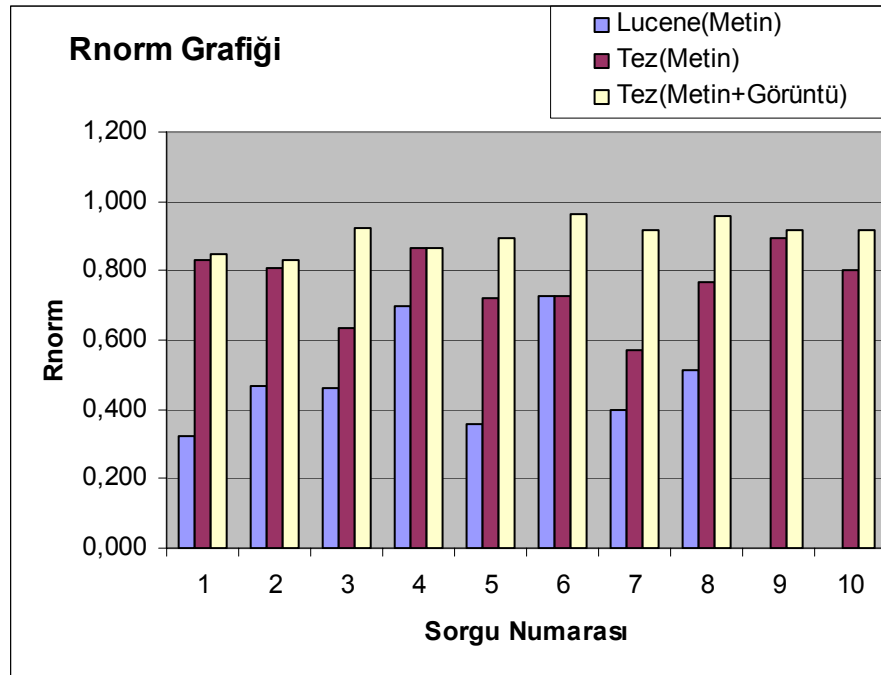


Şekil 5.9: Duyarlık değeri değişim grafiği (Deney2)

Şekil 5.9'da erişimlerin duyarlık değeri görülmektedir. Kapalı derlem olması sebebi ile gerek Lucene ve gerekse metin tabanlı anlamsal erişimde elde edilen duyarlık değeri yüksek ve birbiri ile aynıdır. Anma değeri göz önüne alınırsa, metin tabanlı anlamsal erişimde uzayan erişim çıktısına eklenen yeni sayfaların da sorgular ile ilgili olduğu ve dolayısı ile duyarlıkta bir azalma yaşanmadığı görülmektedir. Aslen eşseslilik gibi anlamsal kaymaların olmadığı kapalı derlemlerde duyarlık değeri daha öncede açıklandığı üzere yüksektir. Metin ve

görüntü tabanlı anlamsal erişimde 3. sorguda, sorgu ile ilgili tüm sayfalara erişildiği anma grafiğinden yorumlanabilirken, ilgisiz sayfalara da erişildiği duyarlık grafiğinde görülmektedir. Duyarlıkta gözlemlenen bu düşüşün sebebi görüntülerden çıkarılan yanlış anlamsallıktır. Ancak görüleceği üzere bu yanlışlık sadece 1 noktada ve çok düşük bir oranda gözlemlenmiştir. Şekil 5.3'te yer alan Deney 1'e ait duyarlık grafiğinde toplam 3 noktada düşüş gerçekleşmiştir. Bu azalma yukarıda bahsi geçtiği üzere abdomen CT görüntülerinin homojenliğine ve abdomen bağlamın konumsal ilişkiler ile tarif edilmeye olan uygunluğu ile açıklanabilir.

Şekil 5.10'da erişimlerin  $r_{norm}$  grafiği görülmektedir. Görüleceği üzere metin tabanlı anlamsal erişimde, Lucene'e göre daha iyi bir sonuç elde edilmiştir. Bu ilerlemenin sebebi, deney 1 ile aynı olmak üzere, ontolojik ilişkilerin terimlerin ağırlıklarına yansıtılması ve bu sayede anlamsal vurgunun ele alınmasıdır. Metin veya görüntü tabanlı anlamsal erişimde elde edilen değerler de diğer iki erişim türünden daha iyidir. Bu iyileşmenin sebebi de, deney 1 ile aynı olmak üzere, görüntülerden gelen anlamsal katkı ile metin ve görüntü ilgililiğinin tespiti yani anlamsal vurgunun ele alınmasıdır.

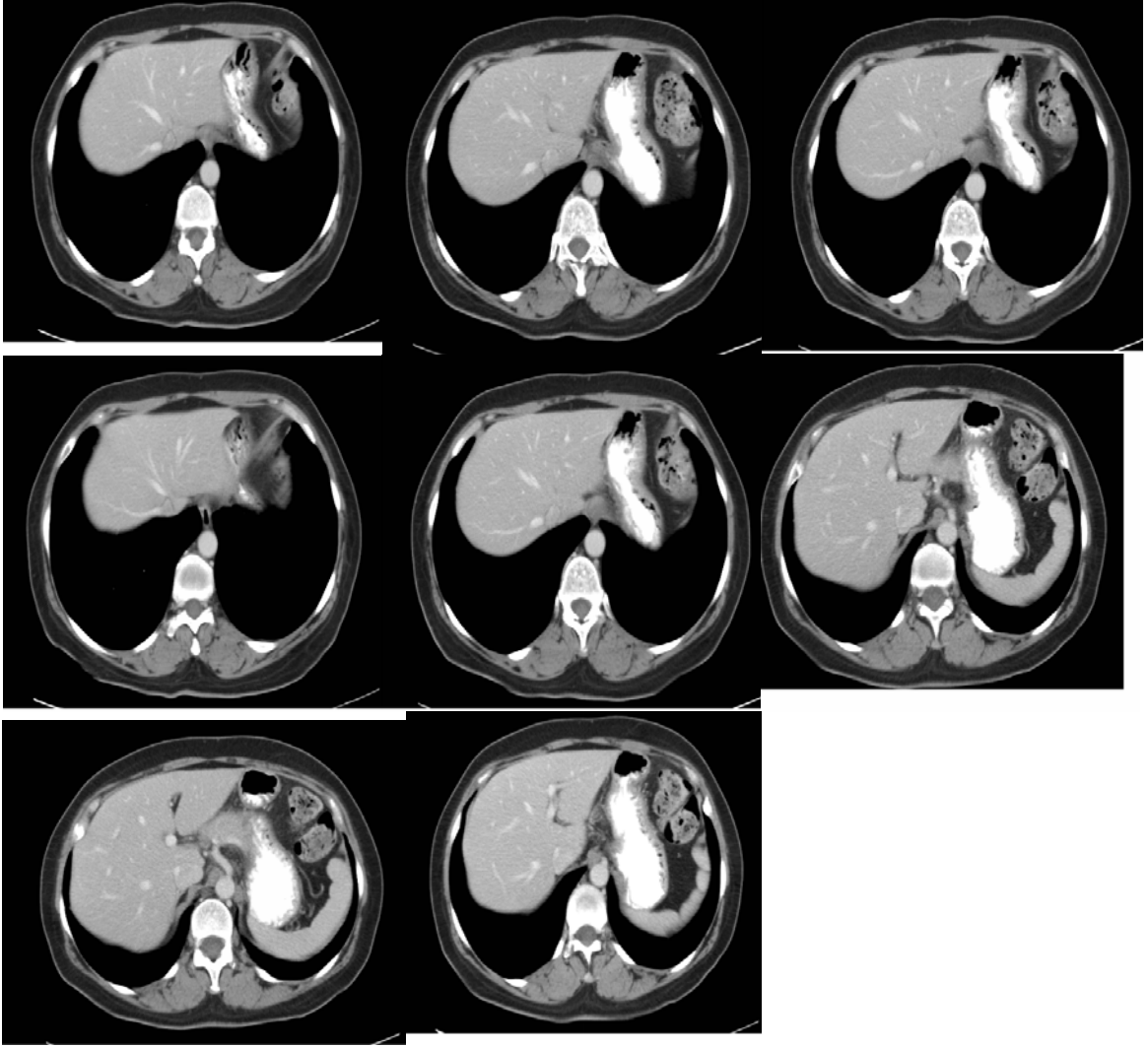


Şekil 5.10:  $R_{norm}$  değeri değişim grafiği (Deney2)

Her üç grafik birlikte yorumlandığında deney 2'ye ait değerlendirmeler şu biçimde ifade edilebilir:

- Metin tabanlı anlamsal erişimler: Anma ve  $r_{norm}$  değerlerinin artmasının sebebi, deney 1 ile aynı olmak üzere ontolojide ilişkilendirilmiş terimlerin, derlemdeki birlikteliklerinin de ele alınmasıdır. Özellikle kavram düzeyinde belirgin bir artış gerçekleşmiştir. Zira Lucene kavram düzeyindeki sorgular için erişim çıktısı üretmezken, ontolojik ilişkiler ile sayfaya ait kavramlar orta kalitede çıkarsanmıştır. Duyarlık değeri derlem özellikleri sebebi ile yüksektir. Anlamsal erişimde uzayan erişim çıktıları, belgelerin ilgililik oranını azaltmamış, yani uzamada ilgili belgeler eklenmiştir.
- Metin veya görüntü tabanlı anlamsal erişimler: Sayfada yer alan görüntülerin de anlamsal dizinlemede kullanılması anma ve  $r_{norm}$  değerlerinin yükselmesini sağlamıştır. Bu yükselmenin sebebi deney 1 ile aynıdır. Ancak deney 2'de kullanılan görüntülerin uzaklık ve perspektif sabitliği yani kalitesi görüntülerin yorumlanma başarımını olumlu yönde açıkça etkilemiştir. Duyarlık değerinde yüksek duyarlık bir nokta hariç korunmuştur. Erişim çıktısı uzamasında yeni belgeler görüntülerinden çıkarılan yanlış anlamsallık sebebi ile eklemiştir. Ancak görüldüğü üzere anma'da elde edilen toplam artış, duyarlıkta yaşanan gerilemeden oldukça yüksektir. Ayrıca  $r_{norm}$  değerlerinde görüldüğü üzere ilgisiz belgeler erişim çıktısının sonundadır.

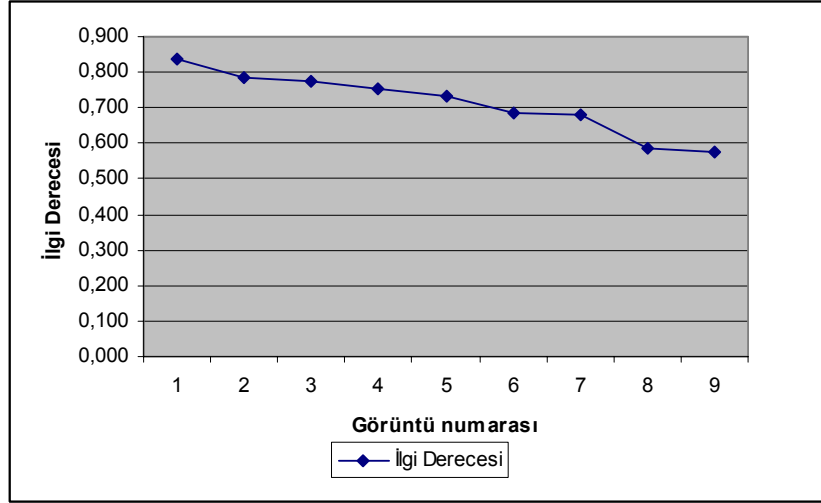
Deney2 ortamında toplam 40 tane görüntü vardır. Deneyde kullanılan görüntü örneklerini teşkil etmesi için "Akciğer" alan terimi sadece görüntüler üzerinde sorgulanmış, elde edilen erişim çıktısındaki görüntüler, soldan sağa doğru ilgi dereceleri azalan biçimde şekil 5.11'de yer almaktadır.



Şekil 5.11: “Akciğer” eylemini içeren görüntü kümesi

Görüntülerin sorgu ile ilgililik derecelerini gösteren grafik şekil 5.12’de görülmektedir. Şekil 5.11’de yer alan görüntüler “toraks”dan başlayıp sıra ile “abdomen”e doğru ilerlemektedir. Dolayısı ile anlamsal bilgi azalan ilgililik derecesine sahip olmalıdır. Bu durum şekil 5.12’deki grafiğe de yansımıştır.





Şekil 5.12: “Akciğer” eylemini içeren görüntüleri ilgi dereceleri

İlerleyen kesimde her iki deneyde elde edilen sonuçlar birbiri ile karşılaştırılmıştır.

### 5.2.3. Deneyler Arası Başarımın Karşılaştırılması

Düzenlenen deneylerde elde edilen başarımların grafikleri birlikte ele alındığında, birbirinden çok farklı iki alan ve iki derlem için elde edilen başarımların çizgilerinde paralellik görülmektedir. Her iki deneyde de anma ve  $r_{norm}$  değerleri artarken uzayan erişim çıktısı sebebiyle duyarlılık değeri kısmen gerilemiştir. Çünkü görüntülerin sorgu terimleri ile benzerliği araştırmasında herhangi bir eşik değeri kullanılmamıştır. Deney 2’de elde edilen başarımların deney 1’e göre yüksek olması iki temel sebebe dayanmaktadır:

1. Deney 2’de üretilen derlem tamamı ile elle üretildiği için alanda yer alan terimler, derlemde kullanılmamıştır. Böylece tezde önerilen anlamsal erişim modelinin somut katkısı bariz bir biçimde grafiklere yansımıştır.
2. Deney 2’de kullanılan görüntüler CT görüntüleri olduğu için her bir görüntü aynı açı ve uzaklıkla elde edilmiştir. Dolayısıyla ilişkilerin doğru tespit edilmesini olumlu etkilemiştir. Ayrıca deney 2’de seçilen alanda yer alan eylemler hareket tabanlı değildir. Dolayısıyla eylemlerin tanımlarının netlik derecesi yüksektir.

## BÖLÜM 6. GENEL SONUÇ VE ÖNERİLER

Geliştirilen tezin ana hedefi farklı veri türlerini bir arada aynı biçimde kullanarak anlamsal sorgulamayı gerçekleştirmektir. Tez kapsamında farklı veri türü olarak birbirini tamamlayan nitelikleri sebebiyle görüntü ve metinler seçilmiştir. Dolayısı ile sistemin hedefi görüntü ve metin veri türlerini aynı sorgu cümlecği ile anlamsal sorgulamak biçimine dönüşmüştür. Anlamsal sorgulama, verilerin içeriklerinden ziyade ifade ettikleri anlamların sorgulanmasıdır. Bu ana hatları ile geliştirilen tez Anlamsal Web konu başlığına girmektedir ve tezde gerçekleştirilen bilgi edinim biçimi Anlamsal Web çizgisine uygundur.

Sistemi geliştirme öngörüsü şu biçimdedir: farklı veri türlerinin birlikte kullanımının belli bir anlamı vurgulamak ya da anlamı zenginleştirmek gibi iki farklı katkısı vardır. Tek bir veri türü ile gerçekleştirilecek erişimin bu iki katkıyı da dışladığı düşünülmüştür. Dolayısı ile iki veri türünü aynı anda kullanarak, bu katkıların erişim başarımı üzerinde olumlu etki yapacağı beklentisi güdülmüştür. Ancak iki veri türünü birlikte kullanabilmek için iki veri türünde aynı sorgu cümlecği ile sorgulanması gereksinimi ortaya çıkar. Bu gereksinim de iki veri türünde aynı tür üst veriler ile ifade edilmesini gerektirir. Tez kapsamında bu gereksinimler karşılanmıştır. Şöyle ki; metin ve görüntülerin ilk etapta çıkarılan içerik özellikleri bir alan ontolojisi ile birlikte anlamsal özelliklerin çıkarımında kullanılmıştır ve çıkarılan anlamsal özellikler terim tabanlıdır. Yani görüntüler de terimlere çevrilmiştir.

Anlamsal çıkarsamada kullanılan alan ontolojisi, bir alanın modelleme biçimine bir öneri getirmektedir. Alan terimleri üç farklı anlamsallık seviyesinde olan nesne, eylem ve kavramlar biçimde ele alınıp, her üç kategori de birbiri ile ilişkilendirilmiştir. Nesnelere alanda yer alan somut nesnelere tür adıdır. Eylemler alandaki an adlarına karşılık gelen, nesne ve konumsal ilişkiler ile tarif edilebilen terimlerdir. Kavramlar ise alandaki üst konu başlıkları gibi düşünülebilir. Yani kavramlar kendi anlamlarında alandaki diğer terimleri içerirler. Bu yapıya bakıldığında alanın durağan yapısı nesne ve kavramlar ile, dinamik yapısı da eylemler ile ele alınmıştır. Bu model kendi içinde herhangi bir alana ilişkin varlık ya

da nitelik içermez ama elde edilen ontolojiler alana bağımlıdır. Bu modele de NEK (nesne, eylem, kavram) Ontoloji modeli adı verilmiştir.

Sistemin başarımını ölçmek için gerçek veri ile “voleybol” ve sentetik veri ile “gövde” alanları seçilmiş ve NEK Ontoloji modeli ile modellenmiştir. Her iki alan için de kapalı derlemeler oluşturulmuştur. Sistemin başarımını anma, duyarlık ve  $R_{norm}$  parametreleri ile ölçülmüştür. Elde edilen sonuçlar kısaca şu biçimdedir:

- Hem salt metin hemde metin veya görüntü tabanlı erişimlerde anma değerlerinde artış gözlemlenmiştir. Bunun sebebi, hem terimler arası ilişkilerin hem de iki farklı veri türünün sayfaya kattığı anlamsal genişlemenin erişim çıktısına yansiyarak uzamasıdır.
- Hem salt metin hemde metin veya görüntü tabanlı erişimlerde  $r_{norm}$  değerlerinde artış gözlemlenmiştir. Bunun sebebi, hem terimler arası ontolojik ilişkilerin hem de iki farklı veri türünün sayfaya kattığı anlamsal vurgunun erişim çıktısına yansımalarıdır.
- Salt metin tabanlı anlamsal erişimde duyarlık değerinde kısmen iyileşme gözlenirken metin veya görüntü tabanlı erişimlerde kısmen bir düşüş elde edilmiştir. Bu düşüşün sebebi ise erişim çıktısının uzamasına sebep olan sorgu ile ilgisiz görüntülerdir. Ancak iki noktayı belirtmekte fayda vardır:
  - Eklenen ilgisiz belgeler erişim listesinin sonunda yer almaktadır. Bu bilgi  $R_{norm}$  değerindeki artış ile açıklanabilir.
  - Sistemin varolan duyarlık değeri oldukça yüksektir. Çünkü kapalı bir derlemde özellikle eşseslilik gibi bir ilişki yoktur. Dolayısı ile sadece anahtar tabanlı bir erişim bile yapılırsa duyarlık yüksek çıkacaktır.

Genel olarak deneylerde elde edilen başarımlara ait yorumlarımız şu biçimdedir:

- Tez kapsamında önerilen NEK Ontoloji modelinin hedefi hem metin hem de görüntü türü verilerin anlamsal özelliklerinin çıkarılmasına hizmet

edebilmesi idi. Elde edilen sonuçlarla, NEK Ontoloji modelinin her iki veri türüne de anlamsal çıkarsama için altyapı oluşturduğu düşünülmektedir.

- Tez öngörüsü “farklı veri türlerinin birlikteliklerinin anlamsal vurgu ve zenginliği oluşturduğu” biçimde idi. Deneylerde elde edilen anma ve  $r_{norm}$  değerleri bu öngörüü desteklemektedir.
- Sistemin anlamsal erişim hedefini de gerçekleştirdiği düşünülmektedir. Şöyle ki; özellikle görüntüler için içerik özelliklerinde yer almayan eylem ve kavram adları sorgulamaya açılmıştır.

Bu çalışmalar ışığında yakın gelecekte yapılmasına ihtiyaç duyulan ilk çalışma konusu görüntüler için çıkarılan anlamsal özelliklerde en uygun eşik değeri ile benzerlik kararının verilmesi üzerine olabilir.

## KAYNAKLAR

1. Altavista, [www.altavista.com](http://www.altavista.com)
2. Yahoo, [www.yahoo.com](http://www.yahoo.com)
3. HotBot, [www.hotbot.com](http://www.hotbot.com)
4. Lycos, [www.lycos.com](http://www.lycos.com)
5. A. Benitez, M. Martinez, H. Rising, Description of a Single Multimedia Document, Coloumbia University, <http://www.ee.columbia.edu/~ana/>,2001
6. G. Amato, F. Rabitti, P. Savino, Supporting Image Search on the Web, The Challenge of Image Retrieval, Newcastle upon Tyne, <http://www.sci.unich.it/~amato/papers/>,1998
7. S. Sclaroff, L. Taycher, M. Cascia, ImageRover: A Content-Based Image Browser for the WWW, Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries, 6/97
8. J.R. Smith, S.F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System", ACM Multimedia, November 20, 1996, <http://www.ctr.columbia.edu/~jrsmith/html/>,1997
9. J.R.Smith, S.F.Chang, An Image and Video Search Engine for the WWW, IS&T/SPIE Proceedings, Storage & Retrieval for Image and Video Databases, <http://www.itmn.colombia.edu/webseek>, 1997
10. J.R.Smith, S.F.Chang, M. Beigi, Visual Information Retrieval from Large Distributed On-Line Repositories, <http://www.ctr.columbia.edu/~jrsmith/html/>,1997
11. J.R. Smith, Integrated Spatial and Feature Systems: Retrieval, Analysis and Compresion, Colombia University, <http://www.ctr.columbia.edu/~jrsmith/html/>, 1997
12. J. R. Smith, Content-Based Visual Query Applications, <http://www.ctr.columbia.edu/~jrsmith/html/>, 1998
13. M.Beigi, A. Benitez, S. Chang, MetaSEEk: A Content-Based Meta Search Engine For Images", [http://www.ctr.columbia.edu/papers\\_advent/98/SPIEjan98\\_ana.pdf](http://www.ctr.columbia.edu/papers_advent/98/SPIEjan98_ana.pdf)
14. J. Hendler, Agents and Semantic Web <http://www.cs.umd.edu/users/hendler/AgentWeb.html>

15. A. Jain, M. Murty, P. Flynn, Data Clustering Review, ACM Computing Surveys, Vol 31, No 3, September 1999
16. S. Nepal, M. Ramakrishna, J. Thom, A Fuzzy System for Content Based Retrieval, Department of Computer Science RMIT University, <http://www.ibmalmaden.com>
17. K.S.Leung, I.King, H.Y. Yue, Fuzzy Clustering Method for Content-Based Indexing, <http://www.cse.cuhk.edu.hk/~ksleung/>
18. I. Gokçen, A.Yazıcı, B. Buckles, Fuzzy Content Based Retrieval in Image Databases, First International conference, ADVIS 2000
19. R. Fagin, Combining Fuzzy Information : An Overview, <http://www.almaden.ibm.com/cs/people/fagin/>, 2002
20. R. Fagin, Fuzzy Queries in Multimedia Database Systems, IBM Almaden Research Center, <http://www.almaden.ibm.com/cs/people/fagin/>,1998
21. D. Dubois, H, Prade, F. Sedes, Some Uses of Fuzzy Logic in Multimedia Databases Querying, <http://www.irit.fr/recherches/RPDMP/persos/Didier.html>
22. F.Masulli, M. Artuso, P.Bogus, A. Schenone, Fuzzy Clustering Method for the Segmentation of Multivariate Medical Images, <http://citeseer.nj.nec.com/cache/papers/cs/26475>
23. D. Joyce, P. Lewis, R. Tansley, M. Dobie, W, Hall, Semiotic and Agent for Integrating and Navigating Through Multimedia Representation of Concepts, Multimedia Research Group, University of Southampton, <http://www.citeseer.nj.nec.com/cs,2002>
24. R. Srihari, Z. Zhang, A. Rao, Intelligent Indexing and Semantic Retrieval of Multimodal Documents, Information Retrieval, Nov 1999
25. T. Minka, An Image Database Browser that Learns From User Interaction, MIT Media Laboratory Technical Report 365, <http://www.stat.cmu.edu/~minka/>, 1996
26. N. Guarino, OntoSeek: Content-Based Access to the Web, IEEE Intelligent Systems, May/June1999
27. H. Müller, W. Müller, A Web Based Evaluation System for CBIR, <http://cui.unige.ch/~vision/members/HenningMueller.html>, 2001
28. W. Y. Ma, B.S. Manjunath, NeTra: A Toolbox for Navigating Large Image Databases, Multimedia Systems, Springer-Verlag, 1999

29. Y.A. Aslandođan, C. T. Yu, Evaluating Strategies and Systems for Content Based Indexing of Persons Images on the Web”, ACM Multimedia, <http://www.cs.uic.edu/~yu/>, 2001
30. Y. Du, J. Wang, A Scalable Integrated Region- Based Image Retrieval System, <http://www-db.stanford.edu/~wangz/project/imsearch/SIMPLcity/ICIP2001/wang.pdf>
31. J. Vendring, M. Worrying, A.W.M. smeulders, Filter Image Browsing, Exploiting Interaction in Image Retrieval, <http://carol.wins.uva.nl/~vendrig/publications.html>
32. G. Sheikholeslami, W. Chang, A. Zhang, SemQuery: Semantic Clustering and Querying on Heterogeneous Features for Visual Data, <http://citeseer.nj.nec.com/sheikholeslami98semquery.html> ,1998
33. C. Carson, V, E, Ogle, Storage and Retrieval of Feature Data for a Very Large Online Image Collection”, Data Engineering, Vol 19, No.4, December 1996
34. R. Veltkamp, M. Tanase, Content-Based Image Retrieval Systems: Survey, Technical Report UU-CS-2000-24, Revised 2001, <http://www.aalab.cs.uu.nl/cbirsurvey/cbir-survey/>
35. C.P. Town, D. Sinclair, Content Based Image Retrieval Using Semantic Visual Categories, AT&T Laboratories, <http://www.doc.ic.ac.uk/~mjp3/phd/literature.html>
36. S. K. Cheung, Efficient Content-Based Image Retrieval in a Distrubuted Environment, Depertment of Computer Science and Engineering The Chinese University, <http://www.cse.cuhk.edu.hk/~kcsia/>
37. N. Bujemaa, C. Nastar, Content Based Image Retrieval at the IMEDIA Group of INRIA, <http://www.rocq.inria.fr/imedia>,1999
38. V. Roth, Distrubuted Image Indexing and Retrieval with Mobile Agents, <http://www.jseal2.com/documents.html>,2000
39. R. Tansley, C. Bird, W. Hall, Automating the Linking of Content and Concept, IBM UK Laboratories, Hursley Park, <http://www.bib.ecs.soton.ac.uk/data/5046/html/tansley.html>, 2000
40. A. Gutman, R-Trees, A Dynamic Index Structure For Spatial Searching, <http://public.rz.fh-wolfenbuettel.de/~hoepnef/bib/author/GUTTMAN-A.html> ,1984

41. T. Sellis, N. Roussopoulos, C. Faloutsos, Department of Computer Science University of Maryland, <http://www.dbnet.ece.ntua.gr/~timos/> ,1987
42. G. Cha, C. Chung, A New Indexing Scheme for Content- Based Image Retrieval, *Multimedia Tools and Applications*, 6, 263-288, 1998
43. G. R. Hjaltson, H. Smaet, Incremental Similarity Search in Multimedia Databases, Center for Automation Research, University of Maryland, <http://db.uwaterloo.ca/~gisli/> , 2000
44. H. Kosch, MPEG-7 and Multimedia Databases Systems, University of Klagenfurt, <http://www-itec.uni-klu.ac.at/~harald/>,2000
45. P. Apers, M. Kersten, Content-Based Retrieval in Multimedia Databases Based on Feature Models, University of Twente, <http://www.utwente.nl/~apers>, 2000
46. J. Melton, A. Eisenberg, SQL Multimedia and Application Packages (SQL/MM), <http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/m/Melton:Jim.html>, 2001
47. C. U. Orji, K. Nwosu, Multimedia Object Storage and Retrieval, AT&T Bell Labs, <http://www.nwosu.net/> ,1996
48. Y. Niu, M.T. Özsu, X. Li, A Study of Image Indexing Techniques for Multimedia Database Systems, University of Alberta, Technical Report TR 95-19, <http://db.uwaterloo.ca/~tozsu/>, 1995
49. V. Oria, P. J. Iglinski, M.T. Özsu, A Frame work for Multimedia Database Systems, <http://db.uwaterloo.ca/~tozsu/>,1998
50. I. L. Cheng, Image Databases: A Content-Based Type System and Query By Similarity Match, University of Alberta, Technical Report 99-03, <http://darwell.uwaterloo.ca/~ddbms/projects/multimedia/reports.html>, 1999
51. Y. Zhuang, S. Mehrotra, T.S. Huang, A Multimedia Information Retrieval Model Based on Semantic and Visual Content,, University of Illinois, <http://ai.zju.edu.cn/~yzhuang/>, 1998
52. S. Dumais, H. Chen, Hierarchical Classification of Web Content, Microsoft Research, <http://research.microsoft.com/~sdumais/>, 2000
53. M. Ortega, K. Porkaew, S. Mehrotra, Information Retrieval over Multimedia Documents, Universtiy of Illinois, <http://www.ics.uci.edu/~miki/professional/publications.shtml>, 1998



54. M. Ortega, S. Mehrotra, Y. Rui, T.S. Huang, A Relevance Feedback Architecture for Content-Based Multimedia Information Retrieval System, Universtiy of Illinois, <http://www.ics.uci.edu/~miki/professional/publications.shtml>, 1998
55. M. Ortega, S. Mehrotra, WebMARS: A Multimedia Search Engine for Full Document Retrieval and Cross Media Browsing, <http://citeseer.nj.nec.com/325962.html>, 2001
56. Advanced Multimedia Oriented Retrieval Engine, <http://www.ccrl.neclab.com/amore/index.html>
57. M.Windhouwer, A. Schmidt, M. Kersten, Acoi: A System for Indexing Multimedia Objects, <http://citeseer.nj.nec.com/windhouwer99acoi.html>, 1999
58. Y. Zhuang, S.Mehtaro, T. Huang, A Multimedia Information Retrieval Model Based on Semantic and Visual Content, University of Illinois, <http://citeseer.nj.nec.com/zhuang99multimedia.html>, 1999
59. A. Benitez, S. Chang, J.R.Smith, IMKA: A Multimedia Organization System Combining Perceptual and Semantic Knowledge, <http://citeseer.nj.nec.com/536748.html> , 2000
60. J. Melton, A. Eisenberg, SQL Multimedia and Application Packages (SQL/MM), <http://www.acm.org/sigmod/record/issues/0112/standards.pdf> , 2001
61. M Flickner, Query by Image and Video Content: The QBIC Ssystem, IEEE Computer 28, 9, 1995
62. V. Ogle, Chabot : Retrieval from a Relational Database of Images, IEEE Computer, 28(2), 49-56, 1995
63. D. Florescu, A. Levy, A. Mendelzon, Database Techniques for the World Wide Web: A Survey, ACM SIGMOD Record, 27(3), 59-74, 1998
64. Oracle Visual Information Retrieval User's Guide and Reference , Oracle 8i On-Line Documentation CD-ROM
65. Oracle *interMedia* Audio, Image, and Video User's Guide and Reference, Oracle 8i On-Line Documentation CD-ROM
66. <http://www.software.ibm.com/data/db2/extenders>
67. [www.virage.com](http://www.virage.com)
68. <http://www.microsoft.com/sql/default.asp>

69. V. R. Benjamins, D. Fensel, A. Perez, "(KA)<sup>2</sup>: Bulding Ontologies for the Internet", <http://citeseer.nj.nec.com/276747.html> ,1999
70. E. Kendall, M. Dutra, D. McGinness, "Towards A Commercial Ontology Development Environment", [http://iswc2002.semanticweb.org/posters/mcguinness\\_a4.pdf](http://iswc2002.semanticweb.org/posters/mcguinness_a4.pdf), 2000
71. R. Filkorn, "Semantic Web: Web's Next Generation", Slovak University, <http://www.dcs.elf.stuba.sk/~filkorn/docs/SemanticWeb.pdf>, 2003
72. A. Maedche,"Tutorial: Development and Applications of Ontologies", [http://events.aifb.uni-karlsruhe.de/ontologytutorial/fgml2000/tutorial\\_fgml.pdf](http://events.aifb.uni-karlsruhe.de/ontologytutorial/fgml2000/tutorial_fgml.pdf), 2000
73. S. Cranefield, "UML for Ontology Development", University of Otago, <http://ubot.lockheedmartin.com/ubot/papers/publication/KER4.doc>, 2002
74. S. Cranefield, M. Purvis, "Generating ontology-specific content languages", <http://citeseer.nj.nec.com/cranefield01generating.html> ,2001
75. S.Cranefield, "UML and the Semantic Web", University of Otago, <http://www.semanticweb.org/SWWS/program/full/paper1.pdf>,2001
76. S. Cranefield. M. Purvis. "UML as an Ontology Modelling Language", <http://citeseer.nj.nec.com/cranefield99uml.html>, 1999
77. K. Backlawski, M. Kokar, P Kagut, "Extending UML to Support Ontology Engineering for the Semantic Web", <http://ubot.lockheedmartin.com/ubot/papers/publication/UMLOntology.pdf>, 1999
78. K. Falkovych, M Sabou, H. Stuckenschmidt, "UML for the Semantic Web: Transformation based Approaches" [http://homepages.cwi.nl/~media/publications/UML\\_for\\_SW.pdf](http://homepages.cwi.nl/~media/publications/UML_for_SW.pdf),2002
79. A. Felfering, G. Friedrich, D, Jannach, "UML as Knowledge Acquisition Fronted for Semantic Web Configuration Knowledge Bases" <http://www.soi.city.ac.uk/~msch/conf/ruleml/felfernig.pdf>,2001
80. J. Conallen, "Modelling Web Application Design with UML", [www.rational.com/products/whitepapers/100462.jsp?SMSESSION=NO](http://www.rational.com/products/whitepapers/100462.jsp?SMSESSION=NO), 1998,
81. "UBOT: DAML/UML Based Ontology", [http://ubot.lockheedmaertin.com/ubot/details/uml\\_to\\_daml.html](http://ubot.lockheedmaertin.com/ubot/details/uml_to_daml.html)

82. S. Boll, W. Klas, A Sheth, "Overview on Using Metadata to Manange Multimedia Data", <http://www.informatik.uni-ulm.de/dbis/MMBook/ChapterOne>, 1998
83. D. Fensel, S. Decker, M. Erdmann, "OntoBroker: How to Make the WWW Intelligent", <http://citeseer.nj.nec.com/425424.html> ,1998
84. D. Fensel, S. Decker, M. Erdmann, "OntoBroker: Ontology Based Access to Distrubuted and Semi-Structured Information", [http://www.ira.uka.de/l3v\\_HOME/ARCHIV/JB1999/JB\\_LIT/16511435.htm](http://www.ira.uka.de/l3v_HOME/ARCHIV/JB1999/JB_LIT/16511435.htm) 1998
85. S. Decker, F. Harmelen, J. Broekstra, "The Semnatic Web – on the respective Roles of XML and RDF", <http://www.ontoknowledge.org/oil/download/IEEE00.pdf> ,2000
86. C. Meghini, F. Sebastiani, "MIRLOG: A Logic for Multimedia Information Retrieval", <http://faure.iei.pi.cnr.it/~fabrizio/Publications/LIR98.pdf> 1998
87. C. Meghini, F. Sebastiani, "Modelling the Retrieval of Structured Documents Containing Text and Images", <http://citeseer.nj.nec.com/meghini97modelling.html> ,1997
88. C. Meghini, F. Sebastiani, "A Model Multimedia Information Retrieval" <http://citeseer.nj.nec.com/meghini98model.html> ,1998
89. U. Straccia, "Reasoning with in Fuzzy Description Logics", Journal of Artificial Intelligence Research", Cilt : 4, Sayfa 137-166, 2001
90. V. Gudivada, V. Raghavan, "Modelling and Retrieving Images by Content", <http://citeseer.nj.nec.com/meghini97modelling.html> ,1997
91. F. Creastani, M. Lalmas, "Logic and Uncertainty in Information Retrieval", <http://citeseer.nj.nec.com/541289.html> ,2001
92. S. Vigna, "ERW:Entities and Relations on the Web", <http://www2002.org/CDROM/poster/166>, 2000
93. "Resource Description Framework (RDF) Model and Syntax Specification", <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/> ,1999
94. "Reference Description of the DAML+OIL Ontology Markup Language", <http://www.daml.org/2001/03/reference.html>, 2001
95. "Object Constarint Language", <http://doc.ic.tsu.ru/xml/dox/sections/analysis+design/uml/dox.xml>, 1997

96. F. M. El-Hadidy, H. Poot, "Multimedia Information Retrieval Framework" <http://monetdb.cwi.nl/acoi/DMW/publications/ds99.pdf> , 1998
97. G. Amato, G. Mainetto, P. Savino, "An Approach to a Content based Retrieval of Multimedia Data", *Multimedia Tools and Applications*, Cilt: 7, Sayfa: 9- 36, 1998
98. Foundation for Intelligent Physical Agents," FIPA ACL Message Representation ", <http://www.fipa.org/specs/fipa00070> , 2000
99. K. Barnard, D. Forsyth, Learning the Semantics of Words and Pictures, Computer Devision, Univesity of Berkeley, <http://citeseer.nj.nec.com/barnard00learning.html>, 2001
100. L. Khan, L. Wang, Automatic Ontology Derivation Using Clustering for Images Classification, Department of Computer Science , Texas University, [www.utdallas.edu/~lkhan/papers/AODUCIC\\_Proc8thMIS2002.pdf](http://www.utdallas.edu/~lkhan/papers/AODUCIC_Proc8thMIS2002.pdf) ,2003
101. J. Wang, J.Li, G. Wiederhold, SimplyCity: Semantic-Sensitive Integrated matching for Picture Libraries, *IEEE Transactions on Pattern Analysis and machine Learning*, Cilt.23, Sayı.9, [www-db.stanford.edu/~wangz/project/imsearch/ALIP/ICIP2003/wang.pdf](http://www-db.stanford.edu/~wangz/project/imsearch/ALIP/ICIP2003/wang.pdf) Eylül 2003.
102. H. Zhuge, Semantic Based Web Image Retrieval, China Knowledge Grid Research Group, Chinese Acedemy of Sciences, <http://www2003.org/cdrom/papers/poster/p172/p172z-huge/p172-zhuge.html>, 2003.
103. P.Stanchev, General Image Database Model, Institute of Mathematics and Computer Science, Bulgarian Academy of Sciences, [citeseer.nj.nec.com/stanchev99general.html](http://citeseer.nj.nec.com/stanchev99general.html), 1997
104. Y. Chen, J. Wang, A Region Based Fuzzy Feature Matching Approach to Content Based Image Retrieval, Departmen of Computer Science and Enginerring, Pennsylvania State University, [csdl.computer.org/comp/trans/tp/2002/09/i1252abs.htm](http://csdl.computer.org/comp/trans/tp/2002/09/i1252abs.htm), 2002
105. E.D.Siciascio, G. Mingolla, M. Mongiello, Content Based Image Retrieval over the Web using Query Sketch and Relevance FeedBack, [www-ictserv.poliba.it/disciascio/papers/visual99.pdf](http://www-ictserv.poliba.it/disciascio/papers/visual99.pdf), 1999
106. C. Breen, L. Khan, A. Ponnusamy, Image Classification Using Neural Network and Ontologies, Department of Computer Science, University of

- Texas,  
[www.utdallas.edu/~lkhan/papers/ICUNNO\\_IEEEDEXAFrance2002.pdf](http://www.utdallas.edu/~lkhan/papers/ICUNNO_IEEEDEXAFrance2002.pdf), 2002
107. C. Breen, L. Khan, A. Kumar, L. Wang, Ontology Based Image Classification Using Neural Networks, Department of Computer Science, University of Texas, [www.utdallas.edu/~lkhan/papers/OICUNN\\_SPIE2002.pdf](http://www.utdallas.edu/~lkhan/papers/OICUNN_SPIE2002.pdf), 2002.
  108. J. Li, J. Wang, Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach, IEEE Transactions on Pattern Analysis and Machine Learning, Cilt 25. Sayı 10, [csdl.computer.org/comp/trans/tp/2003/09/i1075abs.htm](http://csdl.computer.org/comp/trans/tp/2003/09/i1075abs.htm), Ekim 2003.
  109. S. Kulkarni, B.Verma, An Intelligent Hybrid Approach for Content Based Image Retrieval, International Journal of Computational Intelligence and Applications, Cilt 2 Sayı 2 Sayfa 173-184, 2002
  110. K. Barnard, P. Duygulu, D. Forsyth, Clustering Art, Computer Division of California, Berkeley, [kobus.ca/research/publications/CVPR-01/clustering-art.pdf](http://kobus.ca/research/publications/CVPR-01/clustering-art.pdf) , 2001.
  111. K. Barnard, N. Shirahatti, A Method for Comparing Content Based Image Retrieval Methods, Department of Computer Science, University of Arizona, [vision.cs.arizona.edu/kobus/research/publications/SPIE-03-bench/](http://vision.cs.arizona.edu/kobus/research/publications/SPIE-03-bench/) , 2002
  112. Y. Chen, J. Wang, R. Krovetz, Content Based Image Retrieval by Clustering, [www-db.stanford.edu/~wangz/project/imsearch/SIMPLicity/ACM03/chen.pdf](http://www-db.stanford.edu/~wangz/project/imsearch/SIMPLicity/ACM03/chen.pdf) , 2003.
  113. Y. Zhuang, X. Liu, Y. Pan, Apply Semantic Template to Support Content Based Image Retrieval, Institute of Artificial Intelligence, ZheJiang University, Proceeding of IS&T and SPIE Storage and Retrieval for Media Databases 2000, [www.andrew.cmu.edu/~xiaoming/publication/SPIE2000.pdf](http://www.andrew.cmu.edu/~xiaoming/publication/SPIE2000.pdf) , Haziran 2000.
  114. Q. Iqbal,J.K. Aggarwal, Feature Integration Multi Image Queries and Relevance Feedback in Image Retrieval, Computer and Vision Research Center Department of Electrical and Computer Engineering, University of Texas, International Conference on Visual Information Systems, Miami, Florida, [amazon.ece.utexas.edu/~qasim/research.htm](http://amazon.ece.utexas.edu/~qasim/research.htm) , Eylül 2003
  115. Q. Iqbal, J.K. Aggarwal, CIRES: A System for Content Based Retrieval in Digital Image Libraries, Computer and Vision Research Center Department of Electrical and Computer Engineering, University of Texas, Seveth

- International Conference on Control, Automation, Robotics and Vision (ICARCV'02), [amazon.ece.utexas.edu/~qasim/papers/iqbal\\_icarvcv02.pdf](http://amazon.ece.utexas.edu/~qasim/papers/iqbal_icarvcv02.pdf) , Aralık 2002
116. Q. Iqbal, J.K. Aggarwal, Retrieval by Classification of Images Containing Large Manmade Objects Using Perceptual Grouping, Computer and Vision Research Center Department of Electrical and Computer Engineering, University of Texas, [citeseer.nj.nec.com/456932.html](http://citeseer.nj.nec.com/456932.html) , 2000
  117. J.K. Aggarwal, Content Based Image Retrieval Via Structure, Color and Texture, Computer and Vision Research Center Department of Electrical and Computer Engineering, University of Texas, [www.calit2.net/events/2002/8-6-aggarwal.html](http://www.calit2.net/events/2002/8-6-aggarwal.html) , 2003
  118. Z. Jin, I. King, X.Li, Content Based Image Retrieval By Relevance Feedback, Nanjing University of Science and Technology, [www.cse.cuhk.edu.hk/~king/PUB/visual2000\\_2.pdf](http://www.cse.cuhk.edu.hk/~king/PUB/visual2000_2.pdf) , 2000
  119. Y.A. Aslandoğan, C.T. Yu, Automatic Feedback for Content Based Image Retrieval on the Web, Department of Computer Science and Engineering, University of Texas [ranger.uta.edu/~alp/publications/aslandoganICME2002.pdf](http://ranger.uta.edu/~alp/publications/aslandoganICME2002.pdf) , 2001
  120. S. Santini, A. Gupta, R. Jain, Emergent Semantics Through Interaction in Image Databases, Visual Computing Laboratory, University of California, [csdl.computer.org/comp/trans/tk/2001/03/k0337abs.htm](http://csdl.computer.org/comp/trans/tk/2001/03/k0337abs.htm) , 2001
  121. CHARIOT: Similarity Search in Large Image Databases, [dbshorus.ethz.ch/Chariot/](http://dbshorus.ethz.ch/Chariot/), 2003.
  122. H. Takeda, <http://www-kasm.nii.ac.jp/~takeda/work2/sw01b/> National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, JAPAN
  123. <http://www.w3.org/RDF/>
  124. <http://protege.stanford.edu/plugins/owl/>
  125. Gruber TR(1993), A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition 5(2):199-220
  126. L. Khan, D. McLeod, E. Hovy, "Retrieval Effectiveness of an Ontology-based model for information selection", The VLDB journal, No: 13, syf: 71-85, 2004
  127. M. Nieto, An Overview of Ontologies, Technical Report, Universidad Las Americas Puebla, Mart 2003

128. P. A.Gomez, Knowledge Sharing and Reuse In The Handbook on Applied Expert Systems, ED CRC yayınevi, 1998
129. OntoWeb Technical Report, Deliverable 1.3: A Survey on Ontology Tools, 2002 Mayıs, [http://ontoweb.aifb.unikarlsruhe.de/About/Deliverables/D13\\_v1-0.zip](http://ontoweb.aifb.unikarlsruhe.de/About/Deliverables/D13_v1-0.zip)
130. S. Akyokuş, A. Güven, Anlamsal Web, Anlamsal Web Dilleri ve Araçları, Bilişim Kurultayı 03, [http://kurultay.tbd.org.tr/kurultay20/Bildiriler/Selim\\_Akyokus/bildiri.pdf](http://kurultay.tbd.org.tr/kurultay20/Bildiriler/Selim_Akyokus/bildiri.pdf) , 2003
131. T. Berners-Lee, Semantic Web Architecture, XML 2000 Conference, <http://www.w3c.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>, 2000
132. SemanticWeb.org, Markup Languages and Ontologies, <http://www.semanticweb.org/knowmarkup.html>, 2003
133. W3C, RDF Vocabulary Description Language 1.0: RDF Schema, <http://www.w3.org/TR/2002/WD-rdf-schema-20021112/>, 2002
134. F. Manola, E. Miller, RDF Primer, W3C Working Draft, Last Update <http://www.w3c.org/TR/rdf-primer/>, 2003
135. W3C, Web Ontology Language Guide, <http://www.w3.org/TR/owl-guide/>, 2004
136. W3C, A Query Language for RDF, <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/> , 2004
137. V. Sugmaran, V.C. Storey, Ontologies for Conceptual Modelling: Their Creation, Use and Management, Data Knowledge Engineering sayı 42, syf. 251-271, 2002
138. L. Khan, D. McLeod, E. Hovy, Retrieval Effectiveness of an Ontology-Based Model for Information Selection, The VLDB Journal, sayı 13, syf. 71-85, 2004.
139. C. Lagoze, J. Hunter, The ABC Ontology Model, <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Lagoze/lagoze-final.pdf>, 2002.
140. D. Vallet, F. Miriam, P. Castells, An Ontology Based Information Retrieval Model, Madrid University, <http://nets.ii.uam.es/~search/publications/eswc05.pdf>, 2005
141. F. Song, W. Zhang, W. Xiao, G. Li, Z. Xu, Ontology-Based Information Retrieval Model for the Semantic Web, The National University of Defense Technology, China, 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) , syf. 152-155, 2005

142. E. Hyvönen, S. Saarela, A. Styrman, K. Viljanen, Ontology Based Image Retrieval, Helsinki University, Finland, <http://www2003.org/cdrom/papers/poster/p199/p199-hyvonen.html>, 2003
143. U. Shah, T. Finin, A. Joshi, R. Cost, J. Mayfield, Information Retrieval on the Semantic Web, Maryland University USA, <http://www.cs.umbc.edu/~finin//papers/cikm02/cikm02.pdf>, 2002. (DARPA destekli, proje no:F30602-97-1-0215)
144. OntoWeb Consortium, A Survey on Ontology Based Applications, <http://ontoweb.org/Members/ruben/Deliverable%201.6>, 2004.
145. D. Widiantoro, J. Yen, A Fuzzy Ontology Based Abstract Search Engine and Its User Studies, Texas A&M University, Pennsylvania University, <http://ist.psu.edu/yen/publications/fuzzieee01.pdf>, 2001.
146. D. Parry, A Fuzzy Ontology For Medical Document Retrieval, Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation , Sayı 32, Syf. 121-126, 2004.
147. L. Chang-Shing, J. Zhi-Wei, A Fuzzy Ontology and Its Application to News Summarization,(SCI) IEEE Transactions on Systems, Man and Cybernetics Part B, sayı. 35, no. 5, syf. 859-880, 2005.
148. Harmalen A., A Sematic web Primer, The MIT Press, 2004.
149. [www.semanticweb.org](http://www.semanticweb.org)
150. [www.mozilla.org](http://www.mozilla.org)
151. <http://hpl.hpcom/semweb/jena-top.html>
152. A. AlpKoçak, A.Kut , E. Özkarahan, Bilgi Bulma Sistemleri için Otomatik Türkçe Dizinleme Yöntemi, Bilişim Bildirileri,s:247-253, 1995
153. M.J. Egenhofer, J.R. Herring, A Mathematical Framework for the Definition of Topological Relationships, Proceedings 4th. int. Symp on Spatial Data handling, syf. 803-813, 2004
154. M. Vazirgiannis, Uncertainty Handling in Spatial Relationships, SAC'00, Italy, ACM num: 1-58113-239-5/00/003
155. P. Huang, C. Lee, Image Database Design Based on 9D-SPA Representation for Spatial Relations, IEEE Transactions on Knowledge and Data Engineering Vol 16. No:12, 2004



156. E. Sezer, "Smart Bilgi Eriřim Sistemi'nin Trke Yerelleřtirilmesi ve Otomatik Gm retimi" H. . Bilgisayar Mhendislięi, Yksek Lisans Tezi, 1999.
157. The Apache Software Foundation, <http://lucene.apache.org/java/docs>.

## EK-1.a. OntolojiEditörü(OE) Nesne Tanımlama Arayüzü

Ontoloji Yardım

Nesne Eylem Kavram

[ Nesneler ]

- top
- el1
- el2
- direk
- baş
- insan**
- sahadışı
- ayak
- baş1
- yanbantlar
- gövde
- sandaşe
- gövde1
- bayrak
- anten
- tribün
- el
- file
- düdük
- saha

[ Adı ]

insan

[ Açıklama ]

voleybolcu, hakem, antrenör olarak görev alabilir

Kaydet

EşAnlam

[ Nesne Parçaları ]

- ayak
- el
- el2
- el1
- gövde
- baş

E S G

## EK-1.b. OntolojiEditörü(OE) Eylem Tanımlama Arayüzü

Ontoloji Yardım

Nesne Eylem Kavram

[ Eylemler ]

- elyanda
- düzvuruş
- bayrakhavada
- fileyeTemas
- smaç
- servis
- ayakçizgide
- manşet
- ayaklavuruş
- elhavada
- topyeredüştü
- blok

[ Adı ]

smaç0

[ Açıklama ]

Saldırı amaçlı vuruş türüdür.

Kaydet

EşAnlam

[ Eylem Tanımları ]

Nesne	Konumsal İlişki	Nesne
top	kısmençinde; üstünde; yanında;	el
el	üstünde;	gövde
el	üstünde;	file
gövde	hizasında; yakın; üstünde;	file

E S ParçaBütün Öncülleri Derecesi E S

[ Eylem Gerçekleştirim ]

Yer	Zaman	Aktor
ankara	01.01.2004	Öznur

Ekle Sil

[ Eylemde Tanımlı Roller ]

Terim Adı	Rolü
insan	smaçör

Ekle Sil

## EK-1.c. OntolojiEditörü(OE) Kavram Tanımlama Arayüzü

Ontoloji Yardım

Nesne Eylem Kavram

[ Kavramlar ]

- vuruş
- skor
- sayı
- kollektifBilinç
- takımsporu
- set
- voleybol**
- maç
- hücum
- savunma
- aykırıDurum

[ Adı ]

voleybol

[ Açıklama ]

Takım sporu türüdür.

Kaydet EşAnlam

[ Kavram Bileşenleri ]

Ontoloji Terimi	İlişki Derecesi
el	1.0
hücum	1.0
aykırıDurum	1.0
vuruş	1.0
maç	1.0
saha	1.0
top	1.0
file	0.8
savunma	1.0

Ekle Sil

[ Kavram Gerçekleştirim ]

Yer	Zaman	Aktor
Ankara	11.03.2005	PolisAkademi...
Ankara	25.01.2005	Halk Bankası-...

Ekle Sil

[ Kavramda Tanımlı Roller ]

Terim Adı	Rolü
insan	voleybolcu
top	voleybolTopu
insan	hakem

Ekle Sil

E S G

## EK-2. OWL ile NEK Ontoloji Modelinde Yer Alan Sınıf ve Bağıntı Tanımları

```
<?xml version="1.0" encoding="ISO-8859-9" ?>

<!DOCTYPE owl [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY NEKOntolojiŞablonu "file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl#">
]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="&owl;"
  xmlns:xsd="&xsd;"
  xmlns="&NEKOntolojiŞablonu;"
  xml:base="file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl#">

  <owl:Ontology rdf:about="file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl#">
    <owl:versionInfo>v1.0</owl:versionInfo>
  </owl:Ontology>

  <owl:Class rdf:ID="SpatialRelation">
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#içinde"/>
      <owl:Thing rdf:about="#üstünde"/>
      <owl:Thing rdf:about="#altında"/>
      <owl:Thing rdf:about="#yanında"/>
      <owl:Thing rdf:about="#hizasında"/>
      <owl:Thing rdf:about="#üzerinde"/>
      <owl:Thing rdf:about="#kısmeniçinde"/>
      <owl:Thing rdf:about="#dışında"/>
      <owl:Thing rdf:about="#sağYanında"/>
      <owl:Thing rdf:about="#solYanında"/>
      <owl:Thing rdf:about="#alttarafında"/>
      <owl:Thing rdf:about="#solunda"/>
      <owl:Thing rdf:about="#sağında"/>
      <owl:Thing rdf:about="#yakın"/>
      <owl:Thing rdf:about="#uzak"/>
    </owl:oneOf>
  </owl:Class>

  <owl:Class rdf:ID="Role">
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#voleybolcu"/>
      <owl:Thing rdf:about="#voleyboltopu"/>
      <owl:Thing rdf:about="#hakem"/>
      <owl:Thing rdf:about="#smaçör"/>
    </owl:oneOf>
  </owl:Class>

  <owl:Class rdf:ID="Realisation">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#hasTime"/>
        <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger"> 1 </owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

```

```

    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource = "#hasPlace"/>
        <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource = "#hasActor"/>
        <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:ID = "ActionRealisation">
    <rdfs:subClassOf rdf:resource = "#Realisation"/>
  </owl:Class>

  <owl:Class rdf:ID = "ConceptRealisation">
    <rdfs:subClassOf rdf:resource = "#Realisation"/>
  </owl:Class>

  <owl:Class rdf:ID = "OntologyTerm">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource = "#hasAlternateTermName"/>
        <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:ID = "Object">
    <rdfs:subClassOf rdf:resource = "#OntologyTerm"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource = "#hasObjectPartOf"/>
        <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:ID = "ObjectRole">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource = "#hasRolledObject"/>
        <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource = "#hasRole"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:ID = "ActionComponent">

```

```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource = "#hasObject"/>
    <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource = "#hasSpatialRelation"/>
    <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource = "#hasSubject"/>
    <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID = "Action">
  <rdfs:subClassOf rdf:resource = "#OntologyTerm"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasActionPrecedence"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasActionPartOf"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasActionComponent"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasActionDegree"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasActionRole"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasActionRealisation"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

```

<owl:Class rdf:ID = "ConceptComponent">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasOntologyTerm"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasRelevanceDegree"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

```

<owl:Class rdf:ID = "Concept">
  <rdfs:subClassOf rdf:resource = "#OntologyTerm"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasConceptComponent"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasConceptRole"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "#hasConceptRealisation"/>
      <owl:cardinality rdf:datatype = "&xsd;nonNegativeInteger"> 1 </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>

```

```

</owl:Class>

```

```

<owl:DatatypeProperty rdf:ID="hasTime">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Gerçekleştirim"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</owl:DatatypeProperty>

```

```

<owl:DatatypeProperty rdf:ID="hasPlace">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Gerçekleştirim"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</owl:DatatypeProperty>

```

```

<owl:DatatypeProperty rdf:ID="hasActor">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Gerçekleştirim"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</owl:DatatypeProperty>

```

```

<owl:DatatypeProperty rdf:ID="hasAlternateTermName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>

```



```

<rdfs:domain rdf:resource="#OntologyTerm"/>
<rdfs:range rdf:resource="#xsd:String"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="hasObjectPartOf">
<rdfs:domain rdf:resource="#Object"/>
<rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasActionPrecedence">
<rdfs:domain rdf:resource="#Action"/>
<rdfs:range rdf:resource="#Action"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasActionPartOf">
<rdfs:domain rdf:resource="#Action"/>
<rdfs:range rdf:resource="#Action"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasActionComponent">
<rdfs:domain rdf:resource="#Action"/>
<rdfs:range rdf:resource="#ActionComponent"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasActionRealisation">
<rdfs:domain rdf:resource="#Action"/>
<rdfs:range rdf:resource="#ActionRealisation"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasConceptRole">
<rdfs:domain rdf:resource="#Concept"/>
<rdfs:range rdf:resource="#ObjectRole"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasActionRole">
<rdfs:domain rdf:resource="#Action"/>
<rdfs:range rdf:resource="#ObjectRole"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasObject">
<rdfs:domain rdf:resource="#ActionComponent"/>
<rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasSubject">
<rdfs:domain rdf:resource="#ActionComponent"/>
<rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasSpatialRelation">
<rdfs:domain rdf:resource="#ActionComponent"/>
<rdfs:range rdf:resource="#SpatialRelation"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasConceptComponent">
<rdfs:domain rdf:resource="#Concept"/>
<rdfs:range rdf:resource="#ConceptComponent"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasConceptRealisation">

```

```
<rdfs:domain rdf:resource="#Concept"/>
<rdfs:range rdf:resource="#ConceptRealisation"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasOntologyTerm">
<rdfs:domain rdf:resource="#ConceptComponent"/>
<rdfs:range rdf:resource="#OntologyTerm"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="hasRelevanceDegree">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#ConceptComponent"/>
<rdfs:range rdf:resource="&xsd;decimal"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="hasActionDegree">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#Action"/>
<rdfs:range rdf:resource="&xsd;decimal"/>
</owl:DatatypeProperty>

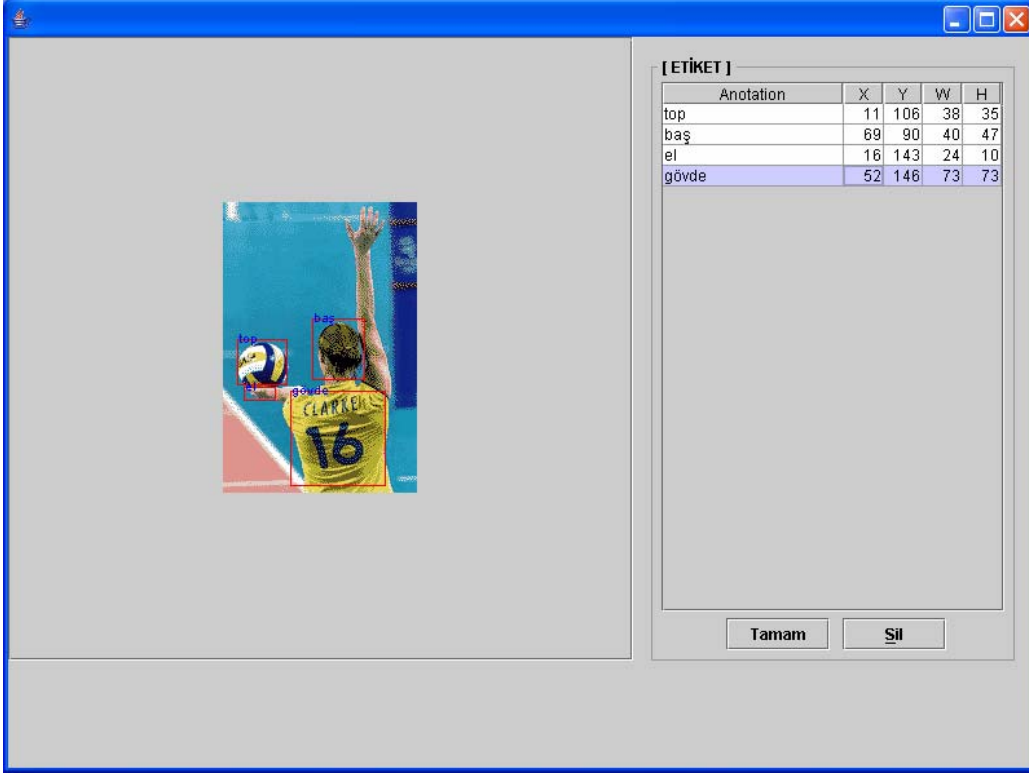
<owl:ObjectProperty rdf:ID="hasRolledObject">
<rdfs:domain rdf:resource="#ObjectRole"/>
<rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasRol">
<rdfs:domain rdf:resource="#ObjectRole"/>
<rdfs:range rdf:resource="#Role"/>
</owl:ObjectProperty>

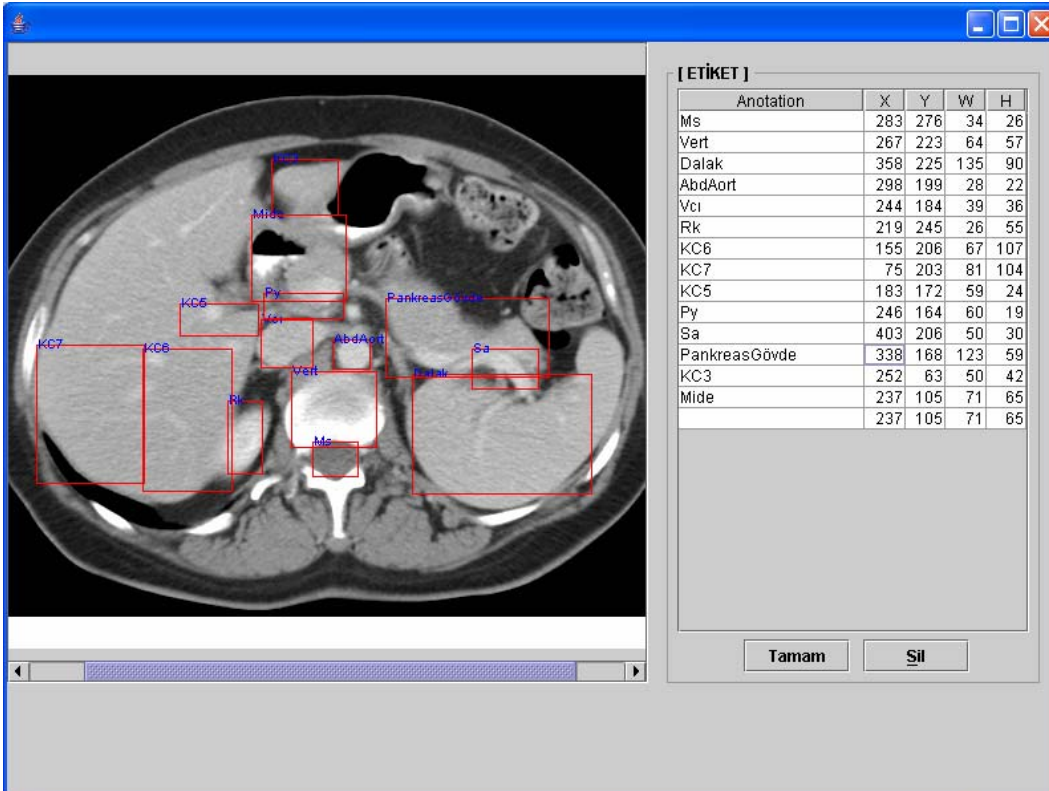
</rdf:RDF>
```

## Ek-3 Ağırlıklandırma Etiketlendirme Aracı(AEA)'nın Görüntü Etiketlendirme Arayüzü

(Deney 1)



(Deney 2)



## Ek-4. Web Sayfası Üst Veri Modelinin RDF ile Gösterimi

```
<?xml version="1.0" encoding="ISO-8859-9" ?>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY      xsd "http://www.w3.org/2001/XMLSchema#">
]>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd ="&xsd;"
  xmlns:dt      = "http://WPSemantic.rdf#">

  <rdfs:Class rdf:ID="IndividualsFile"/>

  <rdfs:Class rdf:ID ="WebPage"/>

  <rdfs:Class rdf:ID ="Term"/>

  <rdfs:Class rdf:ID ="Image"/>

  <rdfs:Class rdf:ID = "Region"/>

  <rdfs:Class rdf:ID = "RegionMatrix"/>

  <rdfs:Class rdf:ID = "HighLevelSemantics"/>

  <rdfs:Class rdf:ID = "Point"/>

  <rdfs:Class rdf:ID = "Relation"/>

  <rdfs:Class rdf:ID = "DistanceRelation">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID = "TopologicalRelation">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID = "SpatialRelation">
    <rdfs:subClassOf rdf:resource="#Relations"/>
  </rdfs:Class>

  <rdf:Property rdf:ID="hasTotalPageNumber">
    <rdfs:domain rdf:resource="#IndividualsFile"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
  </rdf:Property>

  <rdf:Property rdf:ID="hasIndividualFileName">
    <rdfs:domain rdf:resource="#IndividualsFile"/>
    <rdfs:range rdf:resource="&xsd;String"/>
  </rdf:Property>

  <rdf:Property rdf:ID="hasWebPage">
    <rdfs:domain rdf:resource="#IndividualsFile"/>
    <rdfs:range rdf:resource="#WebPage"/>
  </rdf:Property>
```

```

<rdf:Property rdf:ID="hasImage">
  <rdfs:domain rdf:resource="#WebPage"/>
  <rdfs:range rdf:resource="#Image"/>
</rdf:Property>

<rdf:Property rdf:ID="hasTerm">
  <rdfs:domain rdf:resource="#WebPage"/>
  <rdfs:range rdf:resource="#Term"/>
</rdf:Property>

<rdf:Property rdf:ID="hasWebPageID">
  <rdfs:domain rdf:resource="#WebPage"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasWebPageName">
  <rdfs:domain rdf:resource="#WebPage"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasTermName">
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasTermFrequency">
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasTermRawWeight">
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>

<rdf:Property rdf:ID="hasTermOntologicalWeight">
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageID">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageAdress">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageName">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageHeight">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

```

```

<rdf:Property rdf:ID="hasImageWidth">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageRegion">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="#Region"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageRegionMatrix">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="#RegionMatrix"/>
</rdf:Property>

<rdf:Property rdf:ID="hasImageHighLevelSemantics">
  <rdfs:domain rdf:resource="#Image"/>
  <rdfs:range rdf:resource="#HighLevelSemantics"/>
</rdf:Property>

<rdf:Property rdf:ID="hasHighLevelSemanticsName">
  <rdfs:domain rdf:resource="#HighLevelSemantics"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasHighLevelSemanticsDegree">
  <rdfs:domain rdf:resource="#HighLevelSemantics"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRegionID">
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRegionName">
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRightUpPoint">
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range rdf:resource="#Point"/>
</rdf:Property>

<rdf:Property rdf:ID="hasLeftDownPoint">
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range rdf:resource="#Point"/>
</rdf:Property>

<rdf:Property rdf:ID="hasMiddleCenterPoint">
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range rdf:resource="#Point"/>
</rdf:Property>

<rdf:Property rdf:ID="hasPointX">
  <rdfs:domain rdf:resource="#Point"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

```

```
<rdf:Property rdf:ID="hasPointY">
  <rdfs:domain rdf:resource="#Point"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRegion1ID">
  <rdfs:domain rdf:resource="#RegionMatrix"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRegion2ID">
  <rdfs:domain rdf:resource="#RegionMatrix"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRelationName">
  <rdfs:domain rdf:resource="#Relation"/>
  <rdfs:range rdf:resource="&xsd;String"/>
</rdf:Property>

<rdf:Property rdf:ID="hasRelationDegree">
  <rdfs:domain rdf:resource="#Relation"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>

<rdfs:Datatype rdf:about="&xsd;integer"/>
<rdfs:Datatype rdf:about="&xsd;String"/>
<rdfs:Datatype rdf:about="&xsd;decimal"/>

</rdf:RDF>
```

## EK-5. NEK Ontoloji Modeli ile Voleybol Alanı Ontolojisinin OWL ile Gösterimi

```
<?xml version="1.0" encoding="ISO-8859-9" ?>

<!DOCTYPE owl [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY Voleybol "file:///D:/myontoloji/sürüm1.1/Voleybol.owl#">
]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="&owl;"
  xmlns:xsd="&xsd;"
  xmlns="&Voleybol;"
  xml:base="file:///D:/myontoloji/sürüm1.1/Voleybol.owl#">

  <owl:Ontology rdf:about="file:///D:/myontoloji/sürüm1.1/Voleybol.owl#">
    <owl:versionInfo>v1.0</owl:versionInfo>
    <owl:imports rdf:resource="file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl" />
  </owl:Ontology>

  <Object rdf:ID="top">
    <rdfs:comment>volleybol oyunu aracı</rdfs:comment>
  </Object>

  <Object rdf:ID="el2"/>

  <Object rdf:ID="el1"/>

  <Object rdf:ID="direk">
    <rdfs:comment>fileyi ayakta tutan iki dik çubuk</rdfs:comment>
  </Object>

  <Object rdf:ID="baş"/>

  <Object rdf:ID="insan">
    <rdfs:comment>volleybolcu, hakem, antrenör olarak görev alabilir</rdfs:comment>
    <hasObjectPartOf rdf:resource="#ayak"/>
    <hasObjectPartOf rdf:resource="#el1"/>
    <hasObjectPartOf rdf:resource="#el2"/>
    <hasObjectPartOf rdf:resource="#el1"/>
    <hasObjectPartOf rdf:resource="#gövde"/>
    <hasObjectPartOf rdf:resource="#baş"/>
    <hasAlternateTermName>sporcu </hasAlternateTermName>
  </Object>

  <Object rdf:ID="sahadışı">
    <rdfs:comment>volleybol oyun alanının dışında kalan kesim</rdfs:comment>
  </Object>

  <Object rdf:ID="ayak"/>

  <Object rdf:ID="baş1"/>

  <Object rdf:ID="yanbantlar">
    <rdfs:comment>volleybol oyun alanının sınır çizgileri</rdfs:comment>
  </Object>
```



```

<Object rdf:ID = "gövde"/>

<Object rdf:ID = "sandalye">
  <rdfs:comment>saha hakeminin kullandığı file hizasında yüksekçe oturma aracı</rdfs:comment>
</Object>

<Object rdf:ID = "gövde1"/>

<Object rdf:ID = "bayrak"/>

<Object rdf:ID = "anten">
  <rdfs:comment>file hizasında duran ve file, direk ilişkisini hizasını kontrol etmeyi sağlayan
  çubuk</rdfs:comment>
</Object>

<Object rdf:ID = "tribün">
  <rdfs:comment>seyircilerin oturduğu mekan</rdfs:comment>
</Object>

<Object rdf:ID = "el">
  <rdfs:comment>volleybolda topa el ya da ayakla vurulabilir</rdfs:comment>
</Object>

<Object rdf:ID = "file">
  <hasObjectPartOf rdf:resource = "#direk"/>
  <hasObjectPartOf rdf:resource = "#anten"/>
</Object>

<Object rdf:ID = "düdük">
  <rdfs:comment>volleybolda oluşan aykırı durumların bildirilmesini sağlayan
  araç</rdfs:comment>
  <hasAlternateTermName>          duduk          </hasAlternateTermName>
</Object>

<Object rdf:ID = "saha">
  <rdfs:comment>volaybol oyununun oynandığı parke alan</rdfs:comment>
  <hasObjectPartOf rdf:resource = "#yanbantlar"/>
</Object>

<Action rdf:ID = "elyanda">
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#insan"/>
      <hasSpatialRelation rdf:resource = "#yakın"/>
      <hasSpatialRelation rdf:resource = "#dışında"/>
      <hasSubject rdf:resource = "#saha"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#insan"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#sahadıışı"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el"/>

```

```

        <hasSpatialRelation rdf:resource = "#hizasında"/>
        <hasSubject rdf:resource = "#gövde"/>
    </ActionComponent>
</hasActionComponent>
</Action>

<Action rdf:ID ="düzvuruş">
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#top"/>
            <hasSpatialRelation rdf:resource = "#kısmenİçinde"/>
            <hasSpatialRelation rdf:resource = "#yakın"/>
            <hasSubject rdf:resource = "#el1"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el1"/>
            <hasSpatialRelation rdf:resource = "#hizasında"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#top"/>
            <hasSpatialRelation rdf:resource = "#yakın"/>
            <hasSpatialRelation rdf:resource = "#kısmenİçinde"/>
            <hasSubject rdf:resource = "#el2"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el1"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#baş"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el2"/>
            <hasSpatialRelation rdf:resource = "#hizasında"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el2"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#baş"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el1"/>
            <hasSpatialRelation rdf:resource = "#yanında"/>
            <hasSubject rdf:resource = "#el2"/>
        </ActionComponent>
    </hasActionComponent>

```

```

    </hasActionComponent>
</Action>

<Action rdf:ID ="bayrakhavada0">
<rdfs:subClassOf rdf:resource = "#bayrakhavada"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#bayrak"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#baş"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#baş"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#gövde"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#gövde"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#sahadışı"/>
    </ActionComponent>
  </hasActionComponent>
</Action>

<Action rdf:ID ="fileyeTemas4">
<rdfs:subClassOf rdf:resource = "#fileyeTemas"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el2"/>
      <hasSpatialRelation rdf:resource = "#içinde"/>
      <hasSpatialRelation rdf:resource = "#kısmeniçinde"/>
      <hasSubject rdf:resource = "#file"/>
    </ActionComponent>
  </hasActionComponent>
</Action>
<Action rdf:ID ="fileyeTemas3">
<rdfs:subClassOf rdf:resource = "#fileyeTemas"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el1"/>
      <hasSpatialRelation rdf:resource = "#içinde"/>
      <hasSpatialRelation rdf:resource = "#kısmeniçinde"/>
      <hasSubject rdf:resource = "#file"/>
    </ActionComponent>
  </hasActionComponent>
</Action>

<Action rdf:ID ="fileyetemas1">
<rdfs:subClassOf rdf:resource = "#fileyeTemas"/>
<rdfs:comment>kurallara aykırıdır karşı tarafa sayı olarak geçer</rdfs:comment>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>

```

```

        <hasObject rdf:resource = "#el"/>
        <hasSpatialRelation rdf:resource = "#içinde"/>
        <hasSubject rdf:resource = "#file"/>
    </ActionComponent>
</hasActionComponent>
</Action>

<Action rdf:ID = "fileyetemas2">
<rdfs:subClassOf rdf:resource = "#fileyeTemas"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#gövde"/>
            <hasSpatialRelation rdf:resource = "#içinde"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
</Action>

<Action rdf:ID = "smaç0">
<rdfs:subClassOf rdf:resource = "#smaç"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#top"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSpatialRelation rdf:resource = "#kısmeniçinde"/>
            <hasSpatialRelation rdf:resource = "#yanında"/>
            <hasSubject rdf:resource = "#el"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#gövde"/>
            <hasSpatialRelation rdf:resource = "#hizasında"/>
            <hasSpatialRelation rdf:resource = "#yakın"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#gövde"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionPrecedence rdf:resource = "#düzvuruş"/>
    <hasActionRole>
        <ObjectRole>
            <hasRolledObject rdf:resource = "#insan"/>
            <hasRol rdf:resource = "#smaçör"/>
        </ObjectRole>
    </hasActionRole>

```

```

</hasActionRole>
<hasActionRealisation>
  <ActionRealisation>
    <hasTime>01.01.2004</hasTime>
    <hasPlace>ankara</hasPlace>
    <hasActor>Öznur</hasActor>
  </ActionRealisation>
</hasActionRealisation>
</Action>

<Action rdf:ID ="düzservis">
<rdfs:subClassOf rdf:resource = "#servis"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el2"/>
      <hasSpatialRelation rdf:resource = "#uzak"/>
      <hasSubject rdf:resource = "#file"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el1"/>
      <hasSpatialRelation rdf:resource = "#altında"/>
      <hasSubject rdf:resource = "#top"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#insan"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#sahadışı"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#top"/>
      <hasSpatialRelation rdf:resource = "#hizasında"/>
      <hasSubject rdf:resource = "#gövde"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el2"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#gövde"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el1"/>
      <hasSpatialRelation rdf:resource = "#uzak"/>
      <hasSubject rdf:resource = "#file"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#file"/>
      <hasSpatialRelation rdf:resource = "#uzak"/>
      <hasSubject rdf:resource = "#insan"/>
    </ActionComponent>
  </hasActionComponent>

```

```

        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#file"/>
            <hasSpatialRelation rdf:resource = "#uzak"/>
            <hasSubject rdf:resource = "#top"/>
        </ActionComponent>
    </hasActionComponent>
</Action>

<Action rdf:ID ="smaçservis">
<rdfs:subClassOf rdf:resource = "#servis"/>
<rdfs:comment>saldırı amaçlı servis</rdfs:comment>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el1"/>
            <hasSpatialRelation rdf:resource = "#yakın"/>
            <hasSpatialRelation rdf:resource = "#kısmeniçinde"/>
            <hasSpatialRelation rdf:resource = "#hizasında"/>
            <hasSubject rdf:resource = "#top"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#file"/>
            <hasSpatialRelation rdf:resource = "#uzak"/>
            <hasSubject rdf:resource = "#insan"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#insan"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#sahadışı"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el1"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#gövde"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#file"/>
            <hasSpatialRelation rdf:resource = "#uzak"/>
            <hasSubject rdf:resource = "#top"/>
        </ActionComponent>
    </hasActionComponent>
</Action>

<Action rdf:ID ="ayakçizgide">
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#ayak"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>

```

```

        </ActionComponent>
    </hasActionComponent>
</Action>

<Action rdf:ID="manşet">
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource="#el1"/>
            <hasSpatialRelation rdf:resource="#hizasında"/>
            <hasSpatialRelation rdf:resource="#kısmenİçinde"/>
            <hasSubject rdf:resource="#gövde1"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource="#el2"/>
            <hasSpatialRelation rdf:resource="#kısmenİçinde"/>
            <hasSpatialRelation rdf:resource="#hizasında"/>
            <hasSubject rdf:resource="#gövde1"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource="#el1"/>
            <hasSpatialRelation rdf:resource="#kısmenİçinde"/>
            <hasSpatialRelation rdf:resource="#yakın"/>
            <hasSubject rdf:resource="#el2"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource="#top"/>
            <hasSpatialRelation rdf:resource="#kısmenİçinde"/>
            <hasSpatialRelation rdf:resource="#üzerinde"/>
            <hasSubject rdf:resource="#el2"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource="#top"/>
            <hasSpatialRelation rdf:resource="#üzerinde"/>
            <hasSpatialRelation rdf:resource="#kısmenİçinde"/>
            <hasSubject rdf:resource="#el1"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionPrecedence rdf:resource="#smaç"/>
</Action>

<Action rdf:ID="ayaklavuruş">
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource="#top"/>
            <hasSpatialRelation rdf:resource="#yakın"/>
            <hasSpatialRelation rdf:resource="#üstünde"/>
            <hasSubject rdf:resource="#ayak"/>
        </ActionComponent>
    </hasActionComponent>
</Action>

```

```

<Action rdf:ID ="elhavada">
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#baş"/>
      <hasSpatialRelation rdf:resource = "#hızasında"/>
      <hasSpatialRelation rdf:resource = "#yakın"/>
      <hasSubject rdf:resource = "#düdük"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#insan"/>
      <hasSpatialRelation rdf:resource = "#yakın"/>
      <hasSpatialRelation rdf:resource = "#hızasında"/>
      <hasSubject rdf:resource = "#file"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#el"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#baş"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#sandalye"/>
      <hasSpatialRelation rdf:resource = "#yakın"/>
      <hasSpatialRelation rdf:resource = "#dışında"/>
      <hasSubject rdf:resource = "#saha"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#insan"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#sandalye"/>
    </ActionComponent>
  </hasActionComponent>
</Action>
<Action rdf:ID ="topiçerdedüştü">
  <rdfs:subClassOf rdf:resource = "#topyeredüştü"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#top"/>
      <hasSpatialRelation rdf:resource = "#içinde"/>
      <hasSubject rdf:resource = "#saha"/>
    </ActionComponent>
  </hasActionComponent>
</Action>
<Action rdf:ID ="topdışarıdadüştü">
  <rdfs:subClassOf rdf:resource = "#topyeredüştü"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#top"/>

```



```

        <hasSpatialRelation rdf:resource = "#içinde"/>
        <hasSubject rdf:resource = "#sahadışı"/>
    </ActionComponent>
</hasActionComponent>
</Action>

<Action rdf:ID = "fileüstümücadele">
<rdfs:subClassOf rdf:resource = "#blok"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#top"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#gövde1"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#el2"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el1"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#gövde1"/>
            <hasSpatialRelation rdf:resource = "#hizasında"/>
            <hasSpatialRelation rdf:resource = "#yakın"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#el2"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#file"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#gövde1"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#el1"/>
        </ActionComponent>
    </hasActionComponent>
</Action>

<Concept rdf:ID = "vuruş">
<rdfs:comment>voleyboldaki vuruş türleri</rdfs:comment>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#düzvuruş"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
</Concept>

```

```

        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#ayaklavuruş"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#smaçservis"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#servis"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#manşet"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
</Concept>

<Concept rdf:ID = "skor">
</Concept>

<Concept rdf:ID = "sayı">
</Concept>

<Concept rdf:ID = "kollektifBilinç">
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#takımSporu"/>
            <hasRelevanceDegree>0.8</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
</Concept>

<Concept rdf:ID = "takımsporu">
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#voleybol"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
</Concept>

<Concept rdf:ID = "set">
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#sayı"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
</Concept>

```

```

<Concept rdf:ID = "voleybol">
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#aykırıDurum"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#el"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#hücum"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#saha"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#savunma"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#top"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#vuruş"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#file"/>
      <hasRelevanceDegree>0.8</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#maç"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptRole>
    <ObjectRole>
      <hasRolledObject rdf:resource = "#insan"/>
      <hasRol rdf:resource = "#voleybolcu"/>
    </ObjectRole>
  </hasConceptRole>

```

```

        </ObjectRole>
    </hasConceptRole>
    <hasConceptRole>
        <ObjectRole>
            <hasRolledObject rdf:resource = "#top"/>
            <hasRol rdf:resource = "#voleybolTopu"/>
        </ObjectRole>
    </hasConceptRole>
    <hasConceptRole>
        <ObjectRole>
            <hasRolledObject rdf:resource = "#insan"/>
            <hasRol rdf:resource = "#hakem"/>
        </ObjectRole>
    </hasConceptRole>
    <hasConceptRealisation>
        <ConceptRealisation>
            <hasTime>11.03.2005</hasTime>
            <hasPlace>Ankara</hasPlace>
            <hasActor>PolisAkademisi-ÇankayaBelediye</hasActor>
        </ConceptRealisation>
    </hasConceptRealisation>
    <hasConceptRealisation>
        <ConceptRealisation>
            <hasTime>25.01.2005</hasTime>
            <hasPlace>Ankara</hasPlace>
            <hasActor>Halk Bankası-ManavgatSpor</hasActor>
        </ConceptRealisation>
    </hasConceptRealisation>
</Concept>

<Concept rdf:ID = "maç">
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#manşet"/>
            <hasRelevanceDegree>0.8</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#set"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#servis"/>
            <hasRelevanceDegree>0.8</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#fileyeTemas"/>
            <hasRelevanceDegree>0.8</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#skor"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>

```

```

</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#hücum"/>
    <hasRelevanceDegree>0.8</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#savunma"/>
    <hasRelevanceDegree>0.8</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#smaç"/>
    <hasRelevanceDegree>0.8</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
</Concept>

<Concept rdf:ID = "hücum">
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#blok"/>
      <hasRelevanceDegree>0.8</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#smaçservis"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#smaç"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
</Concept>

<Concept rdf:ID = "savunma">
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#manşet"/>
      <hasRelevanceDegree>0.8</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#blok"/>
      <hasRelevanceDegree>0.8</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#ayaklavuruş"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
</Concept>

```

```

    </hasConceptComponent>
</Concept>

<Concept rdf:ID = "aykırıDurum">
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#elhavada"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#elyanda"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#elhavada"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#fileyeTemas"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#topyeredüştü"/>
      <hasRelevanceDegree>0.5</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
</Concept>
</rdf:RDF>

```

## EK-6. NEK Ontoloji Modeli ile Gövde Alan Ontolojisinin OWL İle Gösterimi

```
<?xml version="1.0" encoding="ISO-8859-9" ?>

<!DOCTYPE owl [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY NEKOntolojiŞablonu "file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl#">
]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="&owl;"
  xmlns:xsd="&xsd;"
  xmlns="&NEKOntolojiŞablonu;"
  xml:base="file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl#">

  <owl:Ontology rdf:about="file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl#">
    <owl:versionInfo>v1.0</owl:versionInfo>
    <owl:imports rdf:resource="file:///D:/myontoloji/sürüm1.1/NEKOntolojiŞablonu.owl" />

  </owl:Ontology>

  <Object rdf:ID="Rpv">
    <rdfs:comment>Sağ portal ven</rdfs:comment>
  </Object>

  <Object rdf:ID="Mide">
    <hasObjectPartOf rdf:resource="#MideBağ"/>
    <hasObjectPartOf rdf:resource="#MideGöv"/>
  </Object>

  <Object rdf:ID="Mhv">
    <rdfs:comment>Orta hepatik ven</rdfs:comment>
  </Object>

  <Object rdf:ID="Vert">
    <rdfs:comment>Vertebra korpus</rdfs:comment>
  </Object>

  <Object rdf:ID="KC1">
    <rdfs:comment>Karaciğer 1. segment</rdfs:comment>
  </Object>

  <Object rdf:ID="Radr">
    <rdfs:comment>Sağ Adrenal</rdfs:comment>
  </Object>

  <Object rdf:ID="KC5">
    <rdfs:comment>Karaciğer 5. segment</rdfs:comment>
  </Object>

  <Object rdf:ID="Fl">
    <rdfs:comment>Falsiform Ligament</rdfs:comment>
  </Object>

  <Object rdf:ID="KC2">
    <rdfs:comment>Karaciğer 2. segment</rdfs:comment>
```

```
</Object>

<Object rdf:ID = "KC3">
  <rdfs:comment>Karaciğer 3. segment</rdfs:comment>
</Object>

<Object rdf:ID = "Vc1">
  <rdfs:comment>Vera cava inf</rdfs:comment>
</Object>

<Object rdf:ID = "Ms">
  <rdfs:comment>Medulla Spindis</rdfs:comment>
</Object>

<Object rdf:ID = "Lp"/>

<Object rdf:ID = "C">
  <rdfs:comment>Çölyek</rdfs:comment>
</Object>

<Object rdf:ID = "Lhv">
  <rdfs:comment>Sol hepatik ven</rdfs:comment>
</Object>

<Object rdf:ID = "Pankreas">
  <hasObjectPartOf rdf:resource = "#PankreasGöv"/>
  <hasObjectPartOf rdf:resource = "#PankreasBaşı"/>
</Object>

<Object rdf:ID = "SolBöb">
  <hasObjectPartOf rdf:resource = "#Lp"/>
  <hasObjectPartOf rdf:resource = "#Lk"/>
</Object>

<Object rdf:ID = "Pv">
  <rdfs:comment>Portal Ven</rdfs:comment>
</Object>

<Object rdf:ID = "Hv">
  <rdfs:comment>Hepatik Ven</rdfs:comment>
</Object>

<Object rdf:ID = "KC4b">
  <rdfs:comment>Karaciğer 4b. segment</rdfs:comment>
</Object>

<Object rdf:ID = "Sa">
  <rdfs:comment>Splenic arter</rdfs:comment>
</Object>

<Object rdf:ID = "SolAC">
  <rdfs:comment>Sol akciğer</rdfs:comment>
</Object>

<Object rdf:ID = "Dalak"/>

<Object rdf:ID = "Lk">
  <rdfs:comment>Sol renal korteks</rdfs:comment>
</Object>
```



<Object rdf:ID = "PankreasGöv">  
    <rdfs:comment>Pankreas Gövdesi</rdfs:comment>  
</Object>

<Object rdf:ID = "Rp">  
    <rdfs:comment>Sağ böbrek pelvis</rdfs:comment>  
</Object>

<Object rdf:ID = "Rcı">  
    <rdfs:comment>Sağ common iliak arter</rdfs:comment>  
</Object>

<Object rdf:ID = "KC4a">  
    <rdfs:comment>Karaciğer 4a. segment</rdfs:comment>  
</Object>

<Object rdf:ID = "Lcı">  
    <rdfs:comment>Sol common iliak arter</rdfs:comment>  
</Object>

<Object rdf:ID = "Rhv">  
    <rdfs:comment>Sağ hepatic ven</rdfs:comment>  
</Object>

<Object rdf:ID = "MideBağ">  
    <rdfs:comment>Mide Bağlantısı</rdfs:comment>  
</Object>

<Object rdf:ID = "Skfund">  
    <rdfs:comment>Safra kesesi fundus</rdfs:comment>  
</Object>

<Object rdf:ID = "SağAC">  
    <rdfs:comment>Sağ Akciğer</rdfs:comment>  
</Object>

<Object rdf:ID = "KC7">  
    <rdfs:comment>Karaciğer 7. segment</rdfs:comment>  
</Object>

<Object rdf:ID = "KC6">  
    <rdfs:comment>Karaciğer 6. segment</rdfs:comment>  
</Object>

<Object rdf:ID = "Rk">  
    <rdfs:comment>Sağ böbek kortes</rdfs:comment>  
</Object>

<Object rdf:ID = "SolP">  
    <rdfs:comment>Sol renal pelvis</rdfs:comment>  
</Object>

<Object rdf:ID = "Özo">  
    <rdfs:comment>Özofagus</rdfs:comment>  
</Object>

<Object rdf:ID = "Duo">  
    <rdfs:comment>Duodonum</rdfs:comment>  
</Object>

```

<Object rdf:ID = "MideGöv">
  <rdfs:comment>Mide Gövdesi</rdfs:comment>
</Object>

<Object rdf:ID = "Skkorp">
  <rdfs:comment>Safra kesesi korpus</rdfs:comment>
</Object>

<Object rdf:ID = "K">
  <rdfs:comment>Kole dok</rdfs:comment>
</Object>

<Object rdf:ID = "AbdAort">
  <rdfs:comment>Abdominal Aort</rdfs:comment>
</Object>

<Object rdf:ID = "KC8">
  <rdfs:comment>Karaciğer 8. segment</rdfs:comment>
</Object>

<Object rdf:ID = "Sm">
  <rdfs:comment>Superimer mezantirik arter</rdfs:comment>
</Object>

<Object rdf:ID = "PankreasBaşı"/>

<Object rdf:ID = "SağBöb">
  <hasObjectPartOf rdf:resource = "#Rp"/>
  <hasObjectPartOf rdf:resource = "#Rk"/>
</Object>

<Action rdf:ID = "TamKaraciğer">
  <rdfs:subClassOf rdf:resource = "#TKaraciğer"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#KC2"/>
      <hasSpatialRelation rdf:resource = "#sağYanında"/>
      <hasSubject rdf:resource = "#KC4a"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#KC7"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#SağAC"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Vcı"/>
      <hasSpatialRelation rdf:resource = "#sağYanında"/>
      <hasSubject rdf:resource = "#KC5"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#KC5"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#KC6"/>
    </ActionComponent>
  </hasActionComponent>

```

```

        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Mide"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#KC2"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Pv"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC8"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Pv"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC5"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC7"/>
            <hasSpatialRelation rdf:resource = "#solYanında"/>
            <hasSubject rdf:resource = "#KC6"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC6"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#SağAC"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Rhv"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC6"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#SağBöbrek"/>
            <hasSpatialRelation rdf:resource = "#sağYanında"/>
            <hasSubject rdf:resource = "#KC6"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC8"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC7"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>

```

```

        <hasObject rdf:resource = "#KC4a"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSubject rdf:resource = "#Pv"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Rhv"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSubject rdf:resource = "#KC7"/>
    </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID ="GenelKaraciğer">
    <rdfs:subClassOf rdf:resource = "#TKaraciğer"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Rhv"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC7"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC4a"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSpatialRelation rdf:resource = "#sağında"/>
            <hasSubject rdf:resource = "#KC8"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC2"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC1"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC4a"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#KC1"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC7"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#SağAC"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC4a"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSpatialRelation rdf:resource = "#sağYanında"/>
            <hasSubject rdf:resource = "#KC2"/>
        </ActionComponent>
    </hasActionComponent>
</ActionComponent>

```

```

<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#KC1"/>
    <hasSpatialRelation rdf:resource = "#sağYanında"/>
    <hasSubject rdf:resource = "#KC8"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Vc1"/>
    <hasSpatialRelation rdf:resource = "#altında"/>
    <hasSubject rdf:resource = "#KC2"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#KC8"/>
    <hasSpatialRelation rdf:resource = "#üzerinde"/>
    <hasSubject rdf:resource = "#Rhv"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Mide"/>
    <hasSpatialRelation rdf:resource = "#altında"/>
    <hasSpatialRelation rdf:resource = "#sağYanında"/>
    <hasSubject rdf:resource = "#KC2"/>
  </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID ="Akciğer">
  <rdfs:subClassOf rdf:resource = "#TAKciğer"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Vert"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#Ms"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Vert"/>
      <hasSpatialRelation rdf:resource = "#sağYanında"/>
      <hasSubject rdf:resource = "#SağAC"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#KC7"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#SağAC"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#AbdAort"/>
      <hasSpatialRelation rdf:resource = "#üstünde"/>
      <hasSubject rdf:resource = "#Vert"/>
    </ActionComponent>
  </hasActionComponent>

```

```

</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Vert"/>
    <hasSpatialRelation rdf:resource = "#solYanında"/>
    <hasSubject rdf:resource = "#SolAC"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Ms"/>
    <hasSpatialRelation rdf:resource = "#solYanında"/>
    <hasSubject rdf:resource = "#SolAC"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#AbdAort"/>
    <hasSpatialRelation rdf:resource = "#üstünde"/>
    <hasSpatialRelation rdf:resource = "#sağYanında"/>
    <hasSubject rdf:resource = "#SağAC"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Vcı"/>
    <hasSpatialRelation rdf:resource = "#üzerinde"/>
    <hasSubject rdf:resource = "#SağAC"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Mide"/>
    <hasSpatialRelation rdf:resource = "#üzerinde"/>
    <hasSubject rdf:resource = "#SolAC"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#AbdAort"/>
    <hasSpatialRelation rdf:resource = "#solYanında"/>
    <hasSubject rdf:resource = "#SolAC"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Ms"/>
    <hasSpatialRelation rdf:resource = "#sağYanında"/>
    <hasSubject rdf:resource = "#SağAC"/>
  </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID = "ÇiftBöbrek">
  <rdfs:subClassOf rdf:resource = "#ÇBöbrek"/>
  <hasActionPartOf rdf:resource = "#TSağBöbrek"/>
  <hasActionPartOf rdf:resource = "#TSolBöbrek"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Vert"/>

```

```

        <hasSpatialRelation rdf:resource = "#sağında"/>
        <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Lk"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSpatialRelation rdf:resource = "#solYanında"/>
        <hasSubject rdf:resource = "#Dalak"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#KC6"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSpatialRelation rdf:resource = "#solYanında"/>
        <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Lp"/>
        <hasSpatialRelation rdf:resource = "#içinde"/>
        <hasSubject rdf:resource = "#Lk"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Vert"/>
        <hasSpatialRelation rdf:resource = "#solunda"/>
        <hasSubject rdf:resource = "#Lk"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Duo"/>
        <hasSpatialRelation rdf:resource = "#sağYanında"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Rp"/>
        <hasSpatialRelation rdf:resource = "#içinde"/>
        <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID = "SolBöbrek">
    <rdfs:subClassOf rdf:resource = "#TSolBöbrek"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Lk"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSpatialRelation rdf:resource = "#solYanında"/>
            <hasSubject rdf:resource = "#Dalak"/>
        </ActionComponent>
    </hasActionComponent>

```

```

</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Lp"/>
    <hasSpatialRelation rdf:resource = "#içinde"/>
    <hasSubject rdf:resource = "#Lk"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Lk"/>
    <hasSpatialRelation rdf:resource = "#solunda"/>
    <hasSubject rdf:resource = "#Vert"/>
  </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID = "SolBörekGenel">
  <rdfs:subClassOf rdf:resource = "#TSolBörek"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Vert"/>
      <hasSpatialRelation rdf:resource = "#solunda"/>
      <hasSubject rdf:resource = "#Lk"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Lk"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSpatialRelation rdf:resource = "#solYanında"/>
      <hasSubject rdf:resource = "#Dalak"/>
    </ActionComponent>
  </hasActionComponent>
</Action>
<Action rdf:ID = "TamMide">
  <rdfs:subClassOf rdf:resource = "#TMide"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#MideBağ"/>
      <hasSpatialRelation rdf:resource = "#solYanında"/>
      <hasSubject rdf:resource = "#Mide"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#C"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#MideBağ"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#C"/>
      <hasSpatialRelation rdf:resource = "#solYanında"/>
      <hasSubject rdf:resource = "#Mide"/>
    </ActionComponent>
  </hasActionComponent>
</hasActionComponent>

```



```

        <ActionComponent>
            <hasObject rdf:resource = "#Mide"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#Dalak"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Vert"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#MideBağ"/>
        </ActionComponent>
    </hasActionComponent>
</Action>
<Action rdf:ID ="SindirimSistemi">
    <rdfs:subClassOf rdf:resource = "#TSindirim"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#PankreasBaşı"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSpatialRelation rdf:resource = "#solunda"/>
            <hasSubject rdf:resource = "#PankreasGöv"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC3"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSpatialRelation rdf:resource = "#solunda"/>
            <hasSubject rdf:resource = "#Mide"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Mide"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#PankreasBaşı"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#PankreasGöv"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#Dalak"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Pankreas"/>
            <hasSpatialRelation rdf:resource = "#üstünde"/>
            <hasSubject rdf:resource = "#Dalak"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC4b"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#KC3"/>
        </ActionComponent>
    </hasActionComponent>

```

```

</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#KC6"/>
    <hasSpatialRelation rdf:resource = "#altında"/>
    <hasSubject rdf:resource = "#KC5"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Dalak"/>
    <hasSpatialRelation rdf:resource = "#sağında"/>
    <hasSubject rdf:resource = "#Vert"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#Dalak"/>
    <hasSpatialRelation rdf:resource = "#sağında"/>
    <hasSubject rdf:resource = "#Ms"/>
  </ActionComponent>
</hasActionComponent>
<hasActionComponent>
  <ActionComponent>
    <hasObject rdf:resource = "#KC5"/>
    <hasSpatialRelation rdf:resource = "#altında"/>
    <hasSubject rdf:resource = "#KC4b"/>
  </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID = "SağBöbrek">
  <rdfs:subClassOf rdf:resource = "#TSağBöbrek"/>
  <hasActionDegree>1.0</hasActionDegree>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Duo"/>
      <hasSpatialRelation rdf:resource = "#sağYanında"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Vert"/>
      <hasSpatialRelation rdf:resource = "#sağında"/>
      <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#KC6"/>
      <hasSpatialRelation rdf:resource = "#solYanında"/>
      <hasSpatialRelation rdf:resource = "#üzerinde"/>
      <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
  </hasActionComponent>
  <hasActionComponent>
    <ActionComponent>
      <hasObject rdf:resource = "#Rp"/>
      <hasSpatialRelation rdf:resource = "#içinde"/>
    </ActionComponent>
  </hasActionComponent>

```

```

        <hasSubject rdf:resource = "#Rk"/>
    </ActionComponent>
</hasActionComponent>
</Action>
<Action rdf:ID = "SağBöbrekGenel">
    <rdfs:subClassOf rdf:resource = "#TSağBöbrek"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#KC6"/>
            <hasSpatialRelation rdf:resource = "#solYanında"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#Rk"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Duo"/>
            <hasSpatialRelation rdf:resource = "#sağYanında"/>
            <hasSpatialRelation rdf:resource = "#üzerinde"/>
            <hasSubject rdf:resource = "#Rk"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Vert"/>
            <hasSpatialRelation rdf:resource = "#sağında"/>
            <hasSubject rdf:resource = "#Rk"/>
        </ActionComponent>
    </hasActionComponent>
</Action>
<Action rdf:ID = "TamPankreas">
    <rdfs:subClassOf rdf:resource = "#TPankreas"/>
    <hasActionDegree>1.0</hasActionDegree>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#Dalak"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSubject rdf:resource = "#PankreasGöv"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#PankreasBaşı"/>
            <hasSpatialRelation rdf:resource = "#solunda"/>
            <hasSubject rdf:resource = "#PankreasGöv"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#AbdAort"/>
            <hasSpatialRelation rdf:resource = "#altında"/>
            <hasSpatialRelation rdf:resource = "#solunda"/>
            <hasSubject rdf:resource = "#PankreasGöv"/>
        </ActionComponent>
    </hasActionComponent>
    <hasActionComponent>
        <ActionComponent>
            <hasObject rdf:resource = "#K"/>
            <hasSpatialRelation rdf:resource = "#içinde"/>
        </ActionComponent>
    </hasActionComponent>

```

```

        <hasSubject rdf:resource = "#PankreasBaşı"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Mide"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSubject rdf:resource = "#PankreasBaşı"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#Skkorp"/>
        <hasSpatialRelation rdf:resource = "#solYanında"/>
        <hasSubject rdf:resource = "#PankreasBaşı"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#AbdAort"/>
        <hasSpatialRelation rdf:resource = "#sağYanında"/>
        <hasSubject rdf:resource = "#Vcı"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#KC6"/>
        <hasSpatialRelation rdf:resource = "#sağYanında"/>
        <hasSpatialRelation rdf:resource = "#altında"/>
        <hasSubject rdf:resource = "#PankreasBaşı"/>
    </ActionComponent>
</hasActionComponent>
<hasActionComponent>
    <ActionComponent>
        <hasObject rdf:resource = "#PankreasBaşı"/>
        <hasSpatialRelation rdf:resource = "#üzerinde"/>
        <hasSubject rdf:resource = "#Vcı"/>
    </ActionComponent>
</hasActionComponent>
</Action>

<Concept rdf:ID = "Toraks">
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#TamMide"/>
            <hasRelevanceDegree>0.8</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#TKaraciğer"/>
            <hasRelevanceDegree>1.0</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
        <ConceptComponent>
            <hasOntologyTerm rdf:resource = "#GenelKaraciğer"/>
            <hasRelevanceDegree>0.8</hasRelevanceDegree>
        </ConceptComponent>
    </hasConceptComponent>

```

```

<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#TSolBöbrek"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#ÇiftBöbrek"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#SolBöb"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#SindirimSistemi"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#SolBörekGenel"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#SağBöbrekGenel"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#ÇBöbrek"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#TMide"/>
    <hasRelevanceDegree>0.8</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#TSindirim"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>
<hasConceptComponent>
  <ConceptComponent>
    <hasOntologyTerm rdf:resource = "#TamKaraciğer"/>
    <hasRelevanceDegree>1.0</hasRelevanceDegree>
  </ConceptComponent>
</hasConceptComponent>

```

```

    <hasConceptComponent>
      <ConceptComponent>
        <hasOntologyTerm rdf:resource = "#TamPankreas"/>
        <hasRelevanceDegree>1.0</hasRelevanceDegree>
      </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
      <ConceptComponent>
        <hasOntologyTerm rdf:resource = "#TSağBöbrek"/>
        <hasRelevanceDegree>1.0</hasRelevanceDegree>
      </ConceptComponent>
    </hasConceptComponent>
    <hasConceptComponent>
      <ConceptComponent>
        <hasOntologyTerm rdf:resource = "#SağBöbrek"/>
        <hasRelevanceDegree>1.0</hasRelevanceDegree>
      </ConceptComponent>
    </hasConceptComponent>
  </Concept>

<Concept rdf:ID = "UstAbdominal">
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#SağAC"/>
      <hasRelevanceDegree>0.8</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#TamMide"/>
      <hasRelevanceDegree>0.4</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#SolAC"/>
      <hasRelevanceDegree>0.8</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#TMide"/>
      <hasRelevanceDegree>0.4</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#Akciğer"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#GenelKaraciğer"/>
      <hasRelevanceDegree>0.2</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#TAkciğer"/>

```

```
        <hasRelevanceDegree>1.0</hasRelevanceDegree>
      </ConceptComponent>
    </hasConceptComponent>
  </Concept>
<Concept rdf:ID = "Abdomen">
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#Toraks"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
  <hasConceptComponent>
    <ConceptComponent>
      <hasOntologyTerm rdf:resource = "#UstAbdominal"/>
      <hasRelevanceDegree>1.0</hasRelevanceDegree>
    </ConceptComponent>
  </hasConceptComponent>
</Concept>
</rdf:RDF>
```