



**ONTOLOJİ TABANLI ZEKİ ÖĞRETİM SİSTEMLERİ İLE YABANCI
DİLDE KELİME ÖĞRENME**

Emre AKDEMİR

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR BİLİMLERİ ANABİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

OCAK 2016

Emre AKDEMİR tarafından hazırlanan “Ontoloji Tabanlı Zeki Öğretim Sistemleri İle Yabancı Dilde Kelime Öğrenme” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ / ~~OY ÇOKLUĞU~~ ile Gazi Üniversitesi Bilgisayar Bilimleri Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Başkan : Prof. Dr. Erdoğan DOĞDU

Bilgisayar Mühendisliği, TOBB ETÜ

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Üye: Doç. Dr. Aslıhan TÜFEKÇİ

Bilgisayar ve Öğretim Teknolojisi Eğitimi, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Tez Savunma Tarihi: 06/01/2016

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....
Doç. Dr. Nurettin TOPALOĞLU

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.


Emre AKDEMİR

06/01/2016

ONTOLOJİ TABANLI ZEKİ ÖĞRETİM SİSTEMLERİ İLE YABANCI DİLDE
KELİME ÖĞRENME
(Yüksek Lisans Tezi)

Emre AKDEMİR

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ
Ocak 2016

ÖZET

Zeki öğretim sistemleri eğitim sürecini geliştirmede önemli katkılar sağlamaktadır. Ontoloji tabanlı zeki öğretim sistemleri karmaşık birçok problemi ortadan kaldırmaktadır. Zeki öğretim sistemlerinin sağladığı bireysel gelişim, mekandan ve zamandan bağımsız öğrenme gibi faydalara, ontolojilerin sağladığı konsept ile desteklenerek önemli adımlar atılmıştır. Günümüzde anlamsal web' in sunduğu imkanlar ile eğitim alanında sağlam yazılımlar geliştirilmektedir ve bu yazılımların kullanımları hızla artmaktadır. Teknik bir alan ya da yabancı dil sınavları için yabancı dilde kelime öğrenme ihtiyacı vardır. Bu ihtiyacı gidermek için etkin ve verimli bir şekilde iyi bir öğrenme ortamı sunulması gerekmektedir. İstenilen bu ortamı oluşturmak için kişisel ve akıllı bir sistem tasarlanmalıdır. Bu amaçla bu çalışmada ontoloji tabanlı zeki bir öğretim sistemi geliştirilmiş ve tasarım detaylarının yanı sıra bu sistemin kullanımı ile elde edilen sonuçlar değerlendirilmiştir. Yapılan kullanıcı değerlendirmesi ile olumlu sonuçlar alınmıştır. Ayrıca uzman değerlendirilmesi yapılmış olup elde edilen sonuçlar geliştirilen sistemin öngörülen başarısının istenilen seviyede olduğunu gösterir niteliktedir.

Bilim kodu : 902.1.182
Anahtar Kelimeler : Zeki öğretimi sistemleri, ontoloji, anlamsal web.
Sayfa Adedi : 84
Danışman : Doç. Dr. Hacer KARACAN

VOCABULARY LEARNING IN FOREIGN LANGUAGES BY ONTOLOGY-
BASED INTELLIGENT TUTORING SYSTEMS

(M. Sc. Thesis)

Emre AKDEMİR

GAZİ UNIVERSITY
INFORMATICS INSTITUTE

January 2016

ABSTRACT

Intelligent tutoring systems have had substantial contributions in advancement of educational processes. Ontology-based intelligent tutoring systems solve various complex problems. There have been important developments in the field by supporting benefits of intelligent tutoring systems such as personal development and time/space independent learning, with concepts provided by ontologies. Today robust educational software is developed with the use of facilities provided by semantic web and it finds wide-spread use. In a technical field or in a foreign language exam acquisition of a foreign language vocabulary is needed. In order to address this need, an effective and efficient learning environment is required. In order to create this environment, an personalized and intelligent system must be designed. For this purpose, this study demonstrates development of an ontology-based intelligent tutoring system in which detailed design is provided along with evaluation of system usage results. User evaluation yielded positive results. Furthermore experts evaluations are made. Results have shown that developed system's success is at the anticipated success level.

Science Code : 902.1.182
Key Words : Intelligent tutoring systems, ontology, semantic web.
Page Number : 84
Supervisor : Assoc. Prof. Dr. Hacer KARACAN

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren, kıymetli tecrübelerinden faydalandığım danışmanım Doç. Dr. Hacer KARACAN'na ayrıca anket çalışmalarımda yardımcı olan Gazi Teknik ve Mesleki Anadolu Lisesi öğrencilerine, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli aileme ve arkadaşlarım Abdullah Tarık YILMAZ, Tuğrul YARDIMCI ve Hasan EKİNCİ'ye teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. ZEKİ ÖĞRETİM SİSTEMLERİ VE KULLANIMI	5
2.1. Zeki Öğretim Sistemleri Modülleri.....	6
2.1.1. Uzman bilgi modülü	7
2.1.2. Öğrenci modeli modülü	8
2.1.3. Öğretim modülü.....	8
2.1.4. Kullanıcı arabirim modülü.....	9
2.2. Zeki Öğretim Sistemlerinde Kullanılan Yapay Zeka Teknikleri.....	10
2.2.1. Uzman sistemler	10
2.2.2. Karar ağaçları.....	13
2.2.3. Yapay sinir ağları.....	14
2.2.4. Veri madenciliği	18
2.3. ANLAMSAL WEB	19
2.3.1. Bağlı veri	20
2.3.2. Anlamsal web mimarisi	22
2.3.3. Ontoloji.....	24

	Sayfa
2.3.4. XML	26
2.3.5. RDF.....	26
2.3.6. OWL	27
3. ÖRNEK UYGULAMA	33
3.1. Sistem Mimarisi.....	33
3.2. Ontolojinin Hazırlanması.....	34
3.2.1. Alan ontolojisi	41
3.2.2. Öğrenci modeli ontolojisi	45
3.2.3. Öğretim strateji ontolojisi	46
3.2.4. Arayüz ontolojisi	47
3.3. Ontolojilerin Birleştirilmesi.....	49
3.4. Jena Framework.....	53
3.5. Java Persistence Api	56
3.6. Uygulamanın Arayüzü ve Uygulamanın Kullanımı	56
4. GELİŞTİRİLEN SİSTEMİN DEĞERLENDİRİLMESİ	65
4.1. Yöntem.....	65
4.2. Sonuçlar	66
5. SONUÇ VE TARTIŞMA.....	71
KAYNAKLAR.....	73
EKLER	77
EK-1(a). Uzman değerlendirme cevapları	78
EK-1(b). Uzman değerlendirme cevapları.....	79
EK-1(c). Uzman değerlendirme cevapları	80
EK-1(d). Uzman değerlendirme cevapları.....	81

EK-1(e). Uzman deęerlendirme cevapları	82
EK-1(f). Uzman deęerlendirme cevapları	83
ÖZGEÇMİŞ	84

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. İki ile çarpım örnekler ile eğitim seti ve test seti örnekleri	16
Çizelge 3.1. Durum tabloları	53
Çizelge 3.2. Sistem tabloları	53
Çizelge 4.1. Uzman değerlendirme soruları	65
Çizelge 4.2. Kullanıcı değerlendirme anket soruları ve boyutları	66

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. ZÖS mimarisi	7
Şekil 2.2. Uzman sistemin yapısı	11
Şekil 2.3. Bilginin “Eğer-O halde” kuralı sunumuna ait örnek	12
Şekil 2.4. Bilginin çatı tabanlı sunumuna ait örnek	12
Şekil 2.5. Bilginin semantik ağ sunumuna ait örnek	13
Şekil 2.6. Karar ağacı öğrenmesine ait örnek	14
Şekil 2.7. Sinir ağının yapısı	15
Şekil 2.8. İki ile çarpım tablosunu öğrenen ağ.....	16
Şekil 2.9. ZÖS ile sinir ağlarının kullanımı	17
Şekil 2.10. ZÖS’ne uyarlanmış sinir ağları mimarisi	18
Şekil 2.11. Veri tabanında bilgi keşfi süreci	19
Şekil 2.12. Bağlı veri anlamlandırma örneği birinci adım.....	21
Şekil 2.13. Bağlı veri anlamlandırma örneği ikinci adım	21
Şekil 2.14. Bağlı veri anlamlandırma örneği üçüncü adım.....	21
Şekil 2.15. Bağlı veri anlamlandırma örneği dördüncü adım	22
Şekil 2.16. Veri kümesi.....	22
Şekil 2.17. Anlamsal web mimarisi	23
Şekil 2.18. İki web tabanlı ZÖS’nin ontolojleri paylaşımı	25
Şekil 2.19. XML’e ait bir örnek.....	26
Şekil 2.20. RDF ile tanımlanmış bir örneğin grafiği	27
Şekil 2.21. RDF ile tanımlanmış bir örnek	27
Şekil 2.22. OWL ile tanımlanmış bir örnek	28
Şekil 2.23. Individuals, class ve properties arasındaki ilişki	29
Şekil 2.24. Annotation property, object property ve data property	30
Şekil 3.1. Uygulamanın yapısı	33
Şekil 3.2. Protege editörü.....	35
Şekil 3.3. Eşit sınıf.....	36
Şekil 3.4. Kaplama aksiyomsuz ve kaplama aksiyomlu görünümüne bir örnek	36
Şekil 3.5. Ayrık sınıf.....	37
Şekil 3.6. Nesne özellikleri	38
Şekil 3.7. Inferences modu.....	39

Şekil	Sayfa
Şekil 3.8. Veri özellikleri	40
Şekil 3.9. Bireylerin nesne ve veri özellikleri	41
Şekil 3.10. Kelimeler arası ilişkiler.....	42
Şekil 3.11. Alan ontolojisi	44
Şekil 3.12. Öğrenci modeli ontolojisi	46
Şekil 3.13. Öğretim stratejisi ontolojisi	47
Şekil 3.14. Arayüz ontolojisi	48
Şekil 3.15. Eşitlik sınıfı tanımlayan java kodları	49
Şekil 3.16. Birleştirilmiş ontolojilerin OWLViz ile gösterimi	50
Şekil 3.17. Owl kodları	51
Şekil 3.18. Jena tabloları.....	54
Şekil 3.19. jena_g1t1_stmt tablosu veri görünümü	54
Şekil 3.20. OntModel sınıfından bir nesne tanımlayan java kodları.....	54
Şekil 3.21. OntModelSpec sınıfından bir nesne tanımlayan java kodları	55
Şekil 3.22. Jena tabloları oluşturan java kodları	55
Şekil 3.23. Java kodları veri tiplerini güncelleyen metot.....	56
Şekil 3.24. Projenin sınıf hiyerarşisi	57
Şekil 3.25. Uygulamanın akış diyagramı	58
Şekil 3.26. ILS testi.....	59
Şekil 3.27. ILS testi sonucu	59
Şekil 3.28. Uygulamadaki öğrenci kullanıcısı arayüzü	60
Şekil 3.29. Veri alımı sözel öğrenme stili olan arayüz	61
Şekil 3.30. Veri alımı görsel öğrenme stili olan arayüz.....	61
Şekil 3.31. Uygulamada ki bazı aktiviteler	62
Şekil 3.32. Uygulamada ki bazı uyarı mesajları	62
Şekil 3.33. Ontolojide kelime nesne özelliklerine göre soru ve cevaplarının hazırlanması	63
Şekil 3.34. Uygulamadaki öğretmen kullanıcısı arayüzü	64
Şekil 4.1. Anket sonuçları	68
Şekil 4.2. Öğrenci veri özelliklerindeki değerler	69
Şekil 4.3. Öğrenci puanlarını gösteren grafikler	69

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar

Açıklamalar

BDÖ

Bilgisayar Destekli Öğretim

ILS

Index Of Learning Styles

JPA

Java Persistence Api

KA

Karar Ağaçları

OWL

Web Ontology Language

RDF

Resource Description Framework

URI

Uniform Resource Identifier

US

Uzman Sistemler

W3C

WWW Consortium

XML

Extensible Markup Language

YSA

Yapay Sinir Ağları

ZÖS

Zeki Öğretim Sistemleri

1. GİRİŞ

Hızla gelişen bilgisayar teknolojisi ile çoğu alışkanlıklarımızı da değiştirmektedir. Günümüzde akıllı telefon, tablet bilgisayar, akıllı tahta gibi çeşitli teknolojiler de bütünleşik olarak kullanılmaya başlanmıştır. Bu teknolojilerin giderek kullanımının yaygınlaşmasıyla beraber gündelik işlerimizde önemli bir yer tutmaktadır. Bu teknolojilerin kolay kullanımı, erişimi, taşınabilirliği gibi özellikler bilgiye her zaman her yerde hızlı bir şekilde ulaşma imkânı sağlamaktadır. Her alanda giderek kullanımı artan bu teknolojiler eğitim alanında da önemli bir yere sahiptir.

Eğitim alanında özellikle son yıllarda bilgisayar teknolojileri oldukça önem kazanmıştır. Neredeyse her okulda bilişim teknolojileri laboratuvar ve sınıflarının bulunması, derslerin bilgisayar ve projeksiyon cihazları ile işlenmesi bilgisayar teknolojilerinin önemini göstermektedir. Akıllı tahta ve tablet bilgisayar gibi yeni teknolojilerin kullanılmaya başlanması ile daha etkin ve hızlı bir öğrenmenin gerçekleşmesi kaçınılmazdır. Böylelikle eğitimde kullanılan geleneksel yöntemlerin yerini daha zengin içeriğe sahip etkileşimli sistemlere dayalı yöntemler almaktadır.

Eğitimde öğretmenin rolü önemli bir yere sahiptir. Bilgisayar teknolojilerinin eğitimde kullanımı öğretmenin yerini almak değil daha çok destekleyici amaçlıdır. Öğretmenin her öğrenci gelişimi ile birebir ilgilenmesi oldukça zordur. Ayrıca öğretmenin olmadığı zamanlarda öğrencilerin çalışmalarına devam etmeleri gerekebilir. Bu problemleri çözmek için bilgisayar teknolojilerinin kullanımı oldukça yararlıdır. Bu amaçla bilgisayar teknolojisinin gelişimine paralel olarak kullanılan yöntemlerde sürekli gelişmektedir. Bilgisayar Destekli Öğretim (BDÖ) ile eğitim alanında önemli çalışmalar yapılmıştır. BDÖ ile öğrencilere mekân ve zamandan bağımsız olarak öğrenmelerini sağlar. BDÖ öğrencilerin ortak olarak etkili öğrenmeyi sağlamak için öğrenme etkinliklerinin en iyi teknikler ile planlı olarak yürütülmesini amaçlar [1]. Bu amaçla bilgisayar destekli eğitim yazılımları, web tabanlı öğrenme, eğitim amaçlı bilgisayar oyunları ile çeşitli uygulamalar gerçekleştirilmiştir.

Nikoopour ve Kazemi (2014) tarafından dijital ve dijital olmayan ortamlarda kelime öğrenmesine yönelik bir çalışma sunulmuştur. Bu çalışma öğrencilere haftada 70 kelime mobil, çevrim içi ve kağıt kullanılarak flaş kartlar ile öğretim sağlanmıştır. 10 hafta

sonundaki performansların sonucunda mobil ile kullanımda daha yüksek sonuçlar elde edildiği görülmüştür [2].

Ferreira ve Atkinson (2009) tarafından yabancı dil öğrenmek için geri dönüş sağlayan bir Zeki Öğretim Sistemi (ZÖS) tasarlanmıştır. Bu sistem bireysel öğrenme, öğretim pedagojisi, geri dönüş stratejileri gibi yeteneklere sahiptir. Öğrenciler Yanıt-Verme Stratejisi ve Yanıt-Teşvik Etme Stratejisi olarak deney için iki gruba ayrılmıştır. İlk gruptaki öğrencilerin yaptıkları hataları öğretmen doğrudan hedefe yönlendirmiştir. İkinci gruptaki öğrencilerin yaptıkları hataları ise öğretmen doğrudan değil, kendi kendilerine düzeltmelerine teşvik etmiştir. Deney sonuçlarında ikinci gruptaki öğrencilerin hatalarını düzeltme oranları daha yüksek olduğu görülmüştür [3].

BDÖ ile yapılan çalışmalarda öğrencilere ortak olarak düzenlendiği için kişisel değerlendirme söz konusu değildir. Öğrencilerin kişisel başarılarını artırmak için hangi yöntemin o öğrenci üzerinde başarılı olduğu, bir konuda üzerinde ne kadar durulacağı, yapması gereken soru ve uygulamalar sıklığı hakkında yönlendirilmeleri gereklidir. Yani bir öğretmenin etkisi oluşturulmalıdır. Bu yüzden öğrenciye göre öğretim sağlayan, bir öğreticiyi taklit eden ileri öğrenme tekniklerine ihtiyaç duyulmuştur. Yapay zeka alanındaki gelişmeler ile ZÖS bu konudaki ihtiyaçları gidermeyi amaçlamıştır. Bu sistemler insan belleğinde bilginin hangi yöntemlerle nasıl aktarılacağını öğrenen bir bilişsel öğrenme kuramına dayanmaktadır [4]. Carnegie-Mellon Üniversitesi'nde geleneksel BDÖ sistemleri ile ZÖS'ni karşılaştıran bir araştırmaya göre, ZÖS'nin öğrenme kalitesini %43 arttırdığı, öğrenme sürecinin %30 düşürdüğü tespit edilmiştir [5]. Yapay zeka teknikleri ile geliştirilen ZÖS'nin öğrenci performanslarındaki etkisi açıkça görülmektedir.

ZÖS'nin gelişimi 1970'lerde başlamıştır. 1980'lerde genel model yaklaşımı sunulmuştur. 1990'larda önceki yıllarda yapılan çalışmaların geliştirilmesi yönünde devam etmektedir [6]. ZÖS 2000'li yıllarda artış göstermiştir [7]. ZÖS ile birlikte ortamdan ve zamandan bağımsız, daha etkin bir öğrenme sağlanmıştır. Bugün, Zeki Öğretim Sistemleri ile kurumsal eğitim, ortaöğretim, üniversite ve askeri eğitimleri destekleyen çalışmalar yapılmıştır ve bu teknolojilerin eğitimde kullanımına öncelik verilmiştir [7,8].

ZÖS ile yapılan çalışmalarda karar ağaçları, uzman sistemler, veri madenciliği, anlamsal web gibi yapay zeka teknikleri kullanıldığı görülmektedir. Bu teknikler ile çoğu problemin

çözümünde büyük kolaylık sağlamıştır. Anlamsal web ile kullanılan ZÖS sağlam bir ontoloji ile uygulanabilirliği yüksek ve genişletilebilir bir yapıya sahip olabilir. Ayrıca ZÖS’nde bulunan modüllerin ve aralarında ilişkilerin ontoloji ile kurulması daha sistemlidir.

Anlamsal web teknolojisi ZÖS ile kullanımı hızla yaygınlaşmaktadır. E-öğrenme sistemlerinde ontoloji önemli bir role sahiptir. E-öğrenme sistemlerinin omurgası olarak bilinen ontoloji uygulanmasına ilişkin ilk çalışma Mizoguchi ve Bourdeau (2000) tarafından sunulmuştur [9]. O zamandan beri birçok araştırmacı uyarlanabilir hipermedya, kişiselleştirme, öğrenci modelleme gibi e-öğrenmenin farklı alanlarının ontolojilerini kullanmayı önermiştir [10,11]. Mizoguchi ve Bourdeau (2000) tarafından sunulan çalışmada ontolojiler üç seviyede sunulmuştur. Seviye 1’de terimlerden toplanan bir yapı mevcuttur. Ontolojinin esası olan kavramların tanımı ve hiyerarşik yapısı belirlenmiştir. Seviye 2’de seviye 1’de bulunan kavramların biçimsel tanımı, ilişkileri ve kısıtlamaları belirlenmiştir. Seviye 3’te ise ontoloji anlamsal olarak çalışabilir. Kavramlar ve ilişkilerini sağlayan bazı kodlar tarafında ontoloji tabanlı modeller yapılandırılır [9]. Anlamsal web ile e-öğrenme gelecekteki ilerlemeler için sağlam bir temel hazırlamaktadır [12].

Daha sonraki yıllardaki çalışmalarda daha gelişmiş uygulamalardan söz edilebilir. Oguejiofor ve ark. (2004) tarafından sunulan çalışmada personel hava araçları için uçuş eğitimi çeşitli fotoğraf ve video ile gösterimi yapılmıştır [13]. Jacinto ve ark. (2008) tarafından temel istatistik öğretme amaçlı WebStatisTutor adında bir proje gerçekleştirmişlerdir. Bu projede ontoloji tabanlı bir ZÖS mimarisi sunulmuştur. Mimaride altı kısımdan ve her bir kısımda özel stratejiler olan ontoloji ve alt ontolojiler vardır. Bu kısımlar hazırlanma, sunum, asimilasyon, problem çözme, doğrulama ve değerlendirmedir [14]. Escudero ve ark. (2010) akıllı ders sistemi için kurslar oluşturan bir mimari sunmuştur. Kullanıcıya göre ontoloji kurs seçimi yapmaktadır [15]. Java programlama dili öğretimi sağlayan ontoloji tabanlı ZÖS mimarisi Vesin ve ark. (2012) tarafından sunulmuştur. Bu mimaride alan ontolojisi ile uzman bilgi modülünü, öğrenci modeli ontolojisi ile öğrenci modeli modülü, öğretim stratejisi ontolojisi ile öğretim modülü, arayüz ontolojisi ile kullanıcı arabirim modülü oluşturulmuştur. Bunlara ek olarak görev ontolojisi sistemdeki aktivitelerin sunumunu ve iletişim ontolojisi de bu birimlerin aralarında tanımlanmış semantik mesajlar iletir [10,11]. Yapılan çalışmalara bakıldığında son yıllarda ontoloji tabanlı ZÖS oldukça yaygınlaşmaya başlanmıştır ve oldukça başarılı sonuçlar vermiştir.

Yabancı dilde kelime öğrenme çoğu insanın ihtiyacıdır. Kişiler kelimelerin genel anlamlarını ya da bir alana ait anlamlarını öğrenmek istemektedirler. Özellikle yabancı dil sınavlarına hazırlanırken belirli bir süre içerisinde yeterli kelime öğrenme ihtiyacı oldukça fazladır. Bu süreç kişiler için zor ve zahmetlidir. Bu amaçla kelime öğreniminde bilgisayar destekli yazılımlar mevcuttur. Bu yazılımların her kullanıcı için ortak olmasından ziyade kişisel ve akıllı bir sistem olması kelime öğreniminde geçen bu süreci kolaylaştırmakta ve etkinliği artırmak için önemlidir. Bu çalışmanın amacı kişiye göre, etkin ve verimli bir öğrenme sağlayan bir Zeki öğretim sistemi ile ontoloji tabanlı bir eğitim yazılımını geliştirmektir. Ayrıca geliştirilen bu yazılımın kullanılabilirliği ve öğrenme başarısı üzerindeki etkilerini uzman görüşleri ve kullanıcı değerlendirmeleri ile araştırmaktır.

Bu çalışmada yabancı dilde kelime öğrenimi amaçlanmıştır. İngilizce ve Türkçe dilleri arasında kelime öğrenimi seçilmiştir. Öğrenim, öğrencinin öğrenme amacı, öğrenme yeteneği ve performansına göre sunulmaktadır. ILS testi ile öğrencinin öğrenme stilleri belirlenmektedir. Kişisel öğrenme ile öğrenci takip edilerek en iyi öğrenme ortamı hazırlanmaktadır. Öğrenci düzeyi belirleme, ders sunumları, testler gibi materyallerin hazırlanması ve planlanması gerçek bir öğretmene benzetilmektedir. Öğrenci hakkında bilgiye sahip olup gerekli yerde müdahale edilerek öğrenci yönlendirilmektedir. Ayrıca uygulamadaki kelimeler öğretmen kullanıcıları tarafından güncelleme ve yeni kelime ekleme imkânı sayesinde sürekli genişlemektedir. Böylece zamanla büyüyen bir bilgi kütüphanesine dönüşmektedir.

Bu tezin ikinci bölümünde zeki öğretim sistemleri ve anlamsal web konuları bahsedilecektir. Tezin üçüncü bölümünde uygulamanın mimarisi ve kullanımı ile ilgili bilgi verilecektir. Tezin dördüncü bölümünde geliştirilen sistemin değerlendirilmesinden bahsedilecektir. Tezin beşinci bölümde ise sonuçlara yer verilecektir.

2. ZEKİ ÖĞRETİM SİSTEMLERİ VE KULLANIMI

Zeki Öğretim Sistemleri (ZÖS) yüksek kalitede ve etkin eğitimi amaçlayan, bu amaçla öğrenciye uzman eğitimci ile birebir etkileşimdeymiş gibi bireysel ortam sağlamaya çalışan, gerekli kaynakları zamanında sunan, bireylere göre uyarlanmış eğitim sistemleridir. Alan Uzmanı, Öğretim Tasarımcısı, Eğitim Psikoloğu, Yazılım Mühendisi gibi birçok meslek gruplarının ortak çalışmalarının ürünüdür [5,16].

ZÖS'nin sağladığı en önemli özelliklerden birisi kişisel öğrenmedir. Kişisel öğrenme ile öğrencinin seviyesi, öğrenme hızı ve çalışma süresi kendisine göre belirlenmektedir. Böylece öğrenci bildiği konularda daha az vakit harcar, zorlandığı konularda ise üzerinde durularak öğrenme gerçekleştirilmeden bir sonraki konuya geçilmez. Böylece öğretimin verimliliği artmaktadır.

Kişisel öğrenme ZÖS'ne uygulandığında esneklik getirir. Öğrenci ve öğretmen için destekleyici sistem, öğrenci merkezli sistem, öğretici sistem gibi bazı karakteristik öğrenme meydana getirir. Bu sistemde, öğretilecek bir konu için içerik çatısı oluşturulmaktadır. Ders için bilgi analizi, ders planı ve içerik belirlenmektedir. Kişisel ders öğretim için ise, farklı kullanıcılara göre farklı bilgi getirme, öneri ve gerekli işlemlerde yönlendirme ve kişisel bilgileri toplayıp değerlendiren bir sistem sağlar [17].

Zeki öğretim sistemleri son derece kapsamlı, karmaşık ve birden fazla bilim dalı ile beraber çalışsan sistemlerdir. Kurulan bir sistem genelde bir alana hitap eder. Daha sonra başka teknoloji ile sistem genişletilebilir. Hatta istenilen kısımlar çıkartılabilir, değiştirilebilir ve ilgili içeriğe göre sistem tekrardan uyarlanır. Zeki öğretim sistemleri ile ilerleyen ve birden fazla zeki öğretim sisteminin birlikte çalışabilirliği sağlanır. Farklı ZÖS'i arasındaki bilgi modelinin tekrar kullanımına izin verir. Böylece verimli bir bilgi alanı sürekli değerlendirilebilir, tekrar sistem oluşturulması önlenir, daha az çaba ve zaman harcanır. ZÖS tasarımı ve geliştirmesinde çeşitli örneklerin bu şekilde kullanıldığı görülmektedir [6, 15,18].

ZÖS'nin geliştirilmesi daha öncede belirtildiği gibi zaman alan karmaşık sistemlerdir. Bilgisayar teknolojilerinin ve öğretim teknolojilerinin bir arada bulunduğu yapay zeka (YZ) teknikleri içermektedir. Kullanılan bu teknolojiler ile mimari yapısı uygulama alanlarına

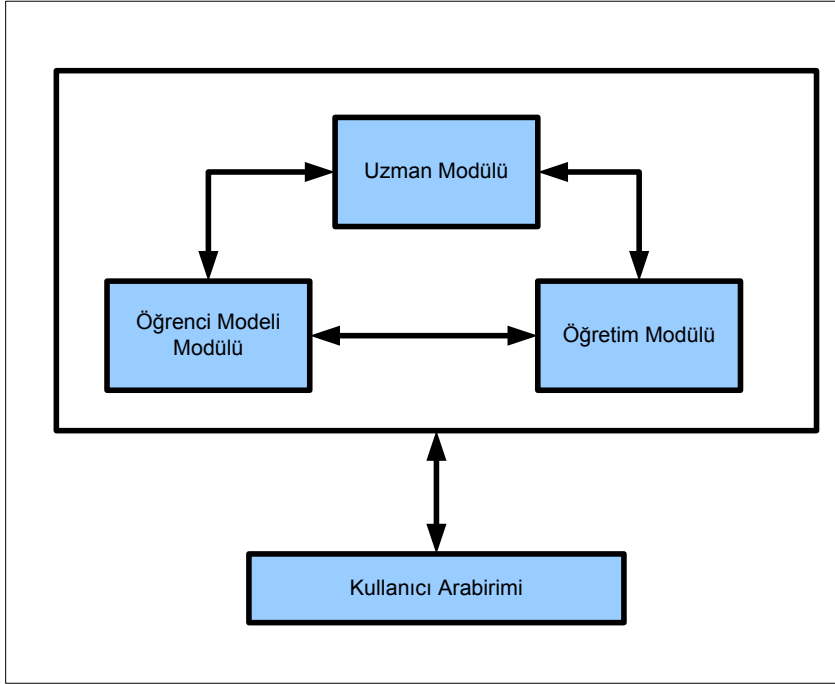
göre farklılık göstermektedir. ZÖS tasarlanırken temel olarak neyi, kime, nasıl öğreteceği planlanmaktadır. Ne öğreteceği tasarlanırken uygulama alanına göre özellik göstermektedir. Örnek, tıp, programlama, yabancı dil gibi alan bilgisine göre oluşturulmaktadır. Kısaca öğretilecek bilgiyi içermektedir. Kime öğreteceği tasarlanırken, öğrenme işleminin gerçekleştirilebilmesi için öğrencilerin özelliklerini sunmaktadır. Öğrenciye kaynak sağlar ve sisteme uyumluluğuna göre bilgisi belirlenir. Nasıl öğreteceği tasarlanırken ise öğrencilerin öğrenmeyi hangi yoldan yapacağını belirtmektedir. Öğrenci için uygun yöntem önerilmektedir [19].

2.1. Zeki Öğretim Sistemleri Modülleri

ZÖS Genel olarak literatürde dört temel modülden oluşmaktadır [5]. Bu modüller;

- Uzman Bilgi Modülü (Expert Module)
- Öğrenci Modeli Modülü (Student Module)
- Öğretim Modülü (Tutor Module)
- Kullanıcı Arabirim Modülü (User Interface Module)

Şekil 2.1’de genel olarak ZÖS mimarisi gösterilmektedir.



Şekil 2.1. ZÖS mimarisi [6]

2.1.1. Uzman bilgi modülü

Alan bilgisine dayalı bir modüldür. Öğrenme yöntemleri ile en iyi öğrenme sağlanması için hazırlanan kural seti ile çıkarım mekanizması tavsiyelerde bulunmaktadır. Öğrenciye göre kişiselleştirilmiş planlamaya dayanır. Bunun için öğrenci modeli modülünde ki bilgilere ihtiyaç duyar. Böylece tavsiyeci bir rol oynar. Kural tabanı öğretilecek konuların seçilmesinde ve sıralanmasına karar vermek için planlayıcıya izin veren ve öğrenciye uygun testlerin hazırlanması ve gösterilmesi konusunda tavsiyeciye izin veren çıkarım mekanizmasından sağlanan kurallardan oluşur [6].

Bu modülde bilgi saklandığı gibi kontrolde edilir. Bu bilgiler, kural tabanlı, semantik ağlar ve çerçeveler gibi yöntemler ile sunulabilir. Kullanılan teknikler problemin nasıl çözüleceğini belirler. Bu yöntemler ile iki temel işlev gerçekleştirilir. Birincisi, bir kaynak gibi davranış gösterme, ikincisi ise bilişsel düzeyi saptamak için öğrenci performanslarını değerlendirmedir. Performans değerlendirme sonucu öğretim içeriği belirlenir. Bu yüzden hem öğretim modülü hem de öğrenci modeli modülü arasında sürekli bilgi alışverişinde bulunmaktadır. Bu yöntemleri bilgisayar ortamında hazırlayıp aralarında bağlantı kurup organize edilmesi oldukça karmaşık ve zaman alan bir süreçtir [4, 5, 7].

Uzman bilgi modülü bir uzmanı taklit eder. Bilgi tabanı ne kadar çok gelişmiş ise çözüm için o kadar yeterli olacaktır. Aksi takdirde yeterli bilgi içermeyen bir bilgi tabanı öğrenci için eksik bir öğretim sağlar. Öğrencinin zorlandığı alanları belirlemede ve bunun için çözümler üretmesinde bir uzman gibi davranış sergileyebilmelidir.

2.1.2. Öğrenci modeli modülü

Zeki Öğretim Sistemlerinin en önemli özelliklerinden biri öğrenciye göre uyarlanabilir olmasıdır. Öğrenci ile bire bir öğrenme ortamı sağlanmaktadır. Böylece öğrenci eksiklerini görebilir. Bu süreçte öğrenmede ki her bir adım kaydedilir. Öğrencinin tutumu, sorulara verdiği cevaplar, işlem adımları izlenerek performansı değerlendirilir. Uzman modülünde verilen kararlar doğrultusunda öğrenme yöntemi öğrencinin özelliklerine bağlı olarak belirlenerek öğretim modülüne gönderilir. Öğrenme süreci tekrarlanarak geri dönüş sağlanır ve öğrenciye bireysel tavsiyeler oluşturularak rehberlik edilir [6, 7].

Öğrencinin davranışlarını ölçmek ve bilişsel bilgi düzeyini saptamak için öğrenci profili oluşturulması gerekmektedir. Öğrenci sistemi kullanırken profiline giriş yapabilmeli ve bilgileri tutulmalıdır. Bir sonraki girişte bu bilgiler kullanılarak devam edilmelidir. Böylece öğrenci takibi bireysel olarak gerçekleştirilebilir. Öğrenci bilgileri genellikle uzun dönemli ve kısa dönemli olmak üzere iki şekilde tutulmaktadır. Öğrencinin bilgi düzeyi, amacı, öğrenme metodu gibi bilgiler uzun dönemlidir. Bu bilgilerin hemen değişmesi beklenmez. Öğrencinin bir oturumdaki bilgileri ise kısa dönemlidir. Değişmesi beklenen ve sürekli takip edilen davranışlardır. Öğrenciyi tanımak için toplanan bilgiler öğrencinin öğrenme programına uyumluluğu ve problem çözümünü belirlemede yardımcı olur [4,5].

2.1.3. Öğretim modülü

Öğrenme süreçlerinin bir modelidir. Öğrenci modeli modülünden sağlanan bilgiler doğrultusunda uzman modülü kararı ile her bir öğrencinin hangi metot ve taktikleri kullanacağı belirlenir. Bu bilgiler doğrultusunda materyalleri öğrenciye göre sunar. Bilgilerin ne zaman ve ne kadar sürede sunulacağı, yeni bir konuya ne zaman geçileceği, hangi konuların sunulacağı ve hata mesajlarının uyarlanabilir sunumu bu modül tarafından kontrol edilir ve öğrenciler yönlendirilir. ZÖS kendi öğretim modülüne göre öğretimi

gerçekleştirir. Öğretim sürecini kontrol eder, öğrenciye geri dönüş sağlar ve gerekli seçimleri yapar. Kısacası bir eğitimci gibi müdahalede bulunur [4, 5].

Bu modülde bir problem çözümünde kullanılan stratejiler öğrencilerin ilerleyebilmeleri için adım adım sunulmaktadır. Böylece öğrencinin hangi adımlarda zorlandığı kolaylıkla tespit edilir ve gerekli seçimler yapılır. Adımlar her adımda zorlaşarak ilerlemesi öğrencinin hangi seviyede olduğunu kolaylıkla öğrenir [20].

Bu modülde kullanılan mekanizma öğrencilerin öğretime uyumluluğu her öğretim stilleri için verilen puanlama ile belirlenir. Belirtilen materyaller kategoriye ayrılarak hangi öğrenme stillerinin daha iyi sonuç elde ettiği tespit edilir [21].

Öğretim modülünün gerçekleştirilmesinde iki model vardır. Bunlar gizli (secretive) ve yönetimsel (executive) modellerdir. Gizli yaklaşımda öğrenciye bir dizi soru sorularak kendi hatalarını bulmaya yöneltir. Böylece öğrencinin yanlış anlamalarını düzeltmesi sağlanır. Yönetimsel yaklaşımda ise öğrencinin çeşitli problem çözme yollarını keşfetmesi sağlanır. Öğrenciyi analiz eder ve alternatif metotlar önerilir [4,5].

2.1.4. Kullanıcı arabirim modülü

Öğrenci ile ZÖS arasındaki iletişim ve etkileşimi sağlayan, diyalogları ve ekran planlarını kontrol eden modüldür. Öğrenci için uygulama içerisinde iyi bir gezinme sağlamalıdır. Kullanıcı arabirimi öğrenciler için kolay ve rahat bir kullanım sağlamayı amaçlar. İyi bir arayüzün kullanılabilirliği yüksek olmalıdır. Her zaman her yerde uygulamaların kullanılması için erişilebilirliği önemlidir.

Arayüz modülü bütün bileşenleri ve kullanıcılar arasındaki iletişimi sağlayan temel bileşendir. Materyallerin etkin bir biçimde sunulması önemlidir. Aksi takdirde engelleyici bir öğretim sunar ve öğrencinin motivasyonunu düşürür. Zengin içeriğe sahip bir arayüz her zaman ilgi çekici olmaktadır. Sunulan bilgiler ses, resim ve video gibi araçlarla desteklenmelidir. Ayrıca bütün duyu organlarına hitap edecek şekilde işitsel, görsel ve dokunmatik olarak tasarlanmalıdır [4, 6, 22]. Günümüzde arayüzlerin girdi araçları klavye ve fare dışında dokunmatik ekran, ses tanıma gibi teknolojilerin gelişmesiyle kullanılabilirliği artmaktadır [5].

Kullanıcı arabirim modülü iletişim için üç tip bilgiye ihtiyaç duyar. Bunlar, sözlü ifade aktivitelerinde konuşmayı anlamak için gerekli olan açıklamalar, iletişim içeriği için gerekli alan bilgisi ve iletişim amacı için gerekli olan bilgidir [4].

Bahsedilen bütün modüllerin kendi başlarına iyi olması yeterli değildir. Sonuçta zeki bir davranış sergileyebilmeleri için birbirleri arasında iletişim güçlü olması ve bir bütün olarak geliştirilmesi gerekmektedir. Bu modüllerin birbiri ile etkileşimli ve sistemi çalışması zeki öğretim sisteminin iyi bir yapıya sahip olmasını belirler.

2.2. Zeki Öğretim Sistemlerinde Kullanılan Yapay Zeka Teknikleri

Zeki Öğretim Sistemleri öğretimi kolaylaştırmak amacıyla bilgisayar programlarında yapay zeka teknikleri kullanan öğrenme sistemleridir [4]. Yapılan araştırmalar sonucunda, eğitim alanında, Uzman sistemler (Expert System), Karar Ağaçları (Decision Trees), Sinir Ağları (Neural Networks), Veri Madenciliği (Data Mining), Ontoloji (Ontology) gibi yapay zeka tekniklerinin kullanıldığı görülmüştür. Her birinin gerçekleştirilmesi için çeşitli yöntemlerle geliştirilmiş hazır algoritmalar ya da araçlar bulunmaktadır. Burada önemli olan bu algoritma ve araçlardan bir ya da birkaçını, Zeki Öğretim Sisteminde gerçekleştirilecek olan uygulamalara dâhil edebilmektir [6].

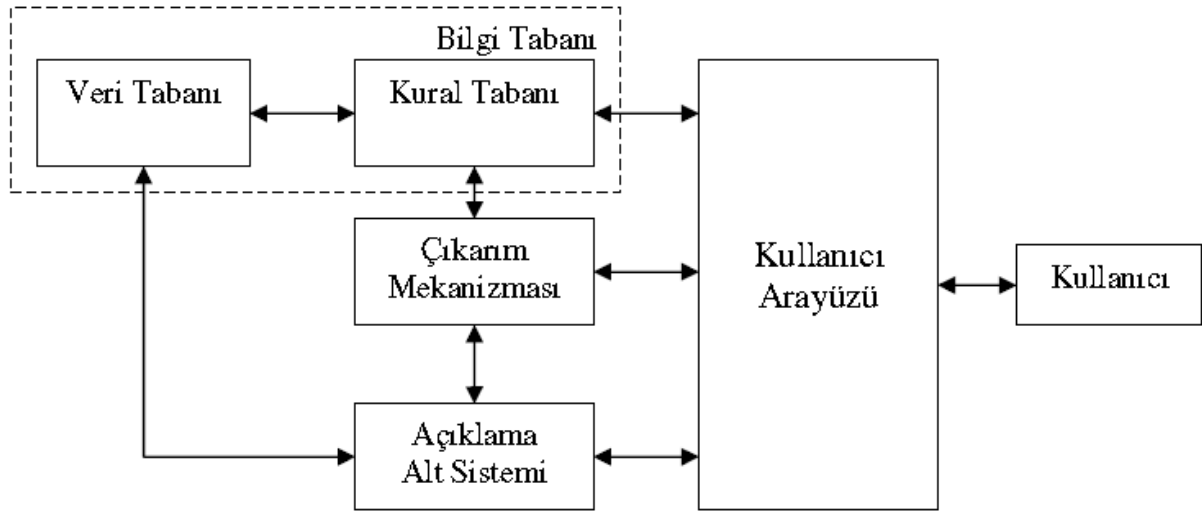
2.2.1. Uzman sistemler

Uzman Sistemler (US), herhangi bir alanda uzman bir kişinin uzmanlığını, bilgisini, tecrübesini ve düşünce şeklini ve açıklamalarını aynen yansıtan, olaylara uzman yaklaşımıyla çözüm ve öneriler getirebilen bilgisayar programıdır. Bilgi tabanlı sistemler “Knowledge Based Systems” olarak ta bilinirler [6].

İnsan yerine bilgisayara aktarılmış bilgi kullanılmaktadır. Bilgi ve çıkarımlara dayanan bir sisteme sahiptir. Algoritması ve veritabanındaki bilgi büyüklüğü performansını oldukça etkiler. Veritabanındaki bilgi sayesinde mantıksal çıkarımlar ile sonuca varmaktadır. Bir nevi uzman kişileri taklit eden yazılımlardır.

Bir uzman sistem genel olarak, bilgi tabanı, çıkarım mekanizması ve kullanıcı arayüzü elemanlarından oluşmaktadır [5]. Bir uzman sistemin yapısı Şekil 2.2’de gösterilmiştir.

Bilgi tabanında veritabanının büyüklüğü önemlidir. Kural tabanında bilgiler “Eğer-O halde” şart cümlesi ile düzenlenir. Doğru değerlendirme sağlanması için bilgiler arası bağlantı önemlidir. Çıkarım mekanizması bilgi ve kuralların çözümlemesini sağlar. Bunu çıkarım ve kontrol ile sağlar. Çıkarım “eğer-o halde” yapısını kullanarak bilgi tabanında uygun verileri getirirken kontrol ise bu verileri değerlendirerek bir sonuca varır. Kullanıcı arayüzü, uzman sistem ile kullanıcının iletişimini sağlar.



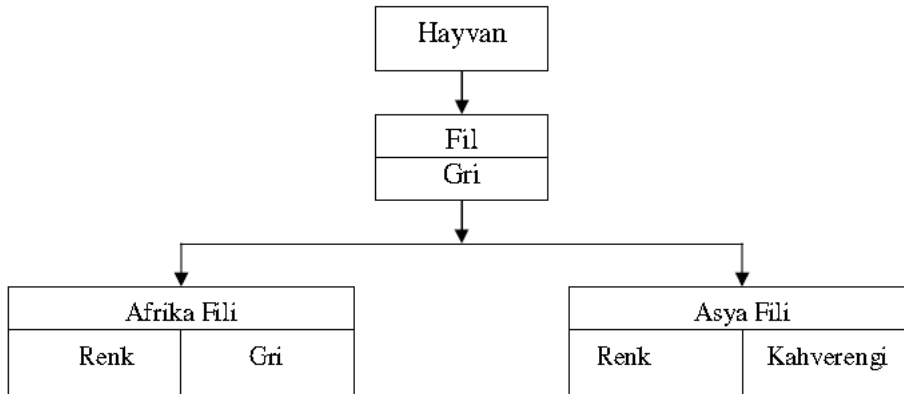
Şekil 2.2. Uzman sistemin yapısı [5]

Uzman Sistemler ile bilgi “Eğer-O halde” kuralı, çatı yapısı, semantik ağ yapısı gibi çeşitli şekillerde sunulmaktadır [23].

Eğer	Giriş ıslak ve mutfak kuru
O halde	Su kaçağı banyoda
Eğer	Giriş ıslak ve banyo kuru
O halde	Problem mutfakta
Eğer	Pencere kapalı veya yağmur yok
O halde	Dışarıdan su girişi yok
Eğer	Problem mutfakta ve dışarıdan su girişi yok
O halde	Su kaçağı mutfakta

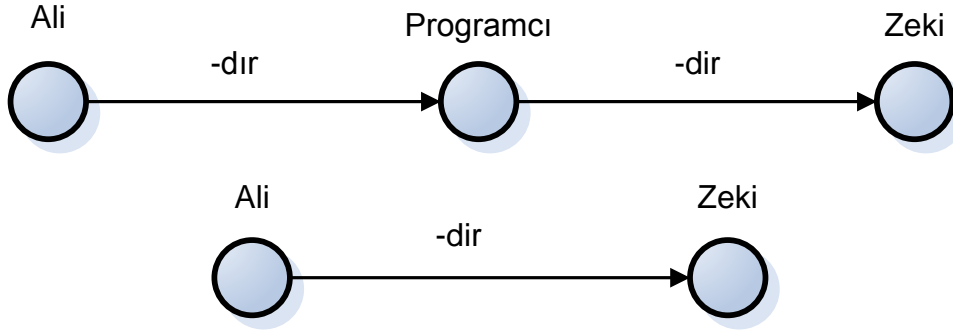
Şekil 2.3. Bilginin “Eğer-O halde” kuralı sunumuna ait örnek [23]

“Eğer-O halde” kuralı şartlı ifadelerden oluşur ve kullandığı mantıksal ilişkiler ile bir sonuca ulaşmaktadır. Şekil 2.3’de “Eğer-O halde” kuralına ait bir örnek verilmiştir. Bu örnekte bilgi tabanından bir dairedeki su kaçağını tespit etmesi gösterilmiştir.



Şekil 2.4. Bilginin çatı tabanlı sunumuna ait örnek [23]

Çatı tabanlı sunumda nesnelere ve özellikleri tutulmaktadır. Özelliklerin bulunduğu kısma slot adı verilir. Nesnelere ilişki kurularak hiyerarşik bir yapı oluşturulur. Şekil 2.4’te gösterilen çatı tabanlı sunuma ait örnekte Afrika fili renk olarak fil nesnesinde gelen gri rengi kalıtım ile alırken, asya fili kendi değeri olduğu için kalıtımla gelen rengi değil kendi değeri olan kahverengi geçerli olmuştur.



Şekil 2.5. Bilginin semantik ağ sunumuna ait örnek [23]

Semantik ağ sunumunda kullanılan cümlelerde bir sentaks ve bir de semantik vardır. Böylece nesnelere arası ilişkiler anlamlı bir şekilde gösterilir. Şekil 2.5'te semantik ağ sunumuna ait örnekte Ali programcıdır ve programcı zekidir ilişkilerinden Ali zekidir ilişkisine varılmıştır.

Uzman Sistemler sağlık, jeoloji, kimya, askeri gibi alanlarda kullanımı mevcuttur [24]. Uzman sistemlerin eğitim alanında kullanımı oldukça yaygındır. Eğitimin planlanması ve karar verme amacıyla uygulamalar geliştirilmiştir [5]. Açıklama yapma özelliği yeterli düzeyde bilgi sahibi olmayan kullanıcılara, karşılaştıkları sorunların çözümleri ile ilgili bilgiler vererek özellikle öğrencilerin eğitimini sağlaması bakımından önemlidir [23].

Uzman sistemler ZÖS'ne özellikle uzman modülü için uyarlanılarak kullanılabilir.

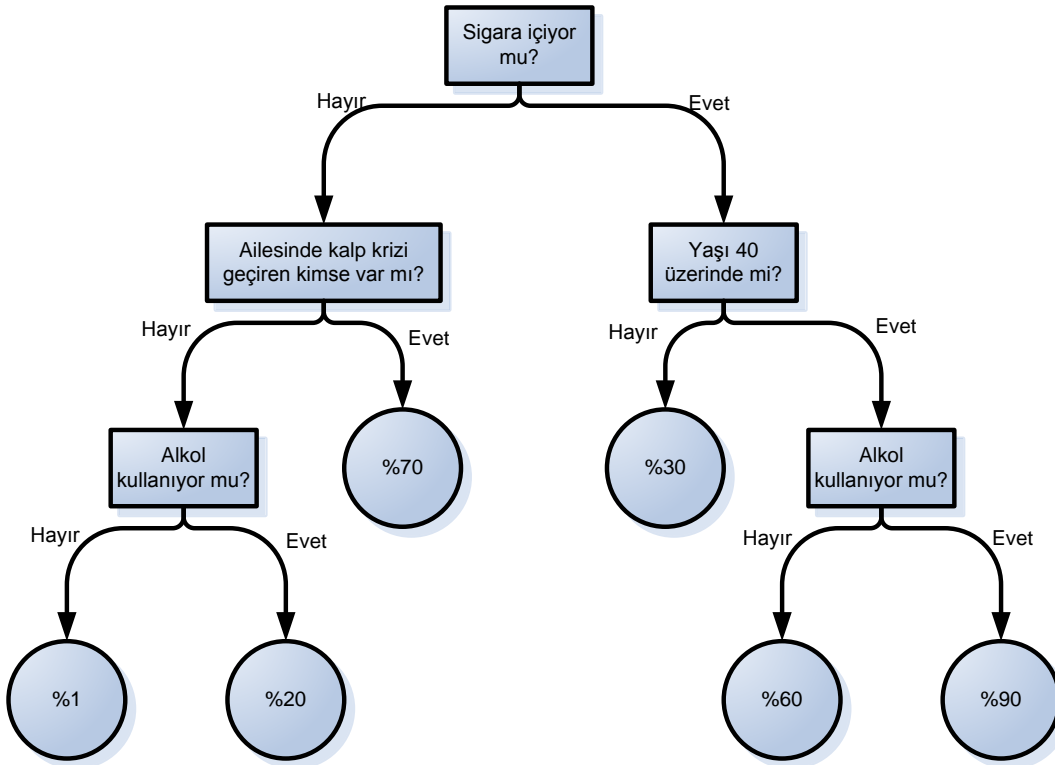
2.2.2. Karar ağaçları

Karar ağacı (KA), böl ve yönet yaklaşımını gerçekleyen ağaç yapılı bir veri yapısıdır. Hem sınıflandırma, hem de bağlanımda kullanılacak etkin bir yöntemdir ve girdi dağılımından bağımsızdır [25].

Karar ağacı öğrenmesi yöntemi, makine öğrenmesi (machine learning) konularından birisidir. Literatürde karar ağacı öğrenmesinin alt yöntemleri olarak kabul edilebilecek sınıflandırma ağacı (classification tree) veya ilkelleştirme ağacı (regression tree, tahmin ağacı) gibi uygulamaları vardır. Karar ağacı öğrenmesinde, bir ağaç yapısı oluşturularak ağacın yaprakları seviyesinde sınıf etiketleri ve bu yapraklara giden ve başlangıçtan çıkan kollar ile de özellikler üzerindeki işlemler ifade edilmektedir. Karar ağacı öğrenmesi sırasında, öğrenilen bilgi bir ağaç üzerinde modellenir. Bu ağacın bütün iç düğümleri (interior nodes) birer girdiyi ifade eder [26,27].

Şekil 2.6’da karar ağacı öğrenmesine ait bir örnek gösterilmiştir: Şekilde, kalp krizi geçiren kişilerin çeşitli özellikleri ele alınarak veriler üzerinden öğrenilmesi durumunda hayali olarak elde edilecek sonuçlardır. Diğer bir deyişle, iç düğümlerde (dikdörtgenler) çeşitli girdilere göre ağacın dallanması söz konusudur. Yapraklarda ise sonuç olarak elde edilen değerler gösterilmiştir [26].

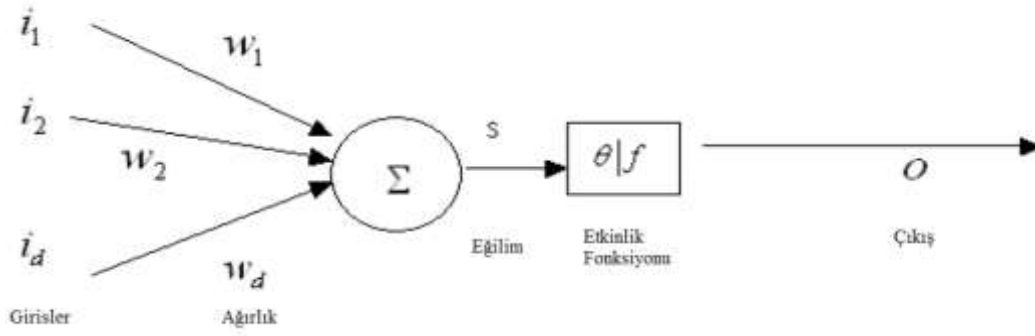
Karar ağacı öğrenmesinde, ağacın öğrenilmesi sırasında, üzerinde eğitim yapılan küme, çeşitli özelliklere göre alt kümelere bölünür, bu işlem, özyineli olarak (recursive) tekrarlanır ve tekrarlama işleminin tahmin üzerinde bir etkisi kalmayana kadar sürer. Bu işleme özyineli parçalama (recursive partitioning) ismi verilir [26].



Şekil 2.6. Karar ağacı öğrenmesine ait örnek [26]

2.2.3. Yapay sinir ağları

Yapay Sinir Ağları (YSA), insan beyninden esinlenerek geliştirilmiş, ağırlıklı bağlantılar aracılığıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır [28]. YSA yeni bilgiler türetebilme, keşfedebilme gibi yetenekleri herhangi bir yardım almadan gerçekleştirmeyi amaçlar [29].



Şekil 2.7. Sinir ağının yapısı [30]

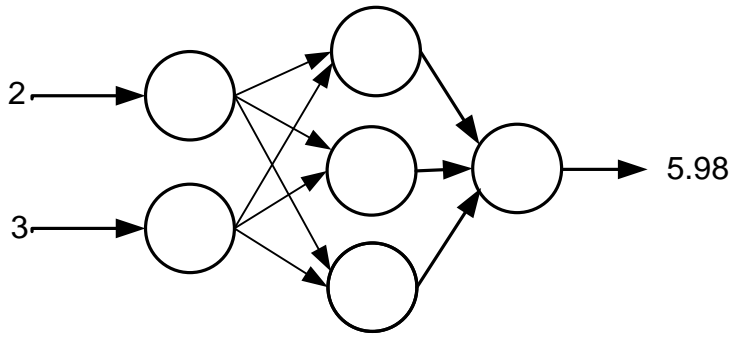
Sinir Ağlarının yapısı Şekil 2.7’de görüldüğü gibi giriş, ağırlık, etkinlik fonksiyonu ve çıkış birimlerinden oluşur. Girişlere gelen bilgiler, ağırlık ile etkinlikleri belirlenir. Her bir ağırlık ve giriş bilgisi çarpılır ve sonuçlar etkinlik işlevinde geçirilerek sonuca aktarılır.

Sinir ağları bu yapısıyla ses tanıma, el yazısı tanıma, karakter tanıma, parmak izi tanıma gibi alanlarda sıkça kullanıldığı gibi öğrenme yöntemleri sayesinde eğitim alanında da kullanılmaktadır. Öğrenme yöntemlerine göre danışmanlı öğrenme ve danışmansız öğrenme olarak ikiye ayrılır. Danışmanlı öğrenmede istenilen çıkış değerleri elde edilene kadar ağırlıklar yeniden hesaplanır. Danışmansız öğrenmede ise çıkış hakkında bilgisi yoktur ve girişlere göre kendi kendisini örnekler. Örneklerden elde ettiği bilgiler ile kendi deneyimlerini oluşturur ve daha sonra benzer konularda benzer kararlar verirler. Yani amaç ezberletme değil sistemin kendi kendine öğrenmesidir.

YSA’ların eğitilmesi ve çalışmasına ilişkin yapılan bir araştırmada çarpım tablosunun 2 ile çarpım yapması öğretilmek istenmektedir. Çalışmada problem uzayı toplam 10 örnekten oluşmaktadır. Eğitim seti ve test setini belirlerken Çizelge 2.1’de gösterildiği gibi çift sayılar ile çarpım eğitim seti, tek sayılar ile çarpım test seti olarak seçilmiştir. Başarılı sonuç verebilmesi için örnek dağılımı önemlidir. Böylece sisteme 2 ile çift sayıların çarpımı öğretilerek, 2 ile tek sayıların çarpımı istenmektedir. Şekil 2.8’de iki çarpım tablosunu öğrenen ağın yapısı gösterilmiştir. Tasarlanan bu ağda 200 iterasyonluk öğrenme sonunda 2x3 sorusuna 5,98 cevabı vermiştir. %0,2’lik bir hata ile doğru sonucu bularak başarılı bir öğrenme gerçekleştiği görülmüştür [29].

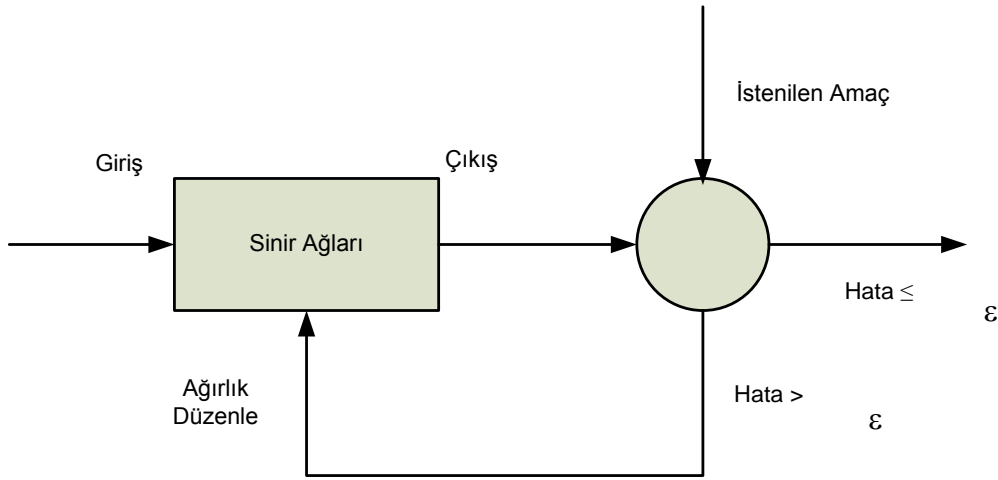
Çizelge 2.1. İki ile çarpım örnekler ile eğitim seti ve test seti örnekleri [29]

2 ile çarpım için örnekler	Eğitim Seti		Test Seti
$2 \times 1 = 2$	$2 \times 6 = 12$	$2 \times 2 = 4$	$2 \times 1 = 2$
$2 \times 2 = 4$	$2 \times 7 = 14$	$2 \times 4 = 8$	$2 \times 3 = 6$
$2 \times 3 = 6$	$2 \times 8 = 16$	$2 \times 6 = 12$	$2 \times 5 = 10$
$2 \times 4 = 8$	$2 \times 9 = 18$	$2 \times 8 = 16$	$2 \times 7 = 14$
$2 \times 5 = 10$	$2 \times 10 = 20$	$2 \times 10 = 20$	$2 \times 9 = 18$



Şekil 2.8. İki ile çarpım tablosunu öğrenen ağ [29]

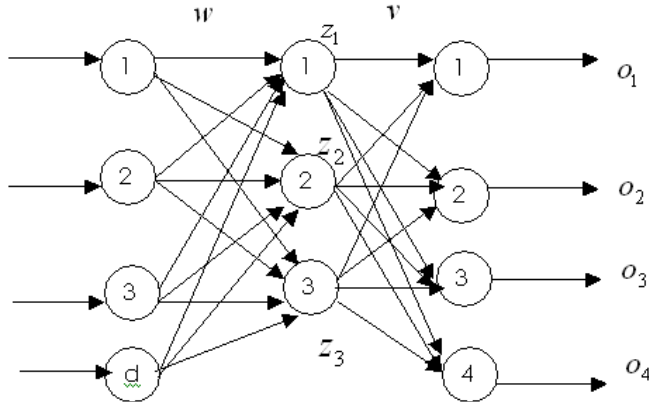
Evrım algoritmaları büyük kompleks problemlerin çözümünde faydalıdır [31]. Eğitim alanında bilgi tabanı şekillendirilmesinin karmaşıklığı için etkili olacaktır.



Şekil 2.9. ZÖS ile sinir ağlarının kullanımı [30]

ZÖS'ne uyarlanmış bir sinir ağları tabanlı sistemlerde konular düğümler ile kavramsal haritalama ya da semantik ağ yapısında sunulur. Her bir düğüm pedagojik kaynakla ilişkilendirilir ve öğretim modülü oluşturulur. Sistemin merkezi her bir öğrenciye özgü öğretim modülü sağlamayı amaçlayan sinir ağlarıdır. Sinir ağları belli bir çıkış elde eden giriş vektörlerini eğitir. Şekil 2.9' da ZÖS ile Sinir Ağlarının kullanımı gösterilmiştir. Çıkış katmanı en fazla performans davranışı olan öğretim modeli ile ilişkilendirilmiştir. İstenilen amaç ile uygun öğretim modülü belirlenir. Belirlenen öğretim modülü her öğrenci için giriş düğümlerini gönderilir. Gizli katmanda giriş ve çıkış sayısı kadar seçim olur. Örneğin;

$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ görsel öğrenme stilini, $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ işitsel öğrenme stilini, $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ kinetikselsel öğrenme stilini temsil eder.



Şekil 2.10. ZÖS'ne uyarlanmış sinir ağı mimarisi [30]

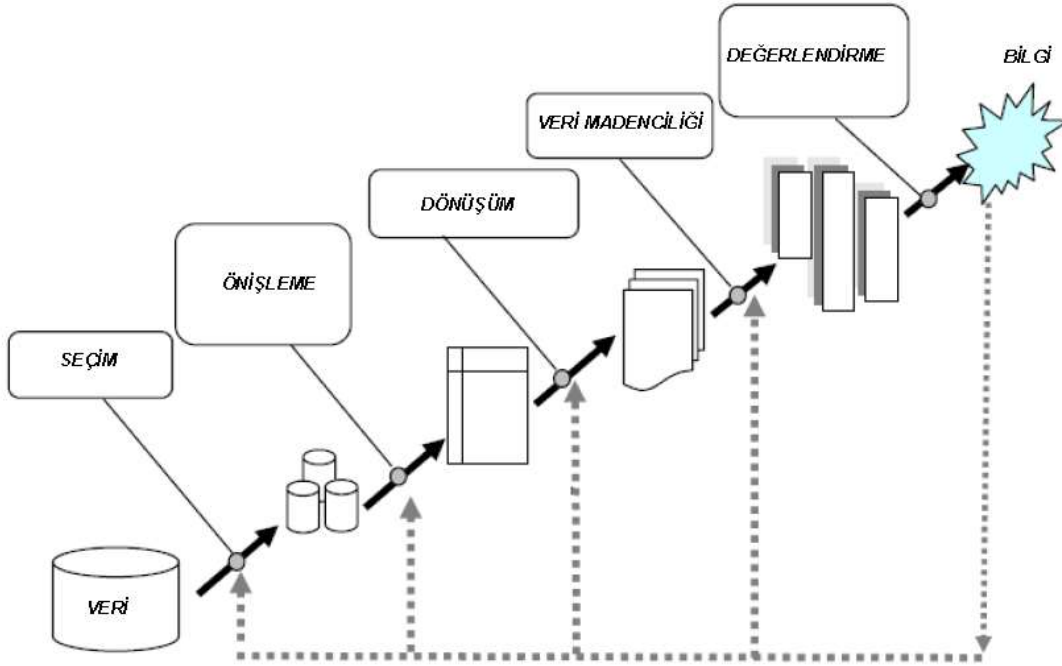
Çıkış bilgilerine göre hata hesaplanır ve eğer istenilen amaç sağlanırsa öğrenme gerçekleşmiş kabul edilir, sağlanmazsa tekrar eğitime alınır [30].

2.2.4. Veri madenciliği

Veri madenciliği daha önceden bilinmeyen, geçerli ve uygulanabilir bilgilerin geniş veritabanlarından elde edilmesi ve bu bilgilerin işletme kararları verirken kullanılmasıdır [27].

Veri madenciliği bir nevi bilgi keşfi yaparak verileri anlamlandırmaktadır. Bu süreç Şekil 2.11'de görüldüğü gibi aşamalardan oluşmaktadır. Seçim aşamasında veri birçok kaynaktan seçilir. Önişleme aşamasında seçilen veriler düzenlenir ve eksik veriler çıkarılır, dönüşüm aşamasında kullanılacak algoritma için daha uygun hale getirilir, veri madenciliği aşamasında kullanılan algoritma gerçekleştirilir, değerlendirme aşamasında ise elde edilen analiz sonuçları incelenir [32, 33].

Veri madenciliği teknikleri, davranış örnekleri tahmini, daha çok etkileşimli olması, sonuçları görselleştirme ve daha gerçek zamanlı veri analizinden dolayı çözüm olmada büyük bir ihtiyaçtır. Veri madenciliği, veriden anlamlı bilgi elde etmek için araştırma ve analiz yapma işidir. Veri madenciliği pazarlama, bankacılık, sigortacılık, tıp gibi birçok alandaki verilerin analiz edilmesinde kullanılır. Eğitim alanında da kullanılarak analiz sonucu doğru ve anlamlı bilgilerin oluşturulması sağlanmaktadır [32, 33].



Şekil 2.11. Veri tabanında bilgi keşfi süreci [32]

Zeki öğretim sisteminde veri madenciliği kullanılarak öğrencilerin not dağılımları, doğru ve yanlış cevap verdikleri sorular, ders konularına göre aldıkları puanlar gibi çeşitli veriler analiz edilerek zeki öğretim sistemleri için öğretim stratejileri belirleme, öğrencilerin kişisel başarıları, öğrenme düzeyleri gibi birçok konuda yardımcı olmaktadır.

2.3. ANLAMSAL WEB

Web doksanlı yıllarda hayatımıza girdi. Web'in ilk nesli Web 1.0 (1995-2000) yalnızca HTML belgelerin yer aldığı statik bir yapıya sahiptir. Web 2.0 (2000-2010) ile birlikte kullanıcılarının da aktif olarak katılabildiği etkileşimli ve insan odaklı bir platforma dönüştü. Web' in hızla büyümesi sadece insanlar tarafından değil bilgisayarlar tarafından da anlaşılmaya başlanması gerekliliğini doğurdu ve anlamsal web fikri ortaya atılmıştır [34].

Anlamsal web, WWW konsorsiyum (W3C)'u tarafından yürütülen ve şu anda kullanılmakta olan web mimarisini, verinin uygulamalar, kurumlar ve topluluklar arasında paylaşarak ve tekrar kullanımını artırarak, geliştirmeyi amaçlayan bir çalışmadır. Anlamsal web fikrini ortaya atan ilk makale Scientific American dergisinde 2001 yılında yayınlanmıştır. Bu makale, anlamsal web'i, şu anda kullanılan web'den ayrı bir web oluşumu değil, onun bir uzantısı olarak tanımlar [35, 36].

Anlamsal web, verileri anlamlaştırır. Bilgisayarın çıkarım yapmasını sağlayacak bir yapıyı oluşturur. Sadece verileri anlamlaştırılmaz aynı zamanda verileri birbirine bağlar. Bu tüm web ortamında gerçekleştiğinde oldukça güçlü bir yapı halini alır. Anlamsal web' in amacı da bu bağlantıyı kurabilmektir. Her hangi bir alanda web de dağınık ve farklı düzeylerde birçok dokümana erişilebilir. Bu dokümanları çözmek oldukça karmaşıktır. Anlamsal web bu karmaşıklığı gidermek için verilerin paylaşılmasını ve birbiriyle bağlantı kurulmasını sağlar. Yani bizim yerimize makinelerin bilgi keşfine çıkmasına olanak sağlar. Böylelikle ortak bir dokümantasyon havuzuna sahip olunmaktadır.

Günümüzde arama motorları ile bir bilgi sorgulanırken herkes için aynı sonuçları getirmektedir. Bir konu araştırması yapılması durumunda ilköğretim ödevini yapan bir ilköğretim öğrencisine de akademik araştırma yapan bir araştırma görevlisine de aynı sayfaların gelmesi çok anlamsızdır. Ya da düğüm konusu arayan bir bilgisayar bölümü öğrencisine veri yapıları dersinde ki düğümler yerine halat ile düğüm atma sayfaları gelmesi çok olasıdır. Anlamsal web arama motorları bu durumları ortadan kaldırmaktadır.

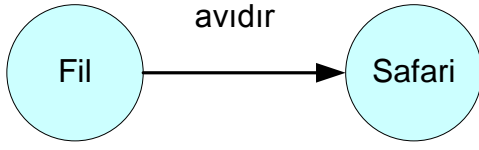
2.3.1. Bağlı veri

Anlamsal web küresel ölçekte akıllı bir veri tabanına dönüşmeyi sağlamak için W3C tarafından geliştirilen bir Bağlı Veri (Linked Data) kavramı sunar. Bağlı veri ile her bilginin belirli bir anlama sahip olacak şekilde modellenmesi ve modellenen bu bilgilerin birbiriyle ilişkilendirilerek birbirine bağlanması sağlanmaktadır. Bir nevi üç boyutlu bir gezinim oluşturur [34, 37].

Bağlı veri sayesinde;

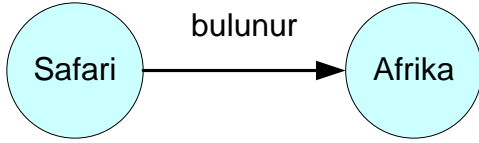
- Bilgilere anlam yüklenebilir.
- Anlamlandırılmış bu bilgiler arasında bağlantı kurulabilir.
- Bilgiler arasında kurulan bu bağlantılar sayesinde otomatik olarak yeni bilgi ve sonuçlara ulaşılabilir [38].

Şekil 2.12'de görüldüğü gibi bağlı veri fil ve Safari bilgilerine “Fil bir Safari avidir” anlamını yükler.



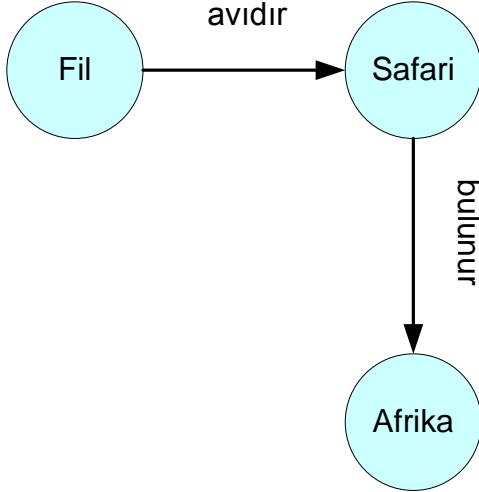
Şekil 2.12. Bağlı veri anlamlandırma örneği birinci adım

Şekil 2.13'te görüldüğü gibi bağlı veri Safari ve Afrika bilgilerine de “Safari Afrika’da bulunur” anlamını yükler.



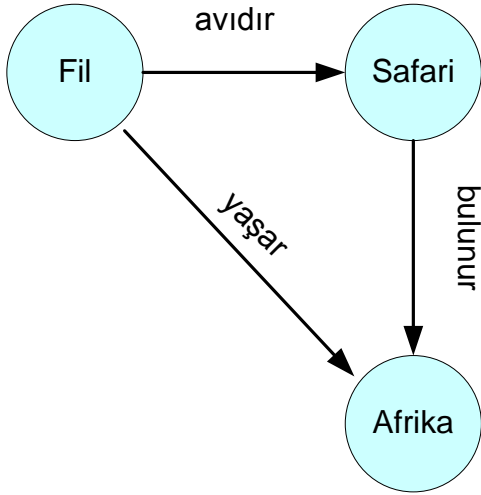
Şekil 2.13. Bağlı veri anlamlandırma örneği ikinci adım

Bağlı veri anlamlandırılmış bu bilgiler arasında ortak Safari bilgisine göre bağlantı kurar. Şekil 2.14’te bu bağlantı gösterilmiştir.



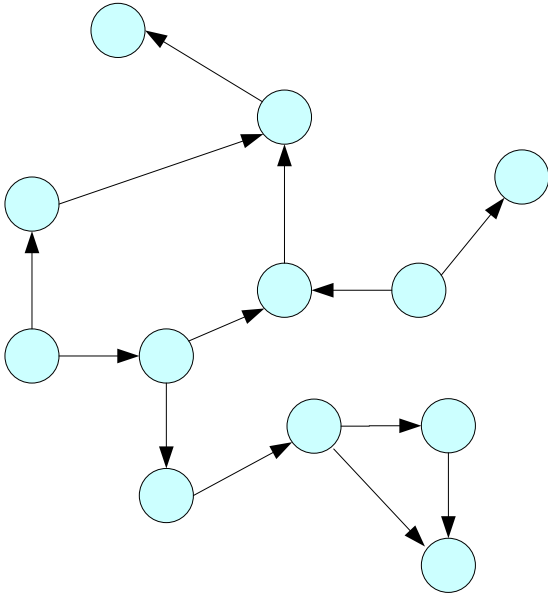
Şekil 2.14. Bağlı veri anlamlandırma örneği üçüncü adım

Bağlı veri bu bağlantılar sayesinde otomatik olarak “Fil Afrika’da yaşar” sonucunu ortaya çıkarır. Şekil 2.15’te bağlantının son hali gösterilmiştir.



Şekil 2.15. Bağlı veri anlamlandırma örneği dördüncü adım

Anlamlandırılmış bilgiler ile bağlantı kurarak genişleyen bağlı veri Şekil 2.16'da ki gibi gezinim yapılan bir veri kümesi haline gelir.

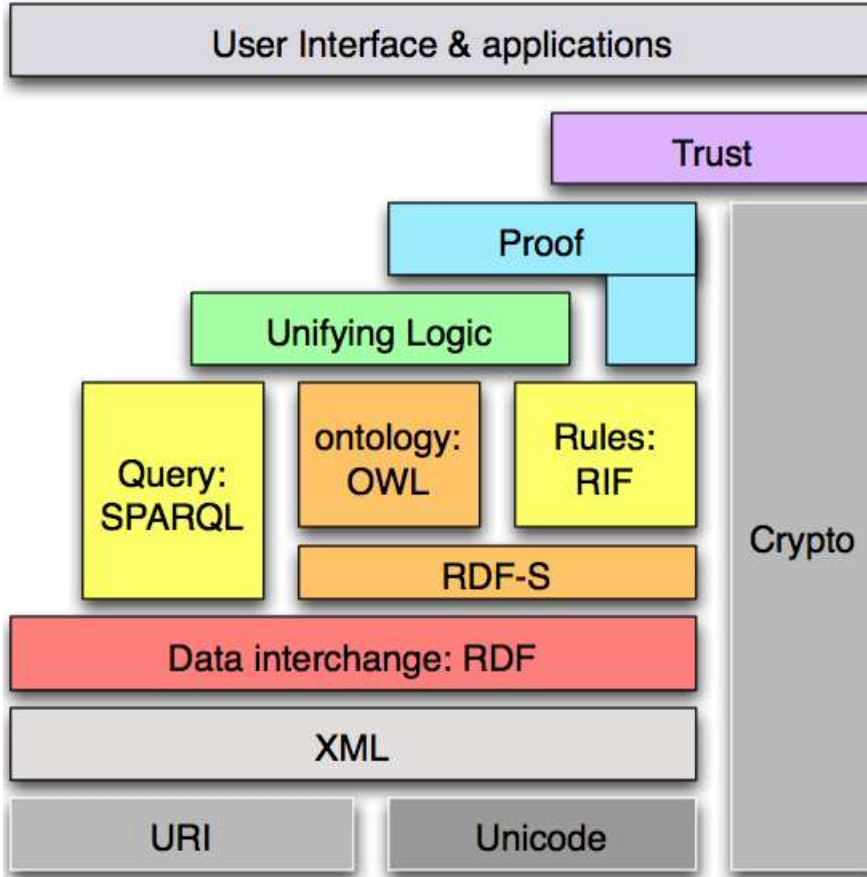


Şekil 2.16. Veri kümesi

2.3.2. Anlamsal web mimarisi

Anlamsal web yapısal olarak etkileşimli çalışma grupları ve yardımcı teknolojilerden oluşur [39]. Şekil 2.17'de anlamsal web' in mimarisi gösterilmiştir. Anlamsal web oluşturmak için çeşitli diller kullanılmaktadır. Bu diller temelinde adres bilgisi tutan URI (Uniform Resource Identifier)'leri tanımlar. XML (eXtensible Markup Language), RDF (Resource Description

Framework) ve OWL (Web Ontology Language) gibi diller ile ontoloji oluşturulur. Bu ontolojiden hızlı sonuçlar ve çıkarımlar almak için SPARQL sorgulama dili mevcuttur. Her katmanda güvenliğin sağlanabilmesi için crypto (şifreleme), proof (ispat) ve trust (güven) kullanılmaktadır. Anlamsal web bu araçlar ile kullanıcıya çok az iş düşen uygulamalar sunmaktadır.



Şekil 2.17. Anlamsal web mimarisi [38]

Anlamsal web eğitim alanında kullanımı kaçınılmazdır. Anlamsal web uzman alanları için sistematik bir yapı sağlar. Bu yapının ilerlemesini destekler ve karmaşık verilerden iyi çıkarımlar sağlar. E-öğrenme ortamları kişiselleştirmeyi amaçladıkları için anlamsal web ile bu ortam sağlanabilmektedir. Her bireyin tanımlanması için anlamsal web güçlü araçlar içermektedir.

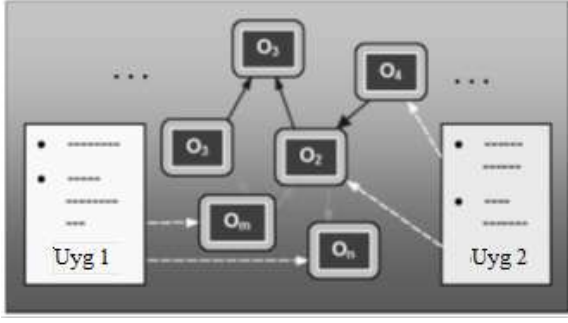
2.3.3. Ontoloji

Bilgi tabanı son derece karmaşık bir yapıya sahiptir. Bilgi tabanına veri girişi ve bu veriye erişim oldukça kolaydır. Fakat bu verileri etkili bir şekilde kullanmak için yeterli değildir. YZ teknikleri ile bilgi tabanındaki verilerden çeşitli yöntemlerle anlam çıkartılmaya çalışılmıştır. Bilgiler üzerinde kurallar, ilişkiler, bilgi toplama gibi işlemler yapılmıştır. Ontoloji ise sadece bilgi değil bilginin yapısını da kullanarak anlamsal ve mantıksal bir bilgi tabanı oluşturmaktadır. Böylece karmaşık yapıyı çözümleyerek makinelerle insanlar arasında ortak bir bilgi çatısı elde etmektedir.

Ontoloji verileri anlamlaştırarak kavramsallaştırma sağlar. Ontoloji kavramlar arasındaki ilişkiler hakkında makinalar tarafından çevrilip kullanılmasıdır [14]. Böylece makinelerin tıpkı insanlar gibi anlamalarıdır. Makineler insanın yükünü alarak az bir çaba ile işlemler yapabilmektedir.

Ontolojide bilgi yapısı, terminolojik olarak sunulur. Yorumlar ve kısıtlamalar kullanan kural setleri ile bilgi içeriğinden tasvir edilir. Varlıklar sınıf ve alt sınıf olarak hiyerarşik bir şekilde düzenlenir. Bu sınıf ve alt sınıflara birkaç özellikler ve kısıtlamalar tanımlanarak varlıklar en uygun sınıflara ayrılır. Bu yapısı ile ontoloji, kavramların temelini özetler, kategorilere ayırır, nesne ve aralarındaki ilişki türlerini tanımlar, iletişimi kolaylaştırır, paylaşım ve bilginin tekrar kullanım sağlar [13, 15].

Ontolojilerin oluşturulması oldukça zahmetlidir. İlgili alan ile daha önceden oluşturulmuş bir ontoloji mevcutsa ontolojinin yeniden oluşturulmasında ziyade mevcut ontolojiyi kullanmak hatta üzerinde değişiklik yapılarak geliştirilmesi daha çok tercih edilir. Semantik web teknolojileri ile bilgiye birlikte çalışabilirlik sağlar [12]. Bu amaçla ontolojilerin portal ve web servisleri olarak kullanımı yaygındır. Ontolojiler birden çok ontoloji ile iletişime geçerek kullanılabilirler. İki ya da daha fazla ontoloji tabanlı ZÖS ile bilgi tabanlarını paylaşan uygulamalar mevcuttur. Her uygulama kendi bilgi ve çıkarım mekanizmalarına sahiptir. Farklı eğitim alanlarında eğitim içeriği ve hizmetlerin paylaşımı ve birlikte çalışabilirlik uygun diller kullanılarak sağlanır. Şekil 2.18'de iki web tabanlı ZÖS'nin ontolojileri paylaşımı gösterilmiştir [18].



Şekil 2.18. İki web tabanlı ZÖS'nin ontolojileri paylaşımı [18]

Ontolojiler için literatür de, ayrıntı düzeylerine göre farklı sınıflandırmalar yapılmıştır. Belirli bir göreve, bir bakış açısına veya karakteristiklerine göre ontolojileri dört sınıfa ayırmaktadır. Bu ontoloji sınıfları üst düzey ontoloji, alan ontolojisi, görev ontolojisi ve uygulama ontolojisidir [40-41].

Üst düzey ontoloji, belirli bir olay veya herhangi bir alandan bağımsız olarak en genel kavramları tanımlar [42].

Alan ontolojisi, herhangi bir alanla (tıp, hidrografiya, topografiya, ekoloji, vb.) ilgili kavramları tanımlar. Alan ontolojisi üst seviyeli ontolojide yer alan kavramları detaylı olarak tanımlar [42].

Görev ontolojisi, herhangi bir olay veya görevle ilgili kavramları üst seviyeli ontoloji kavramlarını özelleştirerek tanımlar [42].

Uygulama ontolojisi, belirli bir alana ve göreve bağlı olarak, alan ve görev ontolojilerinde yer alan ilgili kavramları özelleştirerek tanımlar [42].

Veri ontolojisi, bir veri kaynağı veya servisi tanımlayan ontolojidir. Veri ontolojisi veri tabanı şemasında yer alan kavram tanımlarını içerir [42].

Ontoloji ile verilerin makineler tarafından anlaşılabilir bir yapı sunulması için çeşitli diller kullanılarak oluşturulmaya çalışılmıştır. Bu dillerden en temelleri XML, RDF ve OWL dilleridir. Bu diller ile yapılmış örneklere bakıldığında aralarında ki farklar daha iyi görülecektir.

2.3.4. XML

Farklı platformlardaki veriler arasında standart bir dil sağlayan XML (eXtensible Markup Language) bunlardan birisidir. XML sınıflandırılmış veriler eşleştirilerek iletişim sağlamaktadır. Aynı anlama gelen farklı isimdeki yapıları eşleştiremez. Çünkü XML veriyi yapısal olarak sunar, anlamsal olarak ne ifade ettiğini bilmez.

```
<xml
  <personel>
    <ad>Emre</ad>
    <soyad>Akdemir</soyad>
    <maas>3500</maas>
  </personel>
>
```

Şekil 2.19. XML'e ait bir örnek

Şekil 2.19'da XML ile tanımlanmış bir örnek gösterilmiştir. Bu örnekte bir personele ait bazı kişisel bilgiler tanımlanmıştır. Bu tanımlamalar başka bir uygulama da bulunan tanımlar ile her zaman eşleştirilemez. Örnek olarak bu uygulamada ki maas etiketi diğer uygulamada aylık etiketine karşılık gelebilir. Bundan dolayı XML'in yeterli olmadığı görülmektedir.

2.3.5. RDF

RDF (Resource Description Framework) nenseler ve aralarında ilişkileri veri modeli yapısında sunar ve bu veri modeli için basit bir semantik sağlar. RDF kullanılarak makineler tarafından kısmen anlaşılabilen bir yapı oluşturularak makinelere yargılama ve çıkarım yapma yeteneği sağlanmıştır.

Şekil 2.20'de RDF ile tanımlanmış bir örneğin grafiği, Şekil 2.21'de ise aynı örneğin RDF/XML sözdizimi gösterilmiştir. Bu örnekte Eric Miller adlı kişinin sınıf türü personel, özellikleri ad, eposta, unvanı ve bu özelliklerin değerleri Eric Miller, em@w3.org ve Dr. olarak tanımlanmıştır. RDF her bir düğüm ve düğümler arası ilişkileri belirleyen bağlantılar (URI) içermektedir [35, 43].



Şekil 2.20. RDF ile tanımlanmış bir örneğin grafiği [43]

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>

```

Şekil 2.21. RDF ile tanımlanmış bir örnek [43]

2.3.6. OWL

Makineler ve insanların ortak bir dilde istenilen düzeyde anlaşabilmeleri için OWL (Web Ontology Language) geliştirilmiştir. OWL ile nesne ve ilişki tanımlamalarında daha fazla anlam kazandırılabilir. OWL ile makineler tanımlama mantığı kazanabilmeleri için yargılama yapabilen sistemler geliştirilmiştir [43-44].

Şekil 2.22’de OWL ile tanımlanmış bir örnek gösterilmiştir. Bu örnekte mozzarella bir pizza için peynirli pizzanın alt sınıfı olduğu, baharatlı ve ülkesi özelliğine sahip olduğu değerleri ise hafif ve İtalya olarak kısıtlandığı belirtilmiştir.

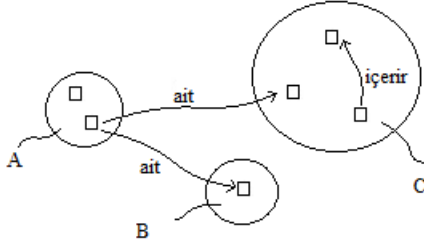
```

<owl:Class rdf:about="#MozzarellaTopping">
  <rdfs:label xml:lang="pt" >CoberturaDeMozzarella</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#CheeseTopping"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasSpiciness"/>
      <owl:someValuesFrom rdf:resource="#Mild"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasCountryOfOrigin"/>
      <owl:hasValue rdf:resource="#Italy"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Şekil 2.22. OWL ile tanımlanmış bir örnek [45]

OWL temel olarak üç elemandan oluşmaktadır. Bunlar individuals, class ve properties’lerdir. Şekil 2.23’de aralarındaki ilişkiler görülmektedir.



Şekil 2.23. Individuals, class ve properties arasındaki ilişki

Individuals: Ontolojide kullanılacak her bir örnek, direk Türkçe anlamı ile bireylerdir. Yani ontolojinin temel elemanlarıdır. Ontoloji kurmak için bu bireyleri sınıflandırıp, çeşitli özellikler ile ilişkilendirip, bazı kısıtlamalar yapılmaktadır.

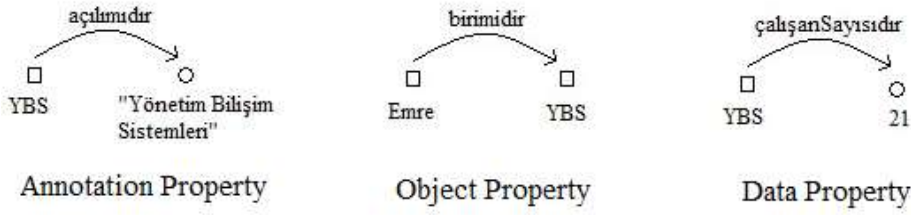
Class: Bireyleri sınıflandırmak için kullanılır. Nesne yönelimli programlamaya göre sınıflandırma işlemleri daha avantajlıdır. Çünkü bir birey istenildiği kadar sınıflarda tanımlanır. Sınıflarının alt sınıflarını hiyerarşik olarak tanımlamak çok kolay ve anlaşılır olduğundan sistemli bir sınıflandırma sağlanır.

Properties: Bireyler arasında bağlantı kurulan özelliklerdir. Bu bağlantılar isimlendirilerek sınıflar, bireyler sistemli ve anlaşılır bir şekilde ilişkilendirilir. Her özelliğin kaynağı olan bir domain ve hedefi olan bir range tanımlaması yapılır. OWL properties'ler range özelliğine göre üç türdür. Bunlar; Annotation property, Object property ve Data property'dir.

Annotation property: range olarak bir açıklamayı işaret eder. Şekil 2.24'teki ilk örnekte görüldüğü gibi *YBS* bireyinin açılımını açıklama olarak *Yönetim Bilişim Sistemleri* olarak ilişkilendirilmiştir.

Object Property: range olarak yine bir bireyi işaret eder. İkinci örnekte *Emre* adlı personelin birimi *YBS* olarak ilişkilendirilmiştir (Bkz. Şekil 2.24).

Data Property: range olarak bir veri tipi işaret eder. Üçüncü örnekte *YBS*'deki çalışan sayısı *21* olarak ilişkilendirilmiştir (Bkz. Şekil 2.24).



Şekil 2.24. Annotation property, object property ve data property

OWL Object Property çeşitli anlamlandırmalarla zenginleştirilmiş özelliklere sahiptir. Bu özellikleri; Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive' dir.

Functional properties (işlevsel özellik), belirli bir birey için, en fazla bir birey ile ilişkilendirilir [45]. Bir kişinin sadece bir biyolojik annesi, bir biyolojik babası olması işlevsel bir özelliktir. A ve B bireyleri C bireyinin *biyolojikAnnesiDir* olarak ilişkilendirildi ise A ve B bireyleri aynıdır.

Inverse functional properties (ters işlevsel özellik) olarak tanımlanmış bir özelliğin ters özelliği işlevsel olduğu anlamına gelir [45]. Bir önceki örnekte *biyolojikAnnesiDir* özelliği yerine *ninBiyolojikAnnesiDir* özelliğini kullanırsak C bireyi A ve B bireyleri *ninBiyolojikannesidir* olarak ilişkilendirildi ise A ve B bireyleri aynıdır.

Transitive properties (geçişli özellik) kullanılarak, A bireyi B bireyi ile ilişkilendirilsin, B bireyi de C bireyi ile ilişkilendirilsin. Buradan A bireyi C bireyi ile ilişkilidir anlamı çıkarılır [45]. A bireyi B bireyinin *sınıfArkadaşdır* özelliği ile ilişkilendirilsin. B bireyi de C bireyinin *sınıfArkadaşdır* özelliği ile ilişkilendirilsin. Sonuç olarak A bireyi C bireyinin *sınıfArkadaşdır* özelliği olduğu sonuca varılır.

Symmetric properties (simetri özellik) kullanılarak A bireyi B bireyi ile ilişkili ise B bireyi de A bireyi ile aynı özellik ile ilişkilidir [45]. A bireyi B bireyinin *kardeşidir* özelliği tanımlanırsa B bireyi de A bireyinin *kardeşidir* özelliği tanımlanmış olur.

Asymmetric properties (asimetrik özellik) kullanılarak A bireyi B bireyi ile ilişkili ise B bireyi de A bireyi ile aynı özellik ile ilişkili değildir [45]. A bireyi B bireyinin *annesidir* özelliği tanımlanırsa B bireyi A bireyinin *annesidir* özelliği tanımlanmaz.

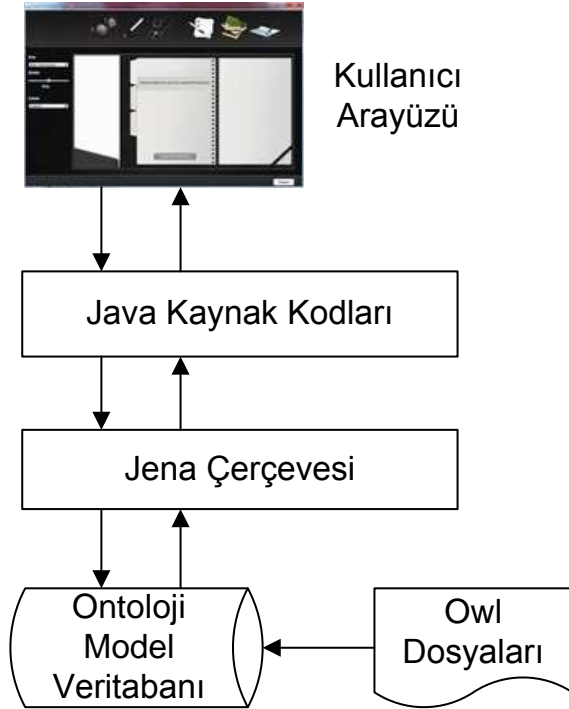
Reflexive properties (refleksif özellik) Bir bireyin kendisi ile ilişkili olduğu zaman kullanılır [45]. A bireyi B bireyinin *tanır* özelliği aynı zamanda A bireyinin kendisini de *tanır* özelliğini de kullanabilir.

Irreflexive properties (refleksif olmayan özellik) Bir bireyin kendisi ile ilişkili olmadığı zaman kullanılır [45]. A bireyi B bireyinin *annesidir* özelliği, A bireyinin kendisinin de *annedir* özelliği kullanılmaz.

3. ÖRNEK UYGULAMA

3.1. Sistem Mimarisi

Şekil 3.1’de uygulamanın yapısı gösterilmiştir. Owl dosyası bir sefer olmak üzere MySQL veritabanına aktarılmaktadır. MySQL veritabanı ontoloji modeli dışında ontolojiye dahil edilmeyen bazı verileride saklamaktadır. Bu veriler öğrencilerin takip edildiği performansları, oturum açma kapatma gibi bilgilerdir. Java Persistence Api (JPA) ile bilgiler düzenlenip kaydedilmektedir. Bu verilerin ontoloji yerine JPA ile düzenlenmesi daha kullanışlı ve performanslıdır. Öğrenciye ait her adım için saklanan performans ve oturum bilgileri ontoloji ile tasarlanması uygun değildir. Jena çerçevesinin sunduğu kütüphane ile java sınıfları ile uyumlu çalışmaktadır. Ontoloji modelinden istenilen bilgiler çekilip, düzenlenebilir ve değişiklikler kaydedilebilir. Uygulama kullanıcı girişi ile başlatılmaktadır. Öğrenci ve öğretmen için ayrı arayüzler tasarlanmıştır.



Şekil 3.1. Uygulamanın yapısı

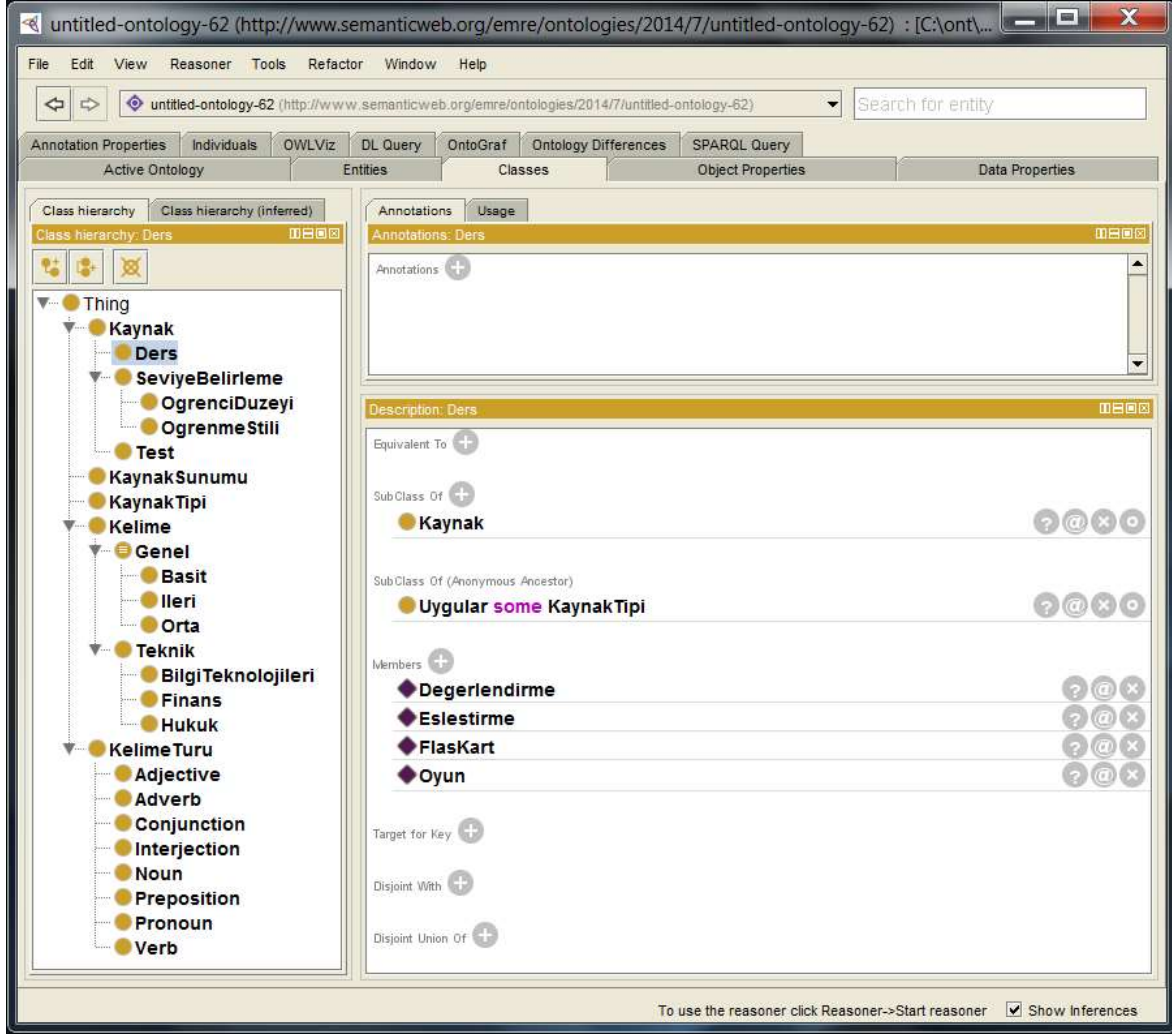
3.2. Ontolojinin Hazırlanması

Ontoloji tabanlı ZÖS mimarisi tasarlanırken modülleri destekleyen ontolojiler ve aralarındaki ilişkiler oldukça sağlam kurulmalıdır. Aksi takdirde görevini yerine getiremez. Ontoloji tabanlı ZÖS mimarisinin yapısında kullanılan ontolojiler uygulamalara ve tasarlayanlara göre farklılık gösterebilir. Bu çalışmada kullanılan ontolojiler aşağıda belirtilmiştir.

- Alan Ontolojisi (Domain Ontology)
- Öğrenci Modeli Ontolojisi (Learner Model Ontology)
- Öğretim Strateji Ontolojisi (Teaching Strategy Ontology)
- Arayüz Ontolojisi (Interface Ontology)

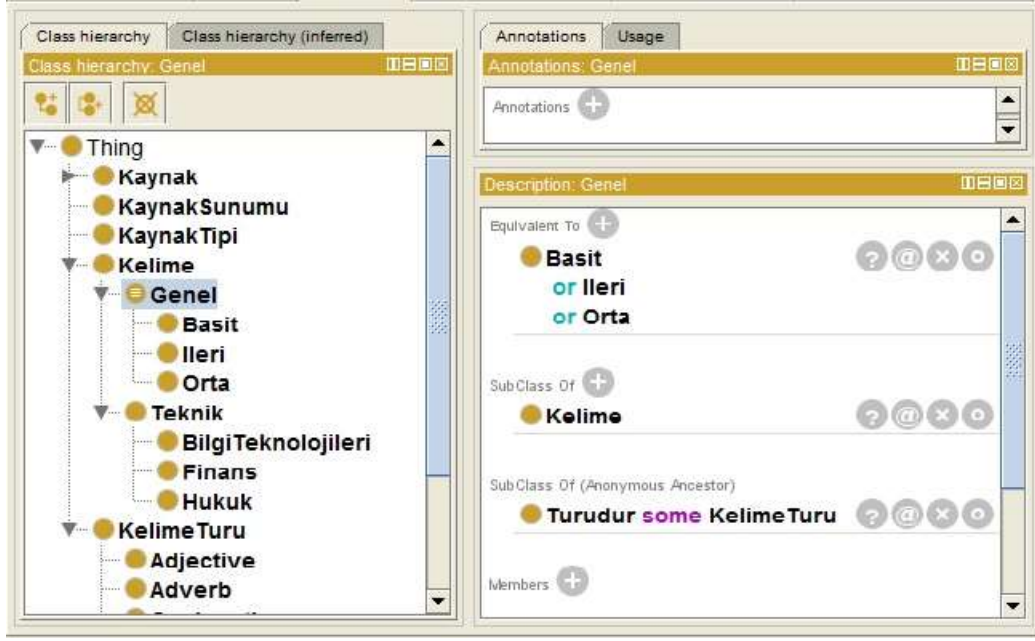
Çalışmada kullanılan ontolojiler Protege editörü ile hazırlanmıştır. Protege editörü ontoloji tasarlamak ve düzenlemek için görsel arayüz ve kolay kullanım sağlamaktadır. Ontoloji oluşturulurken ilk önce *Classes* sekmesinde sınıflar tanımlanmaktadır. Tanımlanan sınıflar *Class hierarchy* penceresinde görülmektedir. Sınıflara *Annotations* penceresinde sınıfla ilgili bilgi vermek amacıyla açıklama yazılabilir. *Description* penceresinde ise sınıfların eşit sınıfları, ayrı sınıfları, alt sınıfları, üyeleri, ilişkileri gibi düzenlemeler yapılır.

Şekil 3.2’de alan ontolojisinde bulunan sınıflar ve bu sınıflardan ders sınıfına ait düzenlemeler görülmektedir. Ders sınıfı Kaynak sınıfının alt sınıfı olarak *Description* penceresinde ki *SubClass of* kısmında belirtilmiştir. Kaynak tipi sınıfı ile *some* kısıtlaması kullanılarak ilişkilendirilmiştir. Üyeleri ise *Degerlendirme*, *Eslestirme*, *FlasKart* ve *Oyun*’dur. *Decription* penceresinde ki *Equivalent to* kısmında ilgili sınıfın eşit sınıfları seçilir.



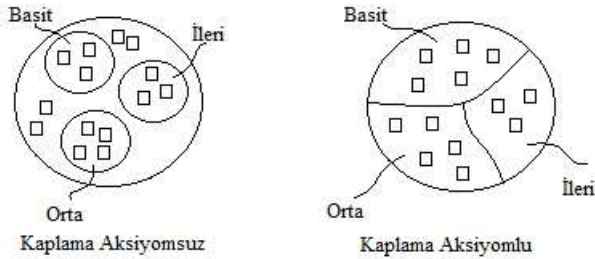
Şekil 3.2. Protege editörü

Eşit sınıf bir tane sınıftan olabileceği birden çok sınıftan da olabilir. Şekil 3.3'te görüldüğü gibi *Genel* sınıfı *Basit*, *Orta* ve *Ileri* sınıflarına denkleştirilmiştir. Sınıflar arasına *or* kısıtlaması kullanılmıştır. Bunun anlamı *Genel* sınıfı sadece *Basit*, *Orta* veya *Ileri* sınıflarındaki bireyleri kapsar. Yani *Genel* sınıfındaki bireyler ya *Basit* ya *Orta* ya da *Ileri* sınıfındaki bireylerden oluşur. *Genel* sınıfın da bulunan hiçbir birey *Basit*, *Orta* ve *Ileri* sınıfı dışındaki bir sınıfa ait olamaz. Bu tanımlama değer bölümleri (value partitions) olarak bilinir.



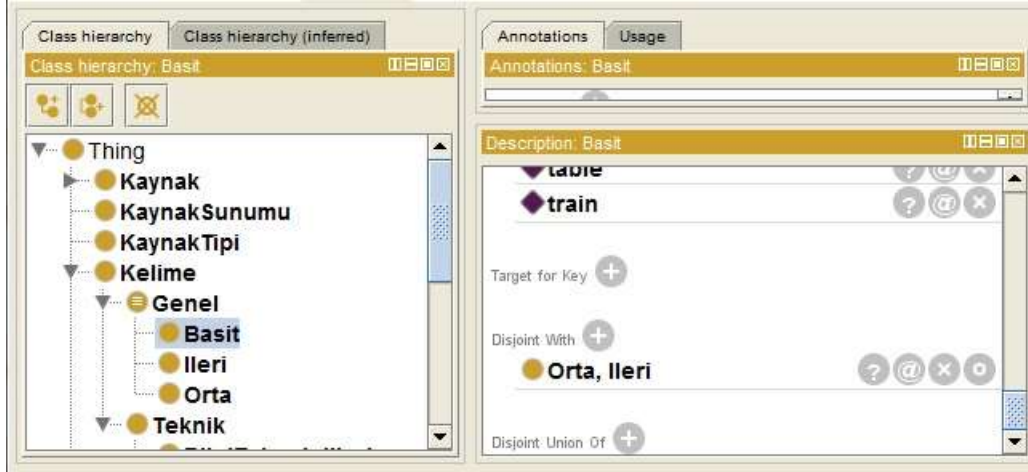
Şekil 3.3. Eşit sınıf

Değer bölümleri ontoloji ya da başka bir dilin bir parçası değildir. Değer bölümleri bir tasarım desenidir. Ontoloji tasarımında tasarım desenleri modelleme sorunlarının çözümlerinde nesne yönelimli programlama modellerine benzer. Değer bölümlerinin bir parçası olan kaplama aksiyomlarını (Covering Axioms) kullanır. Şekil 3.4’te kaplama aksiyomsuz ve kaplama aksiyomlu kullanımı gösterilmiştir.



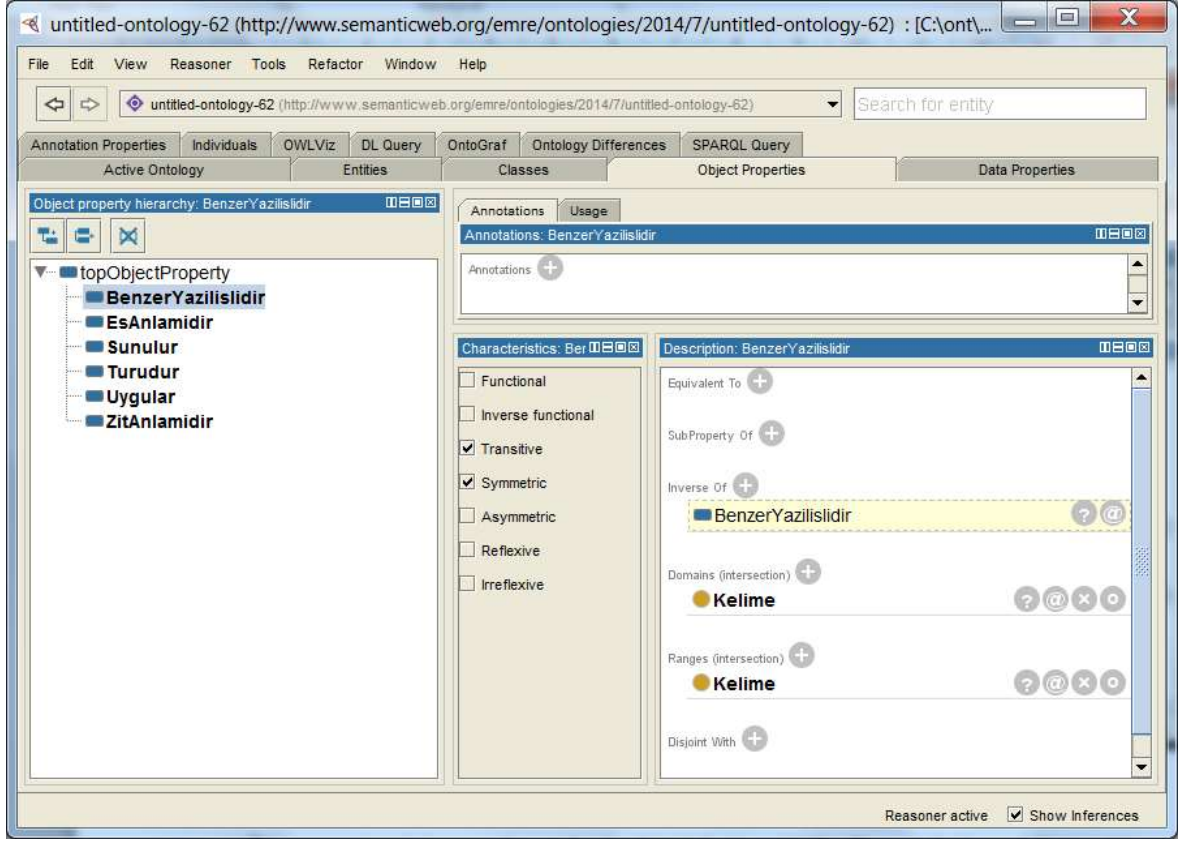
Şekil 3.4. Kaplama aksiyomsuz ve kaplama aksiyomlu görünümüne bir örnek

Şekil 3.5’te *Disjoint With* kısmında *Basit*, *Orta* ve *Ileri* sınıfları ayrık sınıflar olarak düzenlendiğinden dolayı bu sınıfları kesişim bireyleri yoktur. Yani bir birey *Basit* sınıfında ise *Orta* ya da *Ileri* sınıfında olamaz.



Şekil 3.5. Ayırık sınıf

Sınıfları ilişkilendirmek için iki tür özellik kullanılmıştır. Bu özellikler *Object Properties* (Nesne Özellikleri) ve *Data Properties* (Veri Özellikleri) sekmeleri altında tanımlanmaktadır. Şekil 3.6'da alan ontolojisinde bulunan nesne özellikleri görülmektedir.

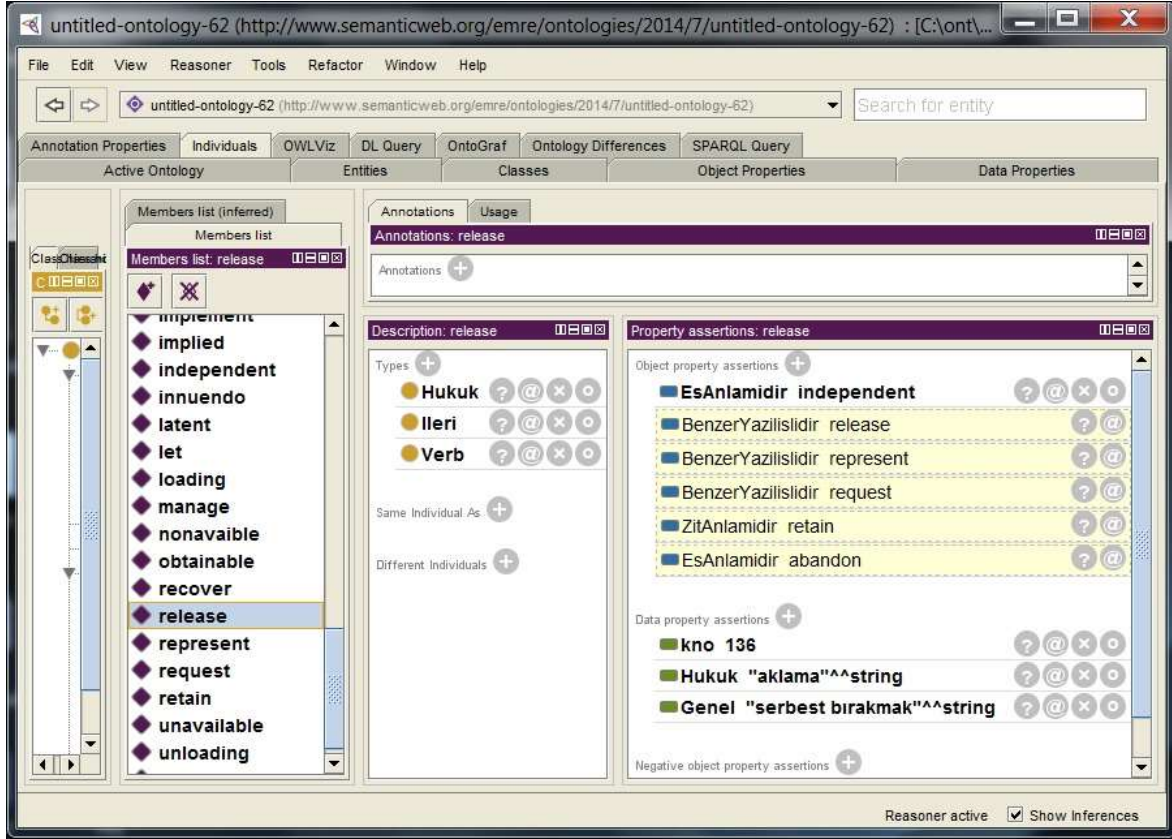


Şekil 3.6. Nesne özellikleri

Characteristics penceresinde bulunan seçenekler ile nesne özellikleri yapılandırılır. Örnekte kelime sınıfları arasında kurulan *BenzerYazilislidir* nesne özelliği için *transitive*(geçişli) ve *symmetric*(simetrik) özellikleri işaretlenmiştir (Bkz. Şekil 3.6). *Release* kelimesi *represent* kelimesi ile benzer yazılışlı olduğundan simetrik işareti seçili olduğundan dolayı *represent* kelimesi de *release* kelimesi ile benzer yazılışlı olur. *Represent* kelimesi ve *request* kelimesi arasında da benzer yazılış ilişkisi vardır. Geçişli işareti seçili olduğundan dolayı *release* ve *request* arasında da benzer yazılış çıkarımına ulaşılır. *BenzerYazilislidir* nesne özelliklerinde olduğu gibi *EsAnlamlidir* ve *ZitAnlamlidir* nesne özellikleri içinde bazı *characteristics* seçenekleri işaretlenmiştir. Hem eş anlamlı hem de zıt anlamlı kelimelerde simetrik özelliği vardır. Çünkü bir kelime başka bir kelimenin eş anlamı ise diğer kelime içinde eş anlamlılık geçerlidir. Zıt anlamlı kelimelerde de bu durum söz konusudur. Fakat eş anlamlı ve zıt anlamlı kelimeler için geçişli oldukları söylenemez.

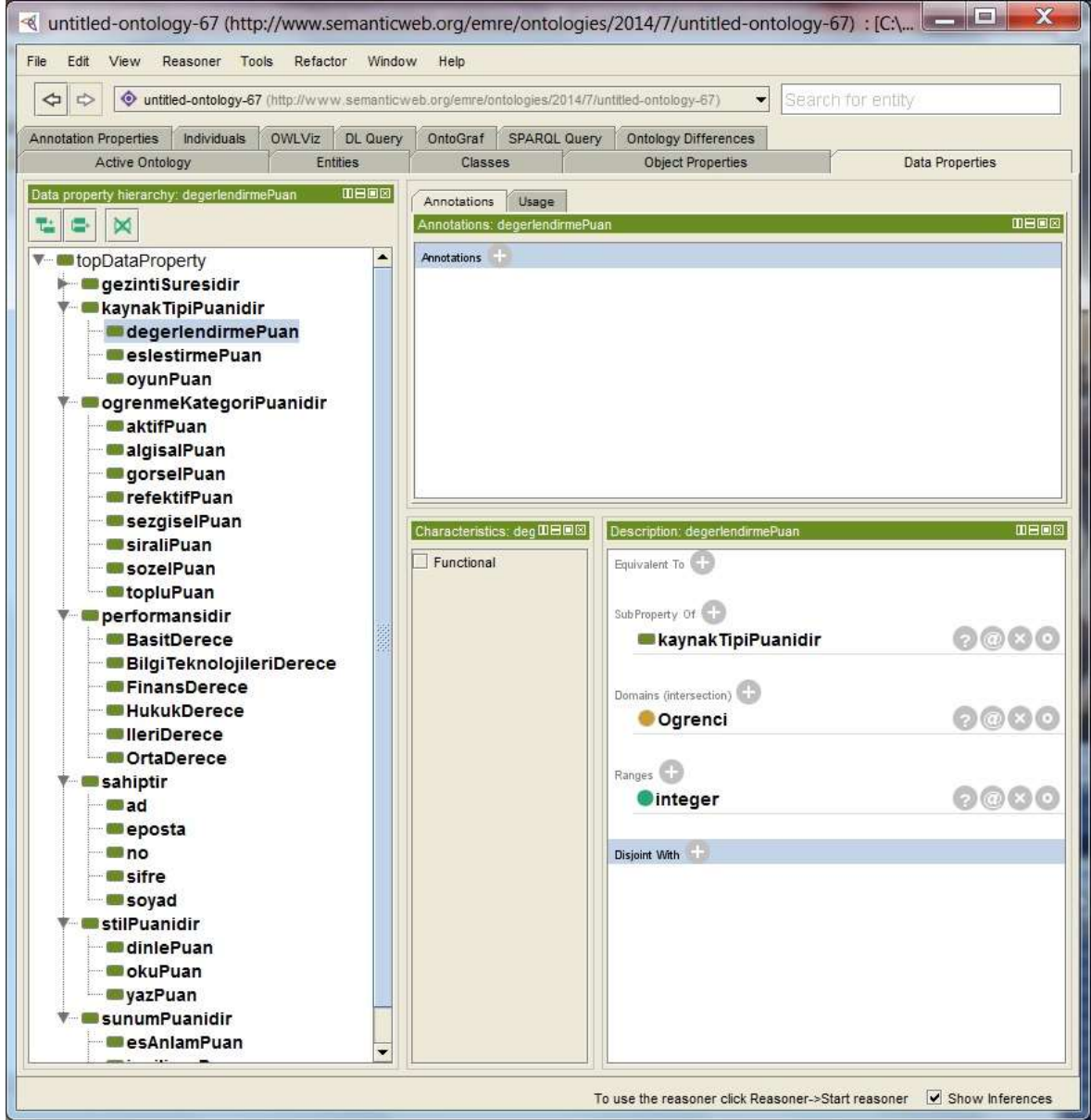
Şekil 3.7’de görüldüğü gibi *release* kelimesi sadece *independent* kelimesi ile *EsAnlamlidir* özelliği ilişkilendirilmiştir. Bunun dışında başka ilişkilerde görülmektedir. Bunun sebebi *release* kelimesi başka kelimeler tarafından *EsAnlamlidir*, *ZitAnlamlidir* ve

BenzerYazılıslidir özellikleri ile ilişkilendirilmiştir. Bu özellikler simetrik veya geçişli oldukları için bir çıkarım yapılmıştır. Örneğin *release* kelimesi *retain* kelimesi tarafından *ZitAnlamlidir* ilişkisi kuruldu ise *release* için de bu ilişkiyi kurmaya gerek yoktur. Çünkü *ZitAnlamlidir* özelliği simetrik olduğundan *retain* kelimesi *release* kelimesinin zıt anlamlısı ise *release* kelimesi de *retain* kelimesinin zıt anlamlısıdır çıkarımına ulaşılır. Bu çıkarımları görebilmek için ontolojinin Inferences modunda çalıştırılması gerekmektedir.



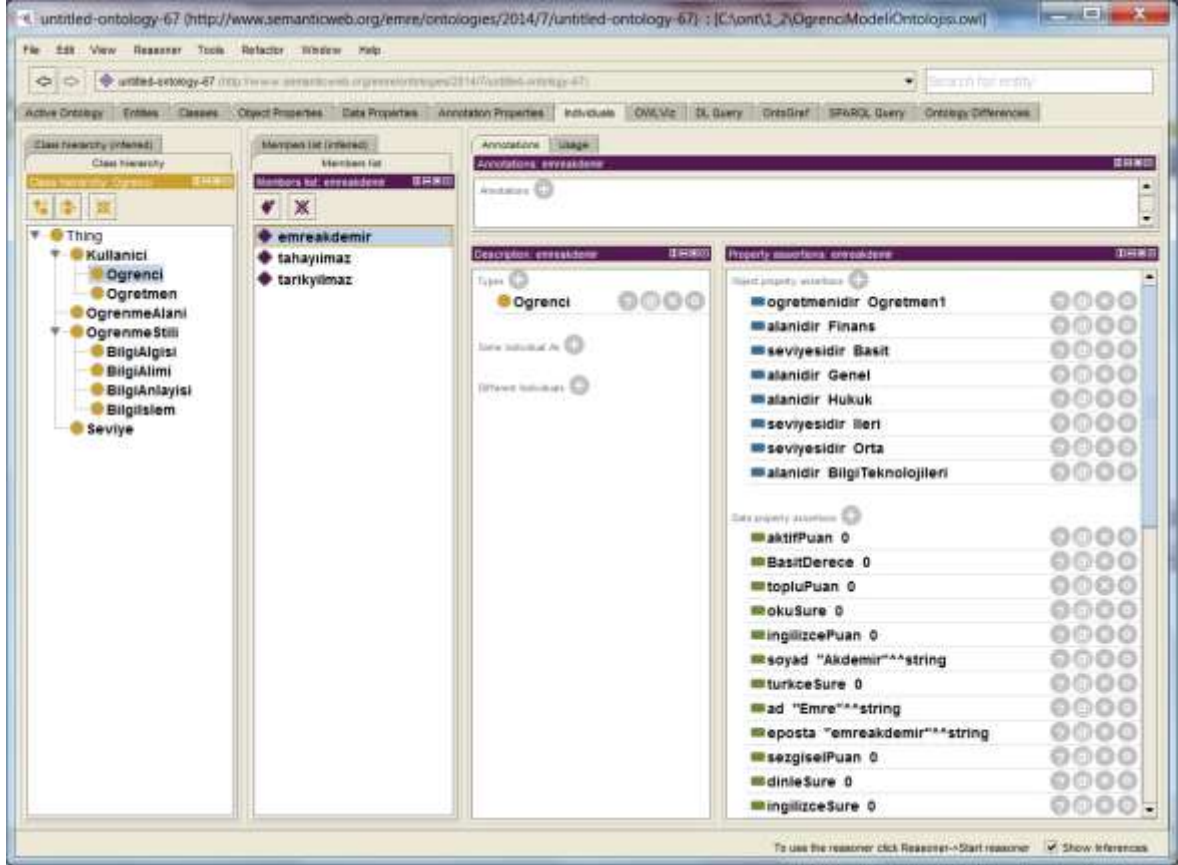
Şekil 3.7. Inferences modu

Şekil 3.8'de Öğrenci Modeli ontolojisine ait veri özellikleri görülmektedir. Veri özelliklerin tanımlanan *range* kısmı sayısal, metinsel vb. bir veri tipidir.



Şekil 3.8. Veri özellikleri

Özellikler tanımlandıktan sonra Individuals (bireyler) sekmesinde bu sınıflara ait varsa bireyler eklenir ve bu bireyler için oluşturduğumuz özelliklere değer verilir. Şekil 3.9'da öğrenci sınıfa ait bireyler görülmektedir. Bu bireylerin nesne özellikleri ve veri özellikleri değerleri ile birlikte görülmektedir.



Şekil 3.9. Bireylerin nesne ve veri özellikleri

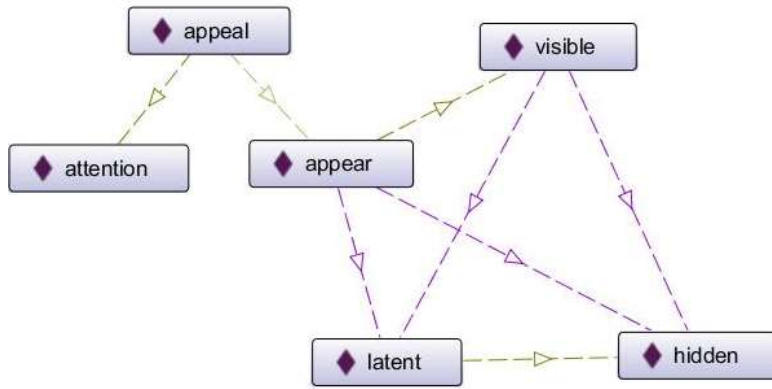
ZÖS modülleri için hazırlanan ontolojiler dört ayrı owl dosyalarında tutulmaktadır. Owl kendi yetenekleri ile birçok veri dağınık ortamlardan toplayabilir çıkarsama yapabilir ve farklı bilgileri birleştirilebilir [39]. Böylece owl dosyalarının ortak kullanacağı bilgiler program esnasında birleştirilerek aralarında iletişim kurulur. Uygulamada kullanılan ZÖS modüllerine ait ontolojiler ve görevleri aşağıda açıklanmıştır.

3.2.1. Alan ontolojisi

Öğrenme sürecinin temel amaçlardan biri bilgi içeriğidir. Bu ontoloji gerekli tüm öğrenme materyalleri, öğretim ve testler sunmaktadır. Öğrenme içeriğinin nasıl olacağı belirlenmektedir. Bunun için kavramlar sınıflandırılmakta ve aralarındaki ilişkiler oluşturulmaktadır. Bu ontolojide bilgi saklandığı gibi kontrolde edilir. Bu bilgiler, kural tabanlı, semantik ağlar ve çerçeveler gibi yöntemler ile sunulabilir. Kullanılan teknikler problemin nasıl çözüleceğini belirler. Bu yöntemler ile iki temel işlev gerçekleştirilir. Birincisi, bir kaynak gibi davranış gösterme, ikincisi ise bilişsel düzeyi saptamak için öğrenci performanslarının değerlendirilmesidir. Performans değerlendirme sonucu ile

öğretim içeriği belirlenir. Bu yüzden hem öğretim stratejisi ontolojisi hem de öğrenci modeli ontolojisi arasında sürekli bilgi alışverişi bulunmaktadır. Bu yöntemleri bilgisayar ortamında hazırlayıp aralarında bağlantı kurup organize edilmesi oldukça karmaşık ve zaman alan bir süreçtir [4, 5, 7].

Alan ontolojisi bir uzmanı taklit eder. Bilgi tabanı ne kadar çok gelişmiş ise çözüm için o kadar yeterli olacaktır. Aksi takdirde yeterli bilgi içermeyen bir bilgi tabanı öğrenci için eksik bir öğretim sağlar. Öğrencinin zorlandığı alanları belirlemesinde ve bunun için çözümler üretmesinde bir uzman gibi davranış sergileyebilmelidir.

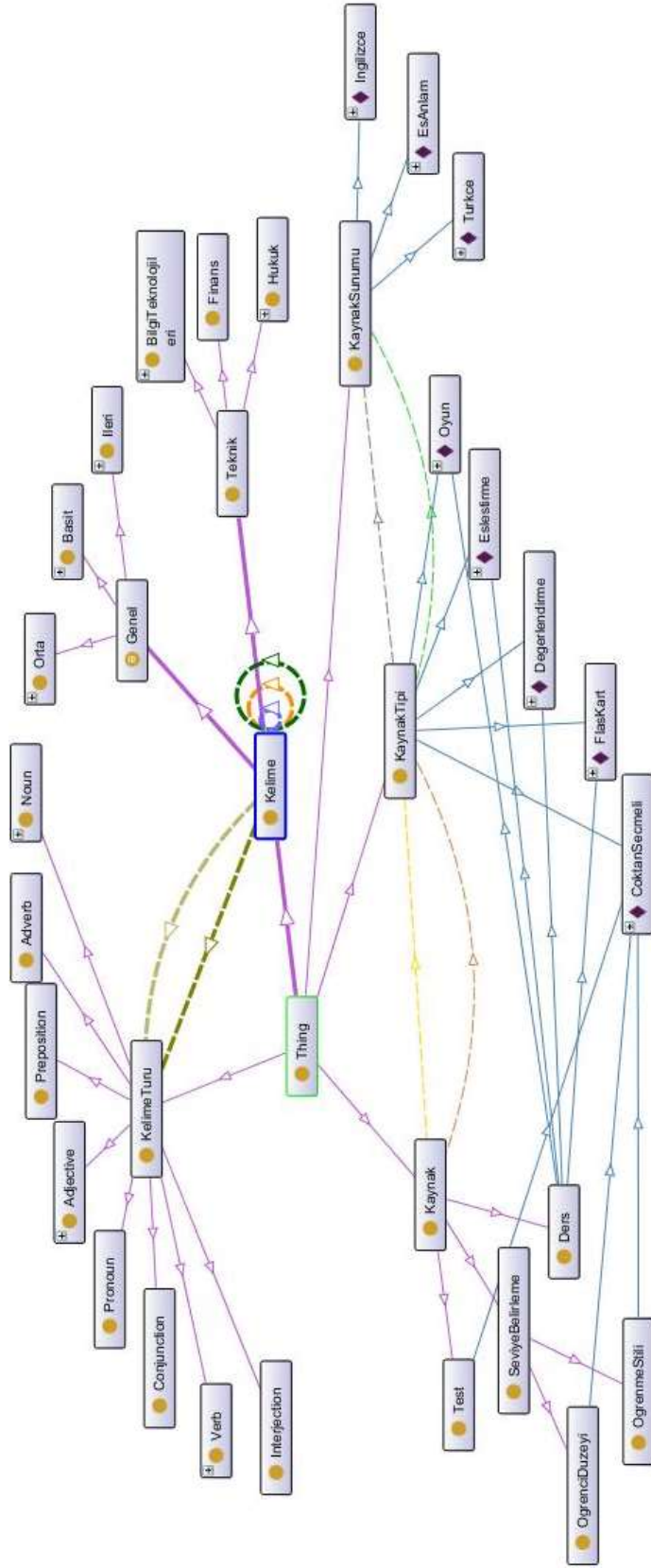


Şekil 3.10. Kelimeler arası ilişkiler

Bu çalışmada alan ontolojisi oluşturulurken kelimeler önce sınıflara ayrılmıştır. Bir kelimenin anlamı bu sınıflara göre değişmektedir. Öğrencinin çalışma alanına ve kelime türüne göre ilgili anlam sunulmaktadır. Kelimeler İngilizce-Türkçe, Türkçe-İngilizce ve İngilizce eş anlamlı olarak sunumu öğrenciye göre ya da isteğe bağlı olarak değişmektedir. Ontoloji ile kelimelerin arasında ilişki kurulurken sınıfları, türleri, eş anlamlı, zıt anlamlı, benzer yazılışları dikkate alınmıştır. Bu ilişkiler Şekil 3.10'da görülmektedir. Bu ontolojide kullanılan kaynaklar seviye belirleme, ders ve test materyalleridir. Öncelikle Solomon ve Felder'in (1996) hazırladığı öğrenme stilini belirlemek için Index of Learning Styles Questionnaire (ILS) kullanılmaktadır. Bu testin temelleri Felder ve Silverman'ın (1988) çalışmalarına dayanmaktadır [46]. ILS testi dört kategoride toplam 44 soru olarak tasarlanmıştır. 1997 yılında çevrim içi versiyonu eklenmiştir [47]. ILS testi ile öğrencinin görsel/sözel, sıralı/genel, aktif/reflektif, algısal/sezgisel öğrenme stil kategorisi belirlenmektedir. Öğrenci davranışlarını ölçmek için ILS kullanılarak yapılan çalışmalardaki

sonular oldukça başarılıdır [47, 48]. ILS kullanımı ZÖS’nde tercih edilme ve başarılı sonuçlar elde edilmektedir [49]. Böylece başlangıta öğrencinin öğrenme stili belirlenerek ders içerikleri bu sonuca göre sunulmaktadır. Daha sonra öğrencinin öğrenme alanındaki öğrenme düzeyi belirlenir. Eğer yeterli ise bir üst seviyeye geçebilir. Ders içeriği flaş kart, oyun, eşleştirme, değerlendirme kaynak rolleri ile sunulmaktadır. Son olarak öğrencinin öğrenmesini ölçmek için test yapılmaktadır ve performansı değerlendirilmektedir. Test sorularında seçenekler belirlenirken eş anlamlı kelime seçeneklerde olmamaktadır. Birden fazla cevap olmamaktadır. Seçeneklerde zıt anlamlısı ve benzer yazılışı anlamı olması tercih edilmiştir. Böylece bir öğretmenin hazırladığı teste benzetilmiştir. Şekil 3.11’de alan ontolojisi ontograf ile gösterilmiştir.

Alan ontolojisi, öğrenci modeli ve öğretim stratejisi ontolojileri için bilgi sağlar. Bu bilgiler doğrultusunda öğrenciye göre arayüz ontolojisi belirlenir.



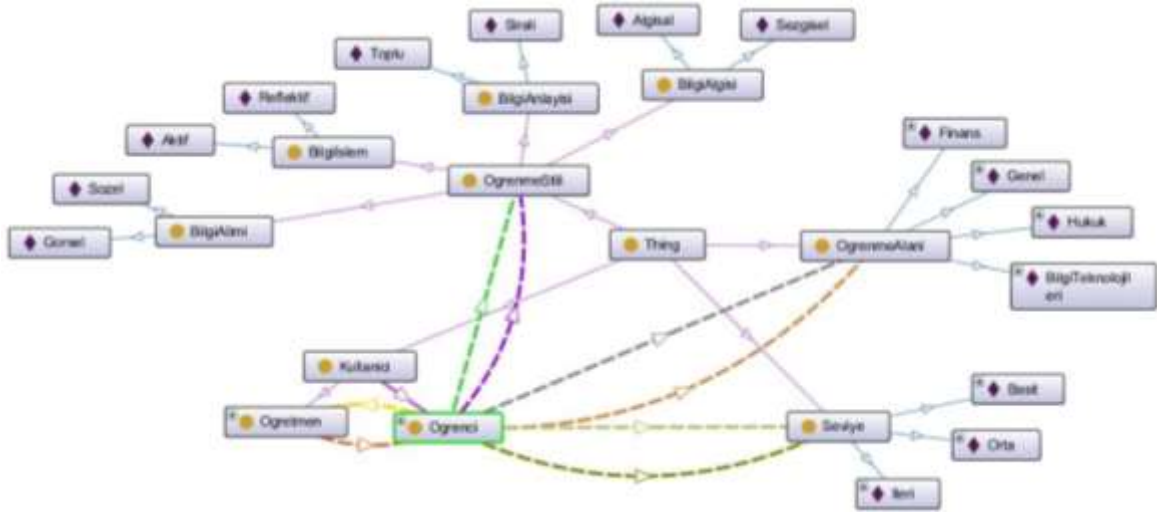
Şekil 3.11. Alan ontolojisi

3.2.2. Öğrenci modeli ontolojisi

Zeki Öğretim Sistemlerinin en önemli özelliklerinden biri öğrenciye göre uyarlanabilir olmasıdır. Öğrenci ile bire bir öğrenme ortamı sağlanmaktadır. Böylece öğrenci eksiklerini görebilir. Bu süreçte öğrenmede ki her bir adım kaydedilir. Öğrencinin tutumu, sorulara verdiği cevaplar, işlem adımları izlenerek performansı değerlendirilir. Alan ontolojisindeki bilgiler ile verilen kararlar doğrultusunda öğrenme yöntemi öğrencinin özelliklerine bağlı olarak belirlenerek öğretim stratejileri ontolojisine gönderilir. Öğrenme süreci tekrarlanarak geri dönüş sağlanır ve öğrenciye bireysel tavsiyeler oluşturularak rehberlik edilir [6, 7].

Daha öncede bahsedildiği gibi öğrencinin davranışlarını ölçmek ve bilişsel bilgi düzeyini saptamak için öğrenci profili oluşturulması gerekmektedir. Öğrenci sistemi kullanırken sayfasına erişebilmek için giriş yapmalıdır. Sayfasında kendisine ait tüm bilgileri görebilmektedir. Sistemi kullandıkça bu bilgiler değişmektedir ve öğrenci bu değişimleri de takip edebilir. Sistemde öğrenci bilgileri uzun dönemli ve kısa dönemli olmak üzere iki şekilde tutulmaktadır. Öğrencinin öğrenme stilleri, seviyesi, öğrenme metodu gibi bilgiler uzun dönemli olarak tutulmaktadır ve bu bilgiler anında değişmez. Öğrencinin öğrendiği kelimeler, bu kelimelere verdiği doğru, yanlış cevaplar, öğrencinin bir alandaki puanı gibi bilgileri ise kısa dönemlidir. Anlık değişmektedir ve sonuçları ekrana anında yansımaktadır.

Öğretmen kullanıcısı öğrencilerinin performanslarını takip edebilmektedir. Öğrenciler ile ilgili grafiksel istatistiksel sonuçları görebilmektedir. Ayrıca kelime ekleme ve güncelleme yetkisi sayesinde daha önce bahsedildiği gibi alan ontolojisinin büyümesinde rol oynamaktadır. Şekil 3.12’de öğrenci modeli ontolojisi ontograf ile gösterilmiştir.



Şekil 3.12. Öğrenci modeli ontolojisi

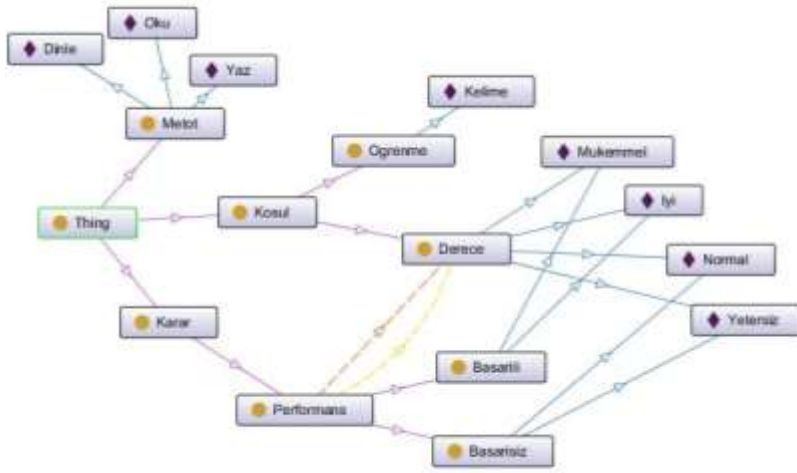
3.2.3. Öğretim strateji ontolojisi

Öğrenme süreçlerinin bir modelidir. Bu ontolojinin temel amacı belirli bir pedagojik stratejisi belirlemektir. Öğrenci modeli ontolojisinden sağlanan bilgiler doğrultusunda her bir öğrencinin hangi metot ve taktiklerini kullanması gerektiğinin kararı verilir. Bu bilgiler doğrultusunda materyalleri öğrenciye göre sunar. Bilgilerin ne zaman ve ne kadar sürede sunulacağı, yeni bir konuya ne zaman geçileceği, hangi konuların sunulacağı ve hata mesajlarının uyarlanabilir sunumu bu ontoloji tarafından kontrol edilir ve öğrenciler yönlendirilir. ZÖS kendi öğretim stratejisi ontolojisine göre öğretimi gerçekleştirir. Öğretim sürecini kontrol eder, öğrenciye geri dönüş sağlar ve gerekli seçimleri yapar. Kısaca bir eğitimci gibi müdahalede bulunur [4, 5].

Bu ontolojide bir problem çözümünde kullanılan stratejiler öğrencilerin ilerleyebilmeleri için adım adım sunulmaktadır. Böylece öğrencinin hangi adımlarda zorlandığı kolaylıkla tespit edilir ve gerekli seçimler yapılır. Adımlar her adımda zorlaşarak ilerlemesi öğrencinin hangi seviyede olduğunu kolaylıkla öğrenir [20].

Bu modülde kullanılan mekanizma öğrencilerin öğretime uyumluluğu her öğretim stilleri için verilen puanlama ile belirlenir. Belirtilen materyaller kategoriye ayrılarak hangi öğrenme stillerinin daha iyi sonuç elde ettiği tespit edilir [21]. Bu çalışmada öğretim stratejileri ontolojisinde görsel, işitsel ve kinestetik metotlar bulunmaktadır. Bu metotların belirlenmesinde kelime öğrenimi esas alınmıştır. Bu metotlar ile öğrenciler kelime ve

anlamalarını hızlı bir şekilde okuyarak, daha kalıcı olabilmesi için yazarak veya telaffuzlarını geliştirmek için duyararak öğrenme sağlayabilirler. Öğrencinin davranışları ve performans sonuçları değerlendirilerek en uygun öğrenme metodu belirlenmektedir. Alan ontolojisindeki öğrencinin kullandığı kaynak bilgileri doğrultusunda derecelerinin belirlenip, bulunduğu alanı tamamlanması kararının alınabilmesi için öğrenci modeli ontolojisi ve alan ontolojisinden yararlanmaktadır. Arayüz ontolojisi ile iletişindedir. Şekil 3.13'te öğretim stratejisi ontolojisi ontograf ile gösterilmiştir.



Şekil 3.13. Öğretim stratejisi ontolojisi

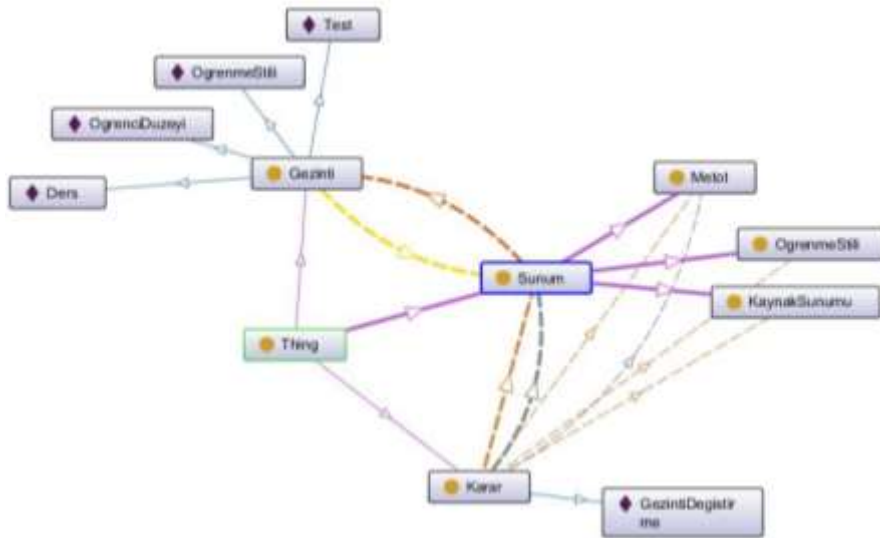
3.2.4. Arayüz ontolojisi

Öğrenci ile ZÖS arasındaki iletişim ve etkileşimi sağlayan, diyalogları ve ekran planlarını kontrol eden ontolojidir. Farklı bileşenlerdeki ontolojiler ile iletişimdeki son aşamadır. Öğretim stratejisi ontolojisi bilgilerine göre öğrencilere önerilen kaynak gezinti sırası oluşturulur ve böylece öğrenen bir arayüz görünümü sağlanır.

Daha önce de bahsedildiği gibi arayüz ontolojisi bütün bileşenleri ve kullanıcılar arasındaki iletişimi sağlayan temel bileşendir. Materyallerin etkin bir biçimde sunabilmek, öğrenci motivasyonunu sağlayabilmek ve ilgi çekici bir içeriğe sahip olabilmek için arayüz tasarımı önemlidir. Bu uygulamada sunulan bilgiler ses, resim ve metinsel araçlarla desteklenmektedir. Ayrıca bütün duyu organlarına hitap edecek şekilde işitsel, görsel ve dokunmatik olarak tasarlanmıştır. Uygulamada ki kelimeler öğrenme stiline göre hem metinsel olarak, hem resimlerle ifade edilmiştir hem de ses ile telefuzları sunulmuştur. Yine

öğrenme stiline göre metinsel kelimelerin doğru cevapları klavye ile öğrencinin yazmasına olanak sağlanmıştır.

Bu çalışmada arayüz ontolojisi öğrenci bilgilerine göre gezinti sırası ve öğrenme stili belirlenerek içerik sunulmaktadır. Yine öğrenciye göre sunumlar İngilizce, Türkçe ve eş anlamlı olarak değişmektedir. Bir öğrenme alanında yeni başlayan bir öğrenci için önce seviye belirleme testleri yapılmaktadır. Daha sonra belirlenen metotlar ile ders aktiviteleri gezinti sırasına göre ya da öğrencinin isteğine göre sunulmaktadır. Flaş kart aktivitesini yeteri kadar çalıştığı kararı verilirse diğer aktivitelere yani oyun, eşleştirme, değerlendirmeye yönlendirilmektedir. Eğer bu aktivitelerde öğrenci başarısız olursa tekrar flaş kart çalışmasına yönlendirilmektedir. Başarılı olduğu kararı verilirse ders tamamlanmadan test yapmaya geçebilir. Test uygulanırken çok fazla yanlış cevap verilirse test tamamlanmadan tekrar ders çalışmaya yönlendirilmektedir. Böylece arayüz ontolojisi bir öğretmen gibi çalışmaların yeterli olduğu ya da yetersiz olduğu kararını verip yönlendirme sağlamaktadır. Şekil 3.14'te arayüz ontolojisi ontograf ile gösterilmiştir.



Şekil 3.14. Arayüz ontolojisi

Öğrenme stillerinin arayüze etkileri büyüktür. Öğrenme stili aktif/reflektif grubundan aktif puanı daha yüksek olanlar kararları kendisi vermek isterken, reflektif puanı yüksek olanlar ise kendilerinin yönlendirilmesinden memnun kalırlar. Görsel/sözel grubundan görsel puan yüksek olanlar görerek ve işiterek öğrenmeyi tercih ederken sözel grubunda olanlar yazılı ifadelerle öğrenmeyi tercih ederler. Genel/Sıralı grubunda genel puan yüksek olanlar

tasarımdaki bütün bir yapıyı görmek isterler, sıralı puanı yüksek olanlar ise baştan sona doğru adım adım yapıya sahip gezinim sistemini tercih ederler. Sezgisel/algısal grubunda sezgisel puanı yüksek olanlar olaylara sezgileri ile yaklaşmayı tercih ederler, algısal puanı yüksek olanlar ise düşünerek, anlayarak yaklaşmayı tercih ederler [50].

3.3. Ontolojilerin Birleştirilmesi

Oluşturulan ontolojilerin aralarında iletişim kurabilmeleri için birleştirilmektedir. Birleştirme işleminde ilgili sınıflar arasında eşit sınıfı (Equivalent) belirtilmektedir. Şekil 3.15'te farklı ontolojideki bazı sınıfların birleştirilmesini sağlayan java kodları gösterilmiştir. Şekil 3.16' da ise birleştirilmiş dört ontolojinin OWLViz ile gösterimi verilmiştir.

```
public static void siniflariBirlestir(){
    OntClass sinif1, sinif2;
    sinif1 = model.getOntClass(ns + "Kaynak");
    sinif2 = model.getOntClass(ns_sunum + "KaynakSunumu");
    sinif1.addEquivalentClass(sinif2);

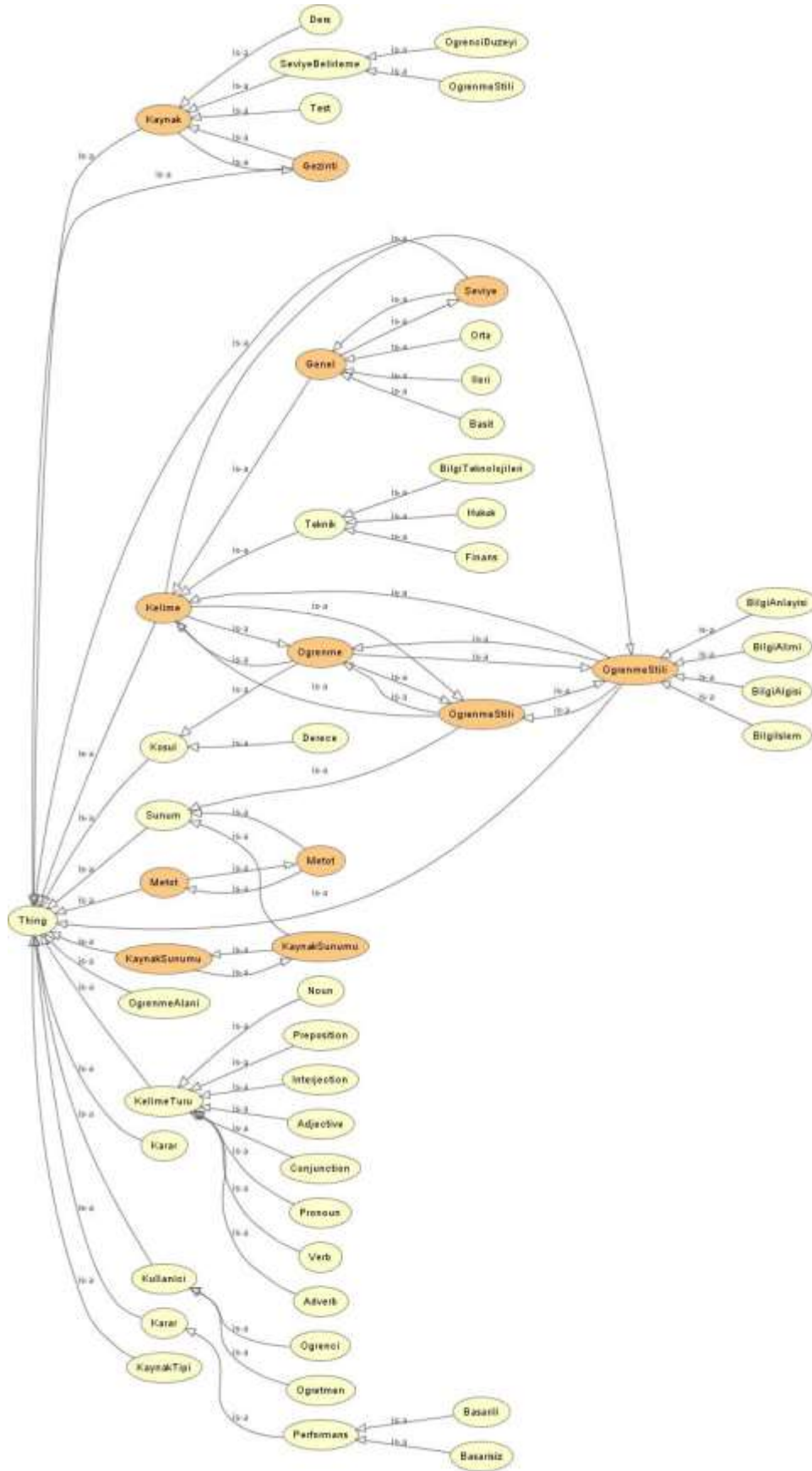
    sinif1 = model.getOntClass(ns + "Kelime");
    sinif2 = model.getOntClass(ns_ogrenci + "OgrenmeAlani");
    sinif1.addEquivalentClass(sinif2);

    sinif1 = model.getOntClass(ns + "Genel");
    sinif2 = model.getOntClass(ns_ogrenci + "Seviye");
    sinif1.addEquivalentClass(sinif2);

    sinif1 = model.getOntClass(ns_sunum + "Metot");
    sinif2 = model.getOntClass(ns_ogretim + "Metot");
    sinif1.addEquivalentClass(sinif2);

    sinif1 = model.getOntClass(ns_sunum + "OgrenmeStili");
    sinif2 = model.getOntClass(ns_ogrenci + "OgrenmeStili");
    sinif1.addEquivalentClass(sinif2);
}
```

Şekil 3.15. Eşitlik sınıfı tanımlayan java kodları



Şekil 3.16. Birleştirilmiş ontolojilerin OWLViz ile gösterimi

Ontoloji ile oluşturulan OWL dosyalarının örnek bir kod kısmı Şekil 3.17’de verilmiştir.

```

<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.w3.org/2002/07/owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <Ontology rdf:about="http://www.semanticweb.org/emre/ontologies/2014/7/untitled-
ontology-62"/>
  <Class rdf:about="&untitled-ontology-62;Genel">
    <equivalentClass>
      <Class>
        <unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="&untitled-ontology-62;Basit"/>
          <rdf:Description rdf:about="&untitled-ontology-62;Ileri"/>
          <rdf:Description rdf:about="&untitled-ontology-62;Orta"/>
        </unionOf>
      </Class>
    </equivalentClass>
    <rdfs:subClassOf rdf:resource="&untitled-ontology-62;Kelime"/>
  </Class>
  <Class rdf:about="&untitled-ontology-62;Kaynak">
    <rdfs:subClassOf>
      <Restriction>
        <onProperty rdf:resource="&untitled-ontology-62;Uygular"/>
        <someValuesFrom rdf:resource="&untitled-ontology-62;KaynakTipi"/>
      </Restriction>
    </rdfs:subClassOf>
  </Class>

```

Şekil 3.17. Owl kodları

```

<Class rdf:about="&untitled-ontology-62;Kelime">
  <rdfs:subClassOf>
    <Restriction>
      <onProperty rdf:resource="&untitled-ontology-62;Turudur"/>
      <someValuesFrom rdf:resource="&untitled-ontology-62;KelimeTuru"/>
    </Restriction>
  </rdfs:subClassOf>
</Class>
<ObjectProperty rdf:about="&untitled-ontology-62;BenzerYazilislidir">
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:domain rdf:resource="&untitled-ontology-62;Kelime"/>
  <rdfs:range rdf:resource="&untitled-ontology-62;Kelime"/>
</ObjectProperty>
<DatatypeProperty rdf:about="&untitled-ontology-62;kno">
  <rdfs:domain rdf:resource="&untitled-ontology-62;Kelime"/>
  <rdfs:subPropertyOf rdf:resource="&untitled-ontology-62;sahiptir"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</DatatypeProperty>
<NamedIndividual rdf:about="&untitled-ontology-62;asset">
  <rdf:type rdf:resource="&untitled-ontology-62;Hukuk"/>
  <rdf:type rdf:resource="&untitled-ontology-62;Ileri"/>
  <rdf:type rdf:resource="&untitled-ontology-62;Noun"/>
  <unt-ontology-62:kno rdf:datatype="&xsd;integer">96</unt-ontology-62:kno>
  <unt-ontology-62:Genel rdf:datatype="&xsd;string">değer</ontology-62:Genel>
  <unt-ontology-62:Hukuk rdf:datatype="&xsd;string">varlık</ontology-62:Hukuk>
  <unt-ontology-62:EsAnlamidir rdf:resource="&unt-ontology-62;advantage"/>
  <unt-ontology-62:BenzerYazilislidir rdf:resource="&unt-ontology-62;assign"/>
  <untitled-ontology-62:ZitAnlamidir rdf:resource="&unt-ontology-62;damage"/>
</NamedIndividual>

```

Şekil 3.17 (Devam)

3.4. Jena Framework

Uygulama Java programlama dili ve MySQL veritabanı ile geliştirilmiştir. Anlamsal web uygulaması geliştirmek için java ile uyumlu çalışabilen Apache Jena çerçevesi tercih edilmiştir. Bu uygulamada Jena 2.5.5 sürümü kullanılmıştır. Jena çerçevesinin sunduğu zengin kütüphane ile ontoloji kolaylıkla yönetilmektedir. Hazırlanan ontoloji dosyalarında programın çalışması esnasında yapılan değişikliklerin devamını sağlayabilmesi için ontoloji modeli veritabanına kaydedilmektedir. Jena kütüphanesinin sunduğu persistence sınıfları ile MySQL de bulunan belirli formattaki tablolara aktarılmaktadır. Bu tablolar durum tabloları ve sistem tabloları olmak üzere iki kısımdan oluşmaktadır. Bu tabloların adları ve fonksiyonları Çizelge 3.1’de ve Çizelge 3.2’de gösterilmiştir.

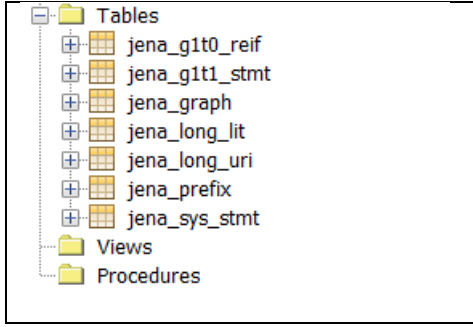
Çizelge 3.1. Durum tabloları [51]

Ad	Fonksiyon
Jena_GiTj_Stmt	Ontoloji verilerini saklar
Jena_GiTj_Reif	Ontoloji verileri davranışı

Çizelge 3.2. Sistem tabloları [51]

Ad	Fonksiyon
Jena_Sys_Stmt	Sistem meta verilerini saklar
Jena_Long_Lit	Durum tablosundaki uzun metin sabitlerini saklar
Jena_Long_URI	Durum tablosundaki URI’leri saklar
Jena_Prefix	URI öneklerini saklar
Jena_Graph	Kullanıcı adı ve benzersiz kimlikleri saklar

Bu tablolarda kayıtlar uri olarak tutulmaktadır. Uri’ ler arası bağlantılar ile verilere erişilmektedir. Şekil 3.18’de MySQL veri tabanında tanımlanan tablolar görülmektedir. Ontoloji ne kadar karmaşık olsa da bu tablolar standart olarak tasarlanmıştır.



Şekil 3.18. Jena tabloları

Eğer ilişkisel veri tabanı kullanılsaydı daha çok tablo ve karmaşık bir ilişkiye ihtiyaç olacaktır. Şekil 3.19'da meta verilerini saklayan *jena_g1t1_stmt* tablosundaki veriler görülmektedir.

#	Subj	Prop	Obj
1	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Perfor...	Urc: http://www.w3.org/2000/01/rdf-schema#SubClassOf	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
2	Bv: 320e2b1c:14b842bd6bf-7ff2:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.w3.org/2002/07/owl#Restriction
3	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Perfor...	Urc: http://www.w3.org/2000/01/rdf-schema#SubClassOf	Bv: 320e2b1c:14b842bd6bf-7ff2:
4	Bv: 320e2b1c:14b842bd6bf-7ff2:	Urc: http://www.w3.org/2002/07/owl#onProperty	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
5	Bv: 320e2b1c:14b842bd6bf-7ff2:	Urc: http://www.w3.org/2002/07/owl#someValuesFrom	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
6	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Denk:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.w3.org/2002/07/owl#NamedIndividual
7	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Denk:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
8	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.w3.org/2002/07/owl#NamedIndividual
9	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
10	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
11	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.semanticweb.org/emre/ontologies/2014/...	Urc: 40: http://www.w3.org/2001/XMLSchema#integer: 70:
12	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.w3.org/2002/07/owl#NamedIndividual
13	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...
14	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.semanticweb.org/emre/ontologies/2014/...	Urc: 40: http://www.w3.org/2001/XMLSchema#integer: 5:
15	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.semanticweb.org/emre/ontologies/2014/...	Urc: 40: http://www.w3.org/2001/XMLSchema#integer: 50:
16	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.semanticweb.org/emre/ontologies/2014/...	Urc: 40: http://www.w3.org/2001/XMLSchema#integer: 50:
17	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.w3.org/2002/07/owl#NamedIndividual
18	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/untitled-ontology-65#Iy1:	Urc: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Urc: http://www.semanticweb.org/emre/ontologies/2014/7/...

Şekil 3.19. jena_g1t1_stmt tablosu veri görünümü

Veritabanındaki verilere erişmek, güncelleme, eklemek vb. işlemler için önce *OntModel* sınıfı tanımlanır. Şekil 3.20'de *OntModel* sınıfından bir nesne tanımlayan java kodları gösterilmiştir.

```
OntModel newmodel = ModelFactory.createOntologyModel(getModelSpec(maker),
base);
```

Şekil 3.20. OntModel sınıfından bir nesne tanımlayan java kodları

OntModel sınıfı ile bir örnek tanımlanırken *OntModelSpec* sınıfı parametre olarak kullanılır. Bu sınıfın kurucu metod parametresi *OntModelSpec* türünde enumlara sahiptir. Şekil 3.21'de bu işlemleri gerçekleştiren java kodları gösterilmiştir

OWL_MEM değeri kullanılırsa inferences modu olmadan ontolojiyi veri tabanına aktarır, *OWL_MEM_MICRO_RULE_INF* değeri kullanılırsa inferences modu ile ontoloji veri tabanına aktarılır. Bu programda inferences (çıkarım) gerektiği için *OWL_MEM_MICRO_RULE_INF* değeri kullanılmıştır.

```
OntModelSpec spec =
new OntModelSpec(OntModelSpec.OWL_MEM_MICRO_RULE_INF);
```

Şekil 3.21. OntModelSpec sınıfından bir nesne tanımlayan java kodları

Jena çerçevesi ile jena tabloları oluşturmak için Şekil 3.22’de ki java kodları kullanılmaktadır. Bu kodlarda *connectDB* metodu jena veritabanı ile kullanıcı adı ve şifre parametresi göndererek bağlantı oluşturur. *createDBModelFromFile* metodu ilgili adresteki OWL dosyalarını okur. *getModelFromDB* metodu ise okunan dosyaları jena tablolarına uyarlayarak verileri aktarır.

```
public static void createDB(){
    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    IDBConnection con = connectDB(DB_URL,DB_USER, DB_PASSWD, DB);
    System.out.println(con);
    if(dosya){
        OntolojiVeri.createDBModelFromFile(con, "expert4", dosyaAlanOntoloji);
        OntolojiVeri.createDBModelFromFile(con, "expert4", dosyaOgretimOntoloji);
        OntolojiVeri.createDBModelFromFile(con, "expert4", dosyaOgrenciOntoloji);
        OntolojiVeri.createDBModelFromFile(con, "expert4", dosyaArayuzOntoloji);
    }
    model = OntolojiVeri.getModelFromDB(con, "expert4");
}
```

Şekil 3.22. Jena tabloları oluşturan java kodları

OntModel oluşturulduktan sonra ontoloji veri tabanında sürekli değişmektedir. Yeni değerler eklenebilir, silinebilir veya güncellenebilir. Ya da sadece veri çekilebilir. Bunlar için jena kütüphanesinin sunduğu metotlardan yararlanacağız. En çok kullanılan veri özelliği güncelleyen metot Şekil 3.23’te gösterilmiştir.

```

static void dataTypePropertyUpdate(String birey, String ozellik, String deger, String
ns){
    String Integer_uri = "http://www.w3.org/2001/XMLSchema#integer";
    Individual i = model.getIndividual(ns+ birey);
    DatatypeProperty dp = model.getDatatypeProperty(ns + ozellik);
    DatatypeProperty dpInt = model.getDatatypeProperty(ns + "no");
    RDFNode n = (RDFNode) i.getPropertyValue(dpInt);
    Resource r = ResourceFactory.createResource(deger + n.toString().substring(1));
    RDFNode node = (RDFNode) r;
    i.setPropertyValue(dp, node);
}

```

Şekil 3.23. Java kodları veri tiplerini güncelleyen metot.

Bu metot parametre olarak gelen namespace'teki, parametre olarak gelen bireyin parametre olarak gelen veri özelliğini yine parametre olarak gelen değer ile güncellemektedir.

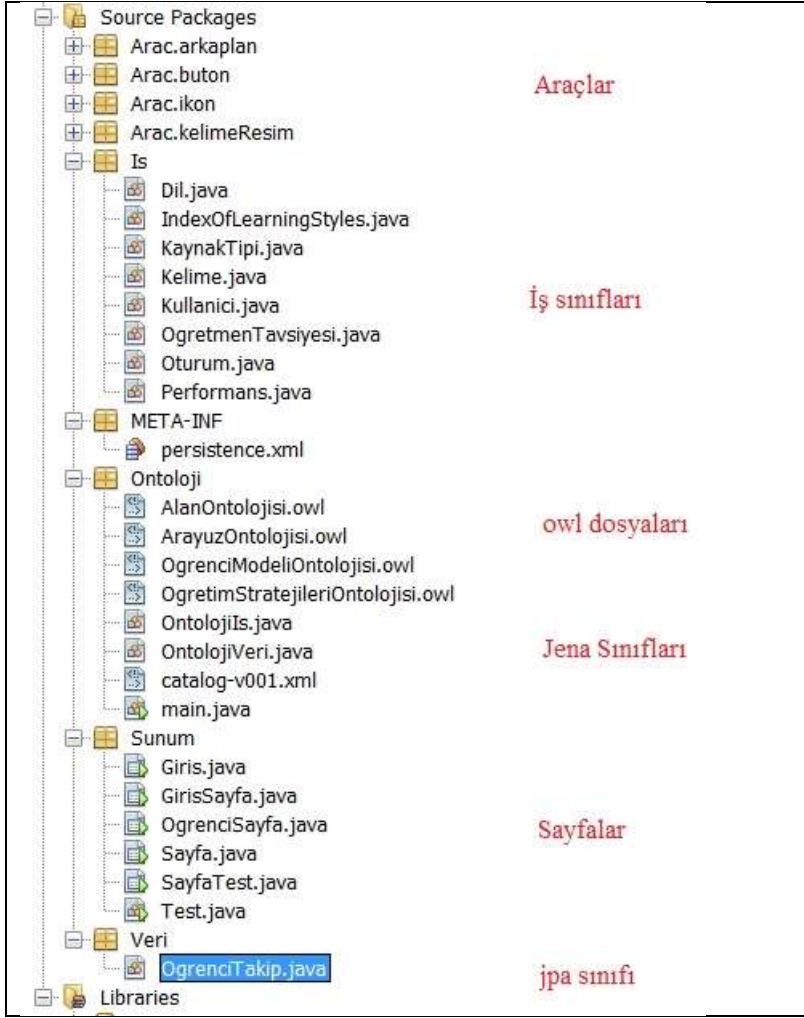
3.5. Java Persistence Api

Ontolojiler veri saklamaktan ziyade bu verileri anlamlandırır. Bu yüzden ortak olan veriler için ontoloji gerekli yapıyı kurduktan sonra bazı veriler veri tabanında tutulur. Bu ontolojinin üzerine yük binmesini engeller ve daha performanslı çalışmasını sağlar. Veri tabanında tutulacak veriler için ise karmaşık bir yapıya ihtiyaç yoktur. Bazen tek bir tablo bile yeterli olabilir.

Bu uygulamada her kullanıcının her kelime için doğru, yanlış ve puan gibi bilgilerini ve kullanıcıların oturum bilgilerini ontolojiler yerine veri tabanında tutulmaktadır. Bu verileri tutmak için java teknolojilerinden Java Persistence Api (JPA) kullanılmıştır. Bu çerçevede bize verilerin erişmesinde oldukça kolaylıklar sağlayan nesne yönelimli bir mimari sağlar.

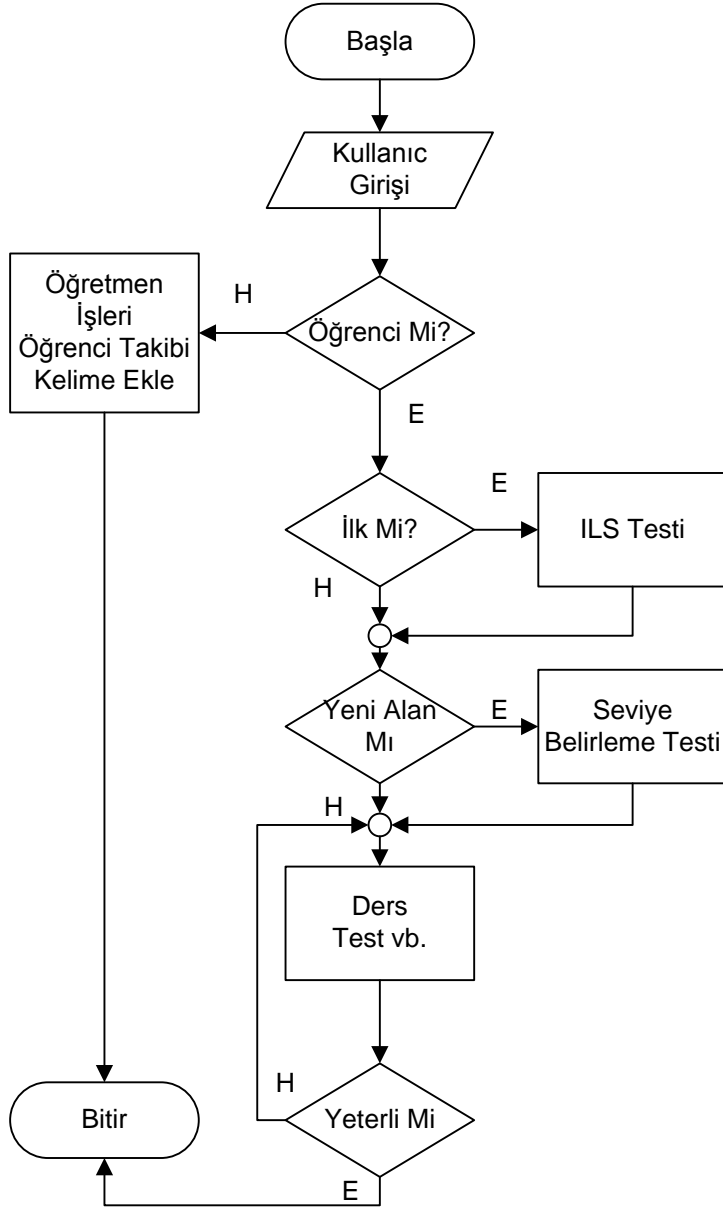
3.6. Uygulamanın Arayüzü ve Uygulamanın Kullanımı

Uygulama NetBeans IDE 8.0 editörü kullanılarak java kodları yazılmıştır. Projenin paketleri ve sınıfları Şekil 3.24'te görülmektedir.



Şekil 3.24. Projenin sınıf hiyerarşisi

Uygulamanın akış diyagramı Şekil 3.25'te verilmiştir. Öğretmen ve öğrenci olmak üzere iki tür kullanıcı vardır. Öğretmen kullanıcısı girişi ise sayfasındaki işlemleri yapıp uygulamayı sonlandırır. Öğrenci kullanıcısı ise; sisteme ilk kez giriyorsa ILS testi uygulanır. Yeni bir alanda çalışmaya başlayacaksa seviye belirleme testi uygulanır. Ders ve test aktiviteleri yeterli ise uygulama sonlandırılır.



Şekil 3.25. Uygulamanın akış diyagramı

Uygulamaya öğrenci girişi yapıldığında eğer kişi ilk kez giriyorsa ILS testi ile öğrencinin öğrenme stili belirlenir. Şekil 3.26’da ILS testine ait bir soru örneği gösterilmiştir. ILS testindeki 44 soru cevaplanınca ILS testi sonuçları görülmektedir. Şekil 3.27’de ILS testi sonuçları gösterilmiştir. ILS testi sonuçlarına göre öğrenci Aktif puanı 5, duyuşsal puan 3, görsel ve genel puanı 1’dir.

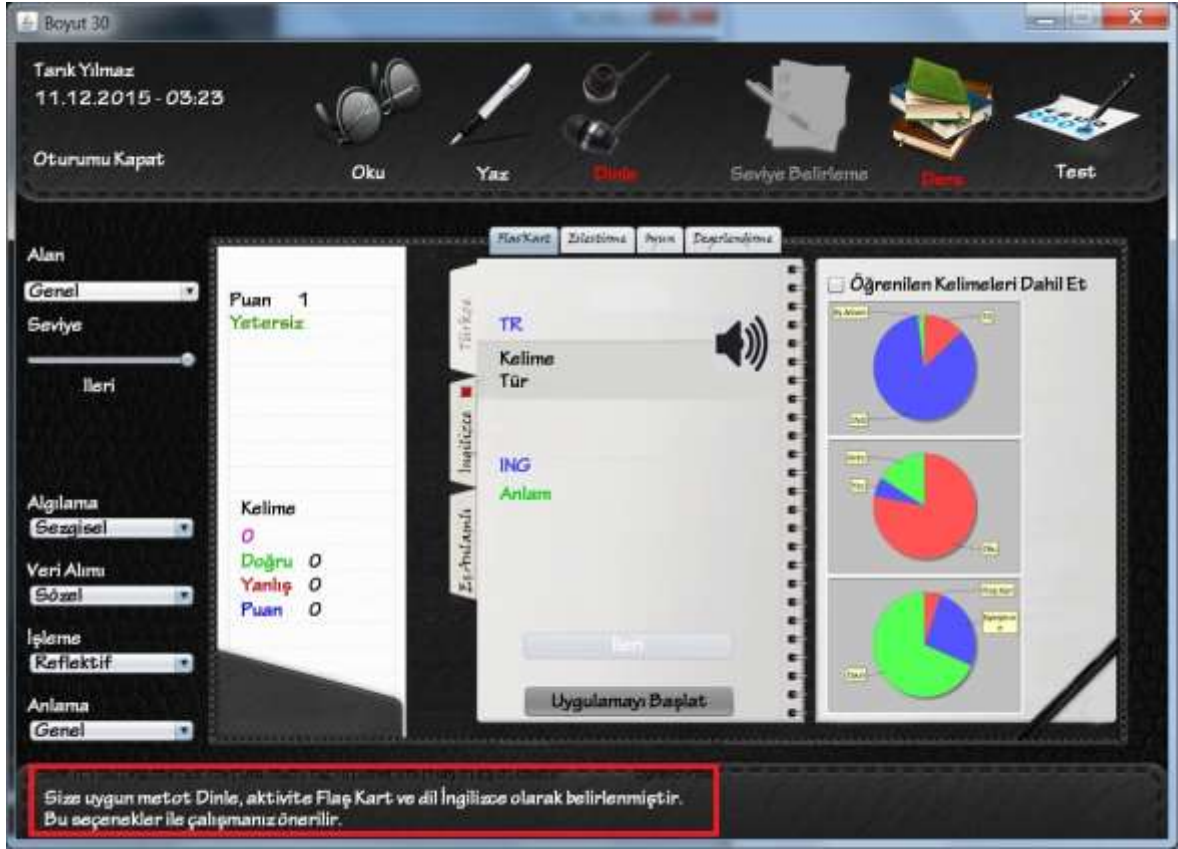


Şekil 3.26. ILS testi



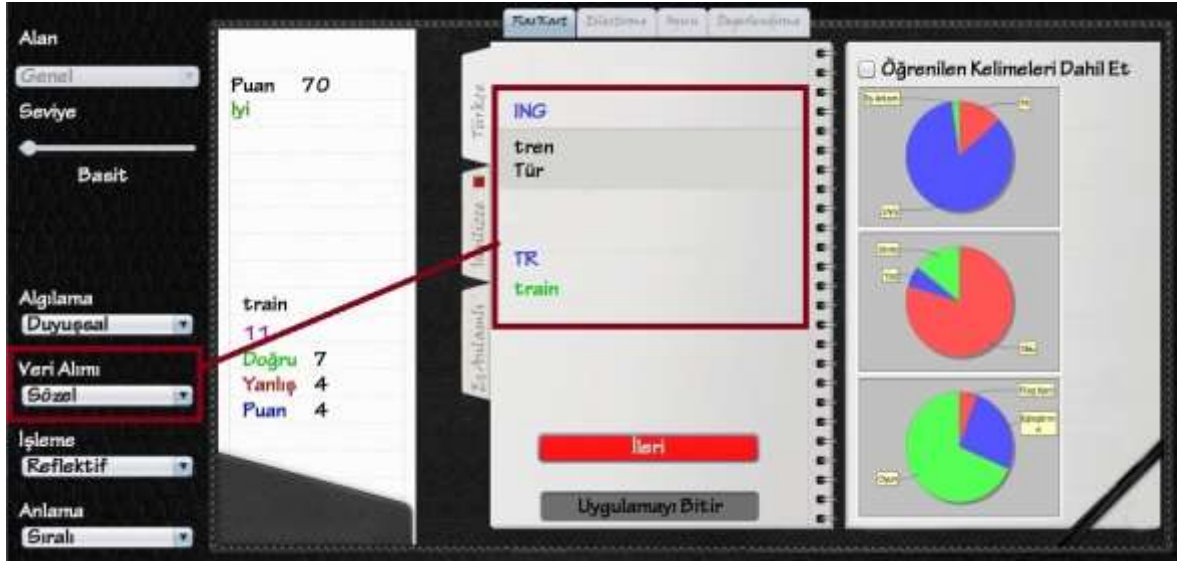
Şekil 3.27. ILS testi sonucu

ILS testi bittikten sonra öğrenci seviye belirleme testini yapar. Seviye tespit sonuçlarına göre öğrenci başarılı ise eğer seviye atlayabilir. Seviye tespit sınavından sonra öğrenci artık çalışmaya başlayabilir. ILS testi sonuçlarına göre öğrenciye bir arayüz sunulur. Bazı direktiflerle öğrenci yönlendirilir. Şekil 3.28’de kırmızı çerçeve ile gösterilmiştir. Çalıştığı her aktivite, metot ve dil seçenekleri süresi ve puanları güncellenir. Bu puanlar grafiksel olarak öğrencinin karşısına gelir ve sürekli güncellenir.

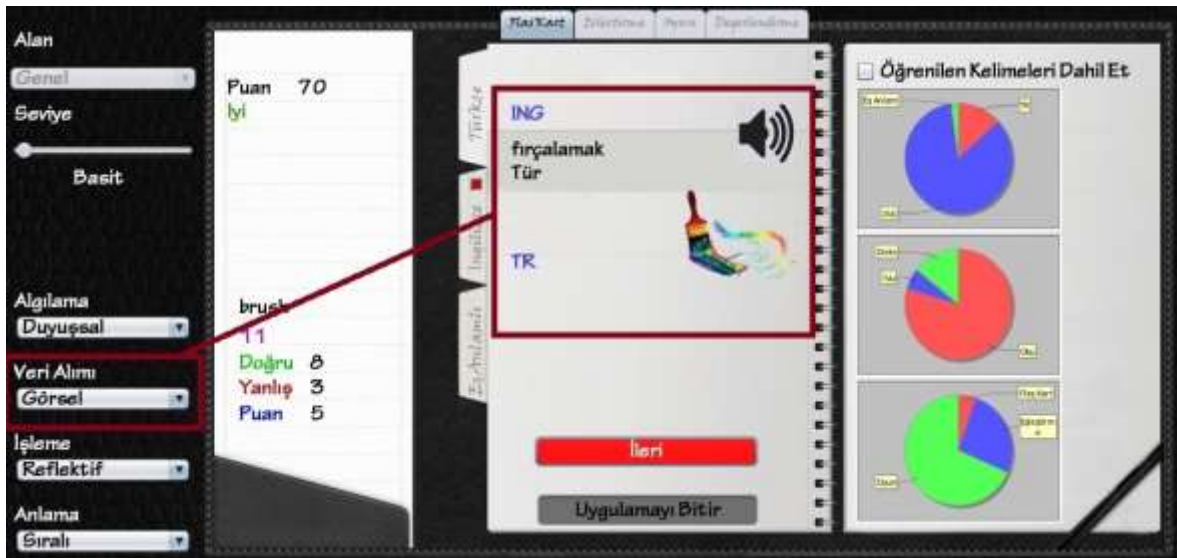


Şekil 3.28. Uygulamadaki öğrenci kullanıcısı arayüzü

Öğrenci öğrenme stiline göre uygun seçenekler ile uygulama başlar. Algılama kategorisinde öğrencinin sezgisel puan daha yüksek ise hafıza oyunları, anagram gibi kısımlar sunulur. Veri Alımı kategorisi sözel puanı yüksek ise metinsel, görsel puanı yüksek ise resimli ve işitsel sunumlar karşısına gelir. İşleme kategorisi reflektif puanı yüksek ise flaş kart, aktif puanı yüksek ise daha çok yazarak ve eşleştirmeli sunumlar kullanılır. Anlama kategorisi genel ise istediği aktiviteleri uygular, sıralı ise bu aktiviteleri ve testleri sırası ile uygular. Şekil 3.29’da Veri Alımı kategorisi sözel olarak seçili iken sunulan arayüz gösterilmiştir. Şekil 3.30’da ise Veri Alımı kategorisi görsel olarak seçili iken sunulan arayüz gösterilmiştir.

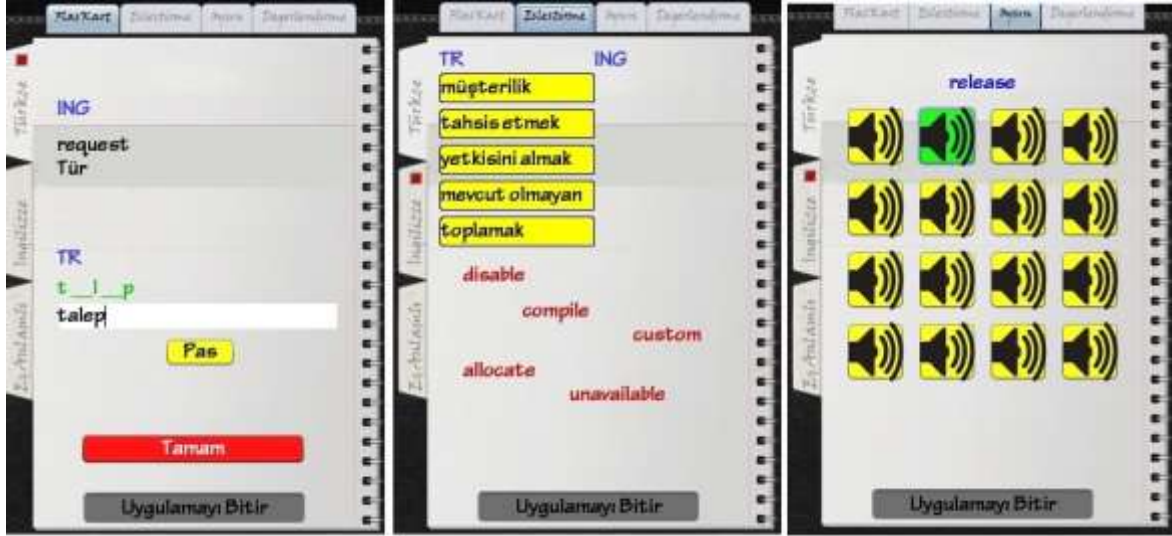


Şekil 3.29. Veri alımı sözel öğrenme stili olan arayüz



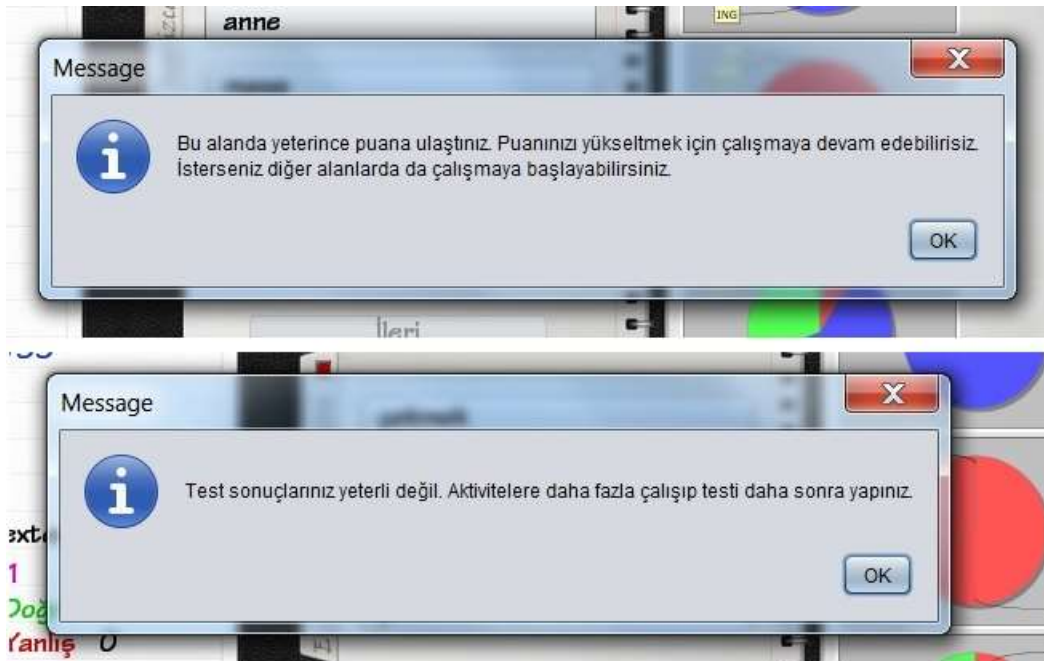
Şekil 3.30. Veri alımı görsel öğrenme stili olan arayüz

Şekil 3.31’de Flaş kart, eşleştirme ve oyun aktivitelerine ait oku, yaz, dinle metotları ve dil tercihlerine göre belirlenen arayüz gösterilmiştir. Aktiviteler de hafıza oyunu, kelime eşleştirme ve yazarak flaş kart sunumları bulunmaktadır. Öğrenci bu uygulamaları kullandıkça öğrenme stilleri öğrenciye uyarlanarak şekillenmektedir.



Şekil 3.31. Uygulamada ki bazı aktiviteler

Öğrenciler bu uygulamayı kullanırken karşısında sanki bir öğretmen varmış hissi olabilmesi için mesaj pencereleri ile yönlendirmeler yapılmıştır. Öğrenci bir aktivitede yeterince çalıştığında, başka bir aktiviteye geçmesi gerektiğinde, test sonuçları yeterli olmadığına bilgi mesajları ile karşılaşmaktadır. Öğrenciyi yönlendiren bu mesajlar sistemin daha doğru bir şekilde kullanılmasına yol açar. Öğrencilerin uygulama esnasında karşılaştığı bazı mesaj kutuları Şekil 3.32’de gösterilmiştir.



Şekil 3.32. Uygulamada ki bazı uyarı mesajları

Öğrenci aktivitelere çalıştıktan sonra öğrenciye değerlendirme ve testler yapılmaktadır. Değerlendirme aktiviteler, metot ve dil seçeneklerini barındırır. Böylece öğrencinin bu seçeneklerine göre puanları oluşmaya başlar. Testte ise sabit bir uygulamadır ve öğrencinin bu çalışmalar sonucunda öğrendiği kelimeyi ölçer. Bir kelimeyi öğrenebilmesi için yanlış cevap vermeden art arda 5 kere doğru cevaplaması gerekir. Öğrenci isterse eğer öğrendiği kelimeleri hem çalışırken hem de değerlendirilirken karşına çıkma olasılığı çok azaltılır. Bilmediği kelimeler daha çok karşısına gelir. Test veya değerlendirmelerde soruların seçeneklerinde zıt anlamlı, benzer yazılışlı kelimeler de bulunur. Eş anlamlı kelimeler aynı soruda bulunmazlar. Soruların bütün seçenekler bilgisayarın rastgele getirdiği kelimelerden oluşmaz. Böylelikle öğrenciler gerçek bir ortamda bir öğretmen tarafından hazırlanmış gibi olan soru ve cevaplar ile karşılaşmaktadır. Şekil 3.33'te görüldüğü gibi cevap *release* olan sorunun şıklarında kelimenin zıt anlamı olan *retain*, yazılışı benzer olan *represent* bulunmaktadır. Kelimenin eş anlamlısı olan *abandon* ve *independent* şıklarda bulunmamaktadır.



Şekil 3.33. Ontolojide kelime nesne özelliklerine göre soru ve cevaplarının hazırlanması

Uygulamaya öğretmen girişi yapıldığında Şekil 3.34'teki sayfa gelmektedir. Bu sayfada öğretmen her öğrencisinin ayrı ayrı performans grafiklerini, oturum açma/kapatma tarih ve saatlerini, her alandaki puanlarını takip edebilmektedir. Ayrıca ontolojiye yeni kelime ekleyebilmektedir. Bu sayede ontolojide ki kelime haznesi giderek artabilir. Kelime eklerken eş anlamlı, zıt anlamlı ve benzer yazılışlı özellikleri ilişkilendirilmektedir. Kelimenin var olan alanlardaki anlamını yazıp veri özelliklerini de ilişkilendirir. Yanlış ilişkilendirme olmaması için kelime ekleme işlemi uzman kişilere yetki verilmiştir.

The screenshot shows the 'Boyut 58' application interface for a teacher user. The interface is dark-themed and contains several sections:

- Header:** 'Ahmet Eren' (user name) and 'Oturumu Kapat' (Close Session).
- Öğrenciler (Students):** A list showing 'Taha Yılmaz' and 'Tanık Yılmaz'.
- Öğrendiği Kelimeler (Words Learned):** A dropdown menu set to 'Başlı' and a score of 'Puan: 70 İy'.
- Oturum (Session):** A table showing session dates and times.

Açılış / Kapanış
23.09.2014 - 10:35 / 23.09.2014 - 10:36
05.10.2015 - 23:07 / 05.10.2015 - 23:15
07.11.2015 - 18:30 / 07.11.2015 - 18:35
07.11.2015 - 23:27 / 07.11.2015 - 23:37
08.11.2015 - 19:42 / 08.11.2015 - 20:00
- Performans (Performance):** Three pie charts showing performance data.
- Alan Ekleme (Add Field):** An input field and a button labeled 'Alan Ekle'.
- Kelimeler Ekleme (Add Words):** A list of words including 'abandon', 'absolute', 'advance', 'advantage', 'adverse', 'allocate', 'allow', and 'appeal'. Below the list are buttons for 'Eş Anlam' (Synonym) and 'Zıt Anlam' (Antonym).
- Other Sections:** 'Eş Anlam' (Synonym) and 'Zıt Anlam' (Antonym) buttons, and a 'Benzer Yazılıp' (Similar Written) section with a 'Ekle' (Add) button.

Şekil 3.34.Uygulamadaki öğretmen kullanıcısı arayüzü

4. GELİŞTİRİLEN SİSTEMİN DEĞERLENDİRİLMESİ

4.1. Yöntem

Uygulamanın uzman ve kullanıcı değerlendirilmeleri yapılmıştır. Uzman değerlendirilmesi için en az 10 yıldır bilgisayar kullanan ve 8 yıldan fazla süredir İngilizce eğitimi veren 6 uzman belirlenmiştir. Bu uzmanlara uygulama tanıtılmıştır. Daha sonra uygulama ile ilgili görüşlerini almak için 7 adet değerlendirme sorusu sorulmuştur. Bu sorular genelde açık uçlu olup sistemi tüm yönüyle değerlendirmeye yöneliktir. Uzman değerlendirmesi için kullanılan sorular Çizelge 4.1’de gösterilmiştir.

Çizelge 4.1. Uzman değerlendirme soruları

Sorular
1) Uygulama amacına uygun mu?
2) Uygulama kullanışlı ve anlaşılır mı?
3) Uygulama tasarımını sizce nasıldır?
4) Uygulama erişimi ve kullanımı kolay mı?
5) Uygulamanın güçlü yönleri nelerdir?
6) Uygulamanın zayıf yönleri nelerdir?
7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Uygulamanın kullanıcılar tarafından değerlendirilmesi için Open ECB Check olarak bilinen ve e – öğrenme sistemlerini değerlendirmeye yönelik olan anket uygulanmıştır. 20 kişilik gruplar halinde kullanıcılara önce sistem tanıtılmıştır. Yaklaşık 40 dakika süren bu tanıtım sonrası her gruptan 2 kullanıcı sistemi kullanmıştır. Daha sonra kullanıcılara anket uygulanmıştır. Open ECB Check ilk olarak 2010 yılında Lockee ve ark. Tarafından uygulanmış ve memnun edici sonuçlar elde edilmiştir. Open ECB Check altı kategoriden oluşmaktadır. Bu kategoriler çevresel boyutu, öğrenci boyutu, öğretim boyutu, ders boyutu, tasarım boyutu ve teknoloji boyutu’dur. Her boyut için üçer adet olmak üzere toplam 15 adet anket sorusu mevcuttur. Çevresel boyut, beş boyut için yapılan anket sonucunda öğrenciden alınan geri dönüşlerle değerlendirilir [41]. Kullanıcı değerlendirmesi için kullanılan anket soruları ve boyutları Çizelge 4.2’de gösterilmiştir.

Çizelge 4.2. Kullanıcı değerlendirme anket soruları ve boyutları

Boyut	Sorular
Öğrenci	1) Bir e-öğrenme platformuna ne kadar alışkınsınız. 2) E-öğrenme amaçlı bilgisayar/dizüstü bilgisayar/mobil cihaz kullanımına yönelik tutumunuz. 3) Sunulan e-öğrenme içeriği anlaşılır.
Öğretim	4) Öğretmen, öğrencinin amaçlarını karşılayacak şekilde düzenlemiştir. 5) E-öğrenme içeriği sistem tarafından zamanında güncellenir. 6) Sınav için yeterli süre verilmiştir.
Ders	7) Ders içeriği kişiselleştirilmiştir. 8) Konuların kapsamı dersin amacına uygundur. 9) Ders içeriği açık ve anlaşılırdır.
Tasarım	10) Arayüz tasarımı ilgi çekici ve kullanışlıdır. 11) Tasarımda resim, ses ve metin materyalleri yeterlidir. 12) E-öğrenme sistemi ihtiyaca cevap verir.
Teknoloji	13) Sisteme erişim kolaydır. 14) Sistem öğrenme stiliinizi belirlemede iyidir? 15) Öğrenme stilleri sisteme yansıtılmıştır.

Anketi yapmak için Ankara Yenimahalle’de bulunan Gazi Teknik ve Anadolu Meslek Lisesi Okulu 10, 11 ve 12. Sınıf olmak üzere 120 öğrenci belirlenmiştir. Öğrencilerin yaş aralığı 15-18 olup 108’i erkek, 12’si kızdır. Anket soruları 0-4 puan arasında derecelendirilmiştir.

4.2. Sonuçlar

Uzman değerlendirme sonuçlarına bakıldığında sonuçlar memnun edicidir. Uygulamanın amacına uygun, kullanışlı ve sade bir e - öğrenme ortamı olduğu söylenmiştir. Beş duyu organına hitap eden kullanıcı dostu bir arayüz olarak tasarlandığı belirtilmiştir. Yine uygulamanın erişimi ve kullanımı kolay olduğu 6 uzman tarafından belirtilmiştir.

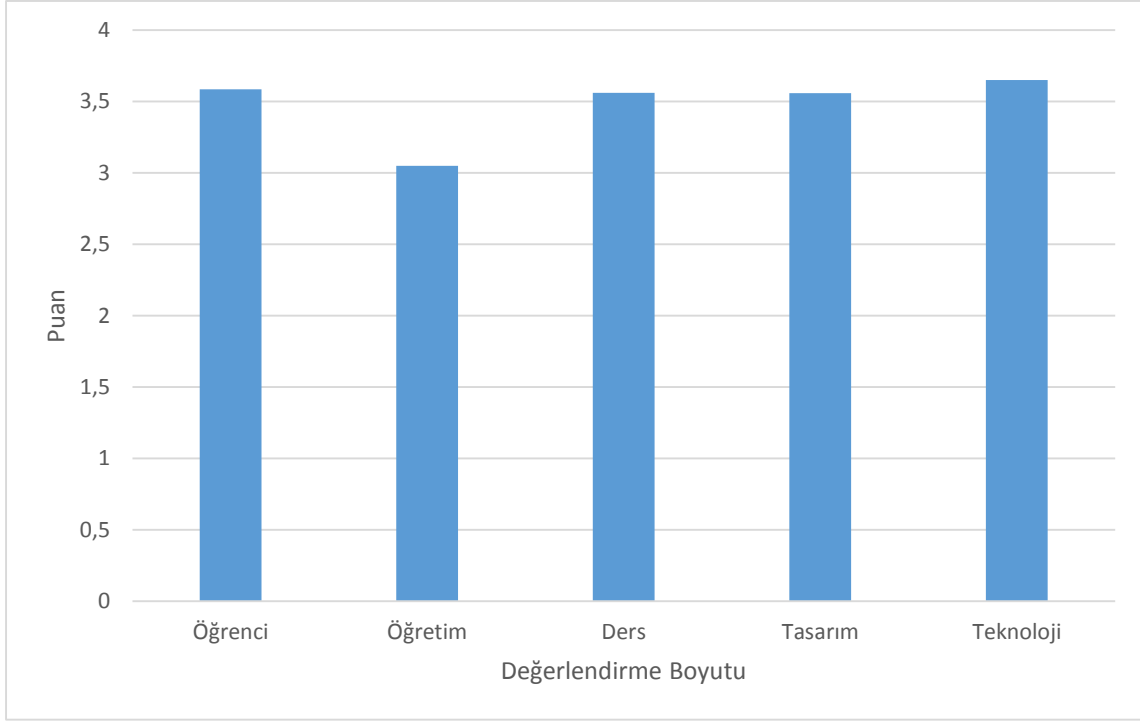
Uygulamanın güçlü yönleri öğrenme stillerinin olması, öğrenci seviyesine ve seçeneğine göre uygun arayüz sunuluyor olması, kişisel öğrenme ortamının bulunması, eş anlamlı ve zıt anlam kelime sunumu ile az zamanda daha çok kelime öğrenilmesi ve kelimenin anlamlarının kullanıldığı alanda veriliyor olması olarak ifade edilmiştir.

Uygulamanın zayıf yönleri olarak kelimelerin cümle içerisinde kullanılmaması ve kelimelerin farklı türdeki anlamlarının olmaması belirtilmiştir. Kelime öğreniminin daha yararlı olabilmesini için bu özelliklerin gerekliliği ortaya konmuştur.

Mevcut uygulamalara göre avantaj ve dezavantajları açısından incelendiğinde uygulamanın sınav İngilizcesi için kullanılabilir olmasının, kelimelerin görsel ve sözel sunumlarının bulunmasının olumlu olduğu ifade edilmiştir. Ancak, kelimelerin görsel sunumlarının zor kelimelerde problemlili olabileceği ve kelimelerin kullanıldıkları bağlamdan yoksun olarak verilmesinin kelime üretimini azaltabileceği belirtilmiştir.

Uzman değerlendirme sonuçlarında 6 uzmanın da olumlu görüş verdiği görülmüştür. En olumlu geri dönüş öğrenme stillerine yönelik hazırlanmış olması, 5 duyu organına hitap ediyor olması, eş anlamlı ve zıt anlamlı kelime öğrenimi sağlıyor olması, terimlerin ve kelimelerin anlamlarını onların kullandığı alanlarda veriyor olması olarak belirtilmiştir. En büyük eleştiri kelimelerin cümle içerisinde olmaması, kelime sayısı ve çeşitliliğin az olması olarak belirtilmiştir. Bir uzman ise mevcut uygulamaların kelimeleri kullanıldıkları bağlamdan yoksun olarak vermesi İngilizce öğrenimini ve üretimini azaltabilir olacağını belirtmiştir.

Değerlendirmenin ikinci kısmı olan kullanıcı değerlendirmesinin sonuçlarını almak için önce uygulama öğrencilere tanıtılmış ve öğrencilerin uygulamayı kullanmaları sağlanmıştır. Daha sonra yöntem kısmında belirtildiği şekilde öğrencilere değerlendirme anketi uygulanmıştır. Cevaplar 4 puan üzerinden değerlendirilmiş ve elde edilen sonuçlar Şekil 5.1'deki grafikte gösterilmiştir. Anket uygulamasından elde edilen sonuçlara göre öğrenci boyutu için 3.59, öğretim boyutu için 3.05, ders boyutu için 3.56, tasarım boyutu için 3.56, teknoloji boyutu için 3.65 ortalama puan elde edilmiştir. Bu sonuçlar bize uygulamanın kullanıcı memnuniyeti açısından başarılı olduğunu göstermektedir.



Şekil 4.1. Anket sonuçları

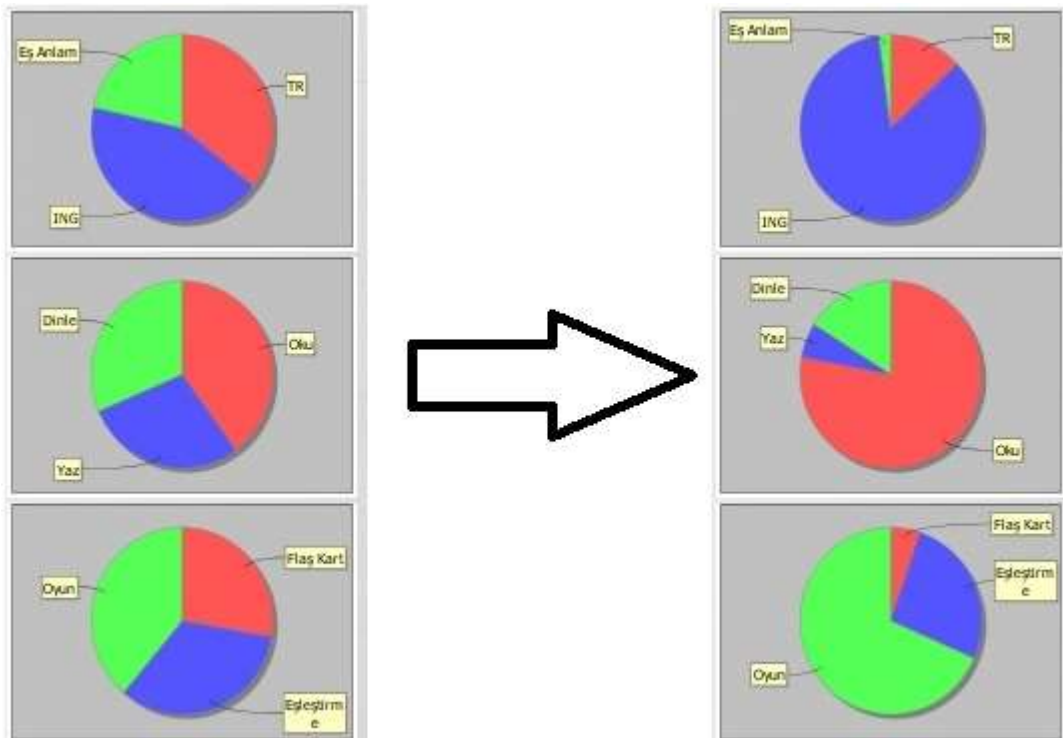
Uygulamanın öğrencileri üzerinde heyecan uyandırıcı bir etkisi olduğu gözlemlenmiştir. Özellikle uygulamanın kişiselleştirilmesi ve öğrenme stillerinin belirlenip arayüzün bu öğrenme stillere göre sunulması öğrencilere dikkat çekici gelmiştir. Öğrencilerin bilişim sistemlerine gün geçtikçe daha çok alışkın olmaya başlamaları bu uygulamaların daha çok kullanılmasına olanak sağlamaktadır.

Uygulamanın kullanılması ile öğrenci verileri sürekli değiştiği gözlemlenmektedir. Şekil 5.2’de ilk kısım öğrenci veri özelliklerinin protege editörü ile hazırlanan başlangıç değerleri görülmektedir. Owl dosyası ile sisteme aktarılan bu bilgiler, öğrenci kullanımı ile sürekli değişerek yeni değerleri ikinci kısımda görüldüğü gibi jena tablolarında tutulmaktadır. Şekilde *BasitDerece* 0 değeri sistemin kullanımıyla 85 değerine değiştiği görülmektedir (Bkz Şekil 5.2).



Şekil 4.2. Öğrenci veri özelliklerindeki değerler

Öğrenci sistemi kullandıkça dil tercihlerindeki Türkçe, İngilizce ve eş anlamlı sunumları, metotlardaki oku, yaz ve dinle sunumları ile aktivitelerdeki oyun, flaş kart ve eşleştirme sunumlarındaki süreler tutulmaktadır. Bu verilere göre değerlendirme esnasında da bu sunum puanları anlık olarak değişmektedir. Bu değişimi öğrenci takip edebilmektedir. Şekil 5.3'te öğrenci uygulamayı kullanıldıkça sunumlarda ki puanların değişimi görülmektedir. Böylece uygulamanın öğrenciye daha çok uyumlu hale geldiği görülmektedir.



Şekil 4.3. Öğrenci puanlarını gösteren grafikler

5. SONUÇ VE TARTIŞMA

Ontoloji tabanlı zeki öğretim sistemleri çalışmaları yaygınlaşmaktadır. Daha önceki çalışmalarda kullanılan bilgisayar destekli öğretim ile dersler daha etkili ve verimli olmaktadır. Öğrenmenin mekandan ve zamandan bağımsızlığı büyük öneme sahiptir. Fakat bilgisayar destekli öğretimde bireysel öğrenme ve gerçek zamanlı etkileşim mevcut değildir. Bu sorun zeki öğretim sistemleri ile giderilmektedir. Öğrenci takibi, öğrenme sürecindeki yönlendirme, öğrenciye uygun pedagojik yöntemler ile ZÖS’nde daha önce bahsedildiği gibi BDÖ’e göre daha iyi sonuçlar elde edilmiştir. ZÖS modüllerinin oluşturulmasında anlamsal web daha iyi teknikler sunmaktadır. Ontoloji tabanlı ZÖS’nde modüller arası iletişim oldukça güçlüdür. Bu çalışmadaki uygulama ile yabancı dilde kelime öğrenimi için dört adet ontoloji kurulmuştur. Bu ontolojiler birleştirilerek aralarında iletişim sağlanmıştır. Daha önceki ZÖS ile yapılan çalışmalardan farklı olarak bu uygulamada alan ontolojisinde ki kelimelerin sayısı ve anlamları öğretmen kullanıcı ile sürekli genişlemektedir. Yeni kategori oluşturabilir ve eklenen kelimelerin eş anlamlı, zıt anlamlı, benzer yazılımlı ilişkilerini çalışma sırasında kurabilir. Uygulamada sunulan test seçeneklerinin doğru cevap olan kelime ile eş anlamlı olmaması sağlanmıştır. Seçeneklerin bazılarının hemen elenebilmesi bazılarının ise hedefe ulaşmada biraz daha düşündürmek için zıt anlamlı ve benzer yazılımlı kelimeler kullanılmıştır. Bu sayede bir öğretmen yaklaşımı benzetilmiştir. Böylece gerçeğe yakın ve ilerlenebilir bir uygulama olduğu görülmektedir.

Bu sistemi daha önce yapılan yabancı dilde kelime öğrenmeye yönelik sistemler ile karşılaştırıldığında en büyük avantajı ontoloji tabanlı bir sistem olmasıdır. Kelimeler arasındaki ilişkiler ve sınıflandırmalar, öğrenci profilinin yapısı, öğrenme stillerinin uyumu ontoloji tabanlı sistemin güçlü özellikleridir. Zeki öğretim sistemleri konsepti ile oluşturulması bir öğretmenin rehberliğinde çalışma hissi ve kişisel öğrenmeyi sağlamıştır. Böylece diğer sistemlerdeki rastgelelik ve her öğrenci için aynı çalışma ortamının dezavantajlarını ortadan kaldırmıştır. ILS testi ile öğrencinin öğrenme stilinin belirlenmesi, öğrenme stilini desteklemeyen uygulamalara göre artı bir özelliktir. Ayrıca sistemdeki verilerin kalıcı olabilmesi için dosya yerine veritabanında tutulması sistemli bir yapı sağlaması dışında yüksek performans sunar.

Uygulamanın uzman değerlendirilmesi ve kullanıcı değerlendirmesi sonuçlarının oldukça olumlu olduğu görülmüştür. Uygulamada bulunan kelimelerin sayısı, yapısı ve aralarındaki

ilişkiler daha da çoğaltılarak geniş kapsamlı bir içeriğe sahip olan bir sistem olması düşünülmektedir.

ZÖS ile hazırlanan uygulamaların başarılı sonuç vermesi gelecekte de kullanılması kaçınılmazdır. Anlamsal web teknolojilerinin de kullanımının artması ontoloji tabanlı ZÖS ile birçok uygulama geliştirileceği düşünülmektedir. Yabancı dilde kelime öğrenme hem genç hem de yetişkin insanlarda sürekli ihtiyaç hissedilmektedir. Yabancı dil sınavları, ikinci bir dil öğrenimi çoğu insanın her zaman hayatlarında yer edinmiştir. Bu öğrenmenin en kısa sürede, etkili ve verimli olabilmesi önem kazanmıştır. Bu yüzden bu alanda daha fazla çalışma yapılması beklenmektedir.

KAYNAKLAR

1. Sanagustin M. P., Gonzalez G. R., Leo D. H., Organero M. M, Santo P., Blat J. and Kloos C.D. (2012), Discovering the campus together: A mobile and computer-based learning experience, *Journal of Network and Computer Applications*, 35, 176-188.
2. Nikoopour J., Kazemi A. (2014), Vocabulary Learning through Digitized & Non-digitized Flashcards Delivery, *Procedia Social and Behavioral Sciences*, 98, 1366-1373.
3. Ferreira A., Atkinson J. (2009), Designing a feedback component of an intelligent tutoring system for foreign language, *Knowledge-Based System* 22, 496-501.
4. Gharehchopogh F. S. , Khalifelu Z. A. (2011), Using Intelligent Tutoring Systems in Instruction and Education, *International Conference on Education and Management Technology*, 13, 250-254.
5. Kaya S. (2005), *Microsoft Excel Öğretimi için Zeki Öğretim Sistemi*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
6. Keleş A. (2007), *Öğrenme- Öğretme Sürecinde Yapay Zekâ ve Web Tabanlı Zeki öğretim Sistemi Tasarımı ve Matematik Öğretiminde Bir uygulama*, Doktora tezi, Atatürk Üniversitesi Fen Bilimleri Enstitüsü, Erzurum.
7. İnternet: Stottler Henke Associate Intelligent Tutoring Systems: Using AI to Improve Training Performance and ROI. URL: http://www.stottlerhenke.com/solutions/training/its_background.htm, Son Erişim Tarihi: 08/01/2014.
8. Aparico F., De Buenaga M., rubio M. and Hernando A. (2012), An Intelligent Information Access System Assisting a Case Based Learning Methodology Evaluated in Higher Education with Medical Students, *Computers & Education*, 58, 1282-1295.
9. Mizoguchi R., Bourdeau J. (2000), Using Ontological Engineering to Overcome Common AI-ED Problems, *International Journal of Artificial Intelligence in Education*, 11, 107-121.
10. Vesin B., Ivanović M., Milićević A. K. and Budimac Z. (2013), Ontology-Based Architecture with Recommendation Strategy in Java Tutoring System, *Computer Science and Information Systems*, 10(1), 237-261.
11. Vesin B., Ivanović M., Milićević A. K. and Budimac Z. (2012), Protus 2.0: Ontology-Based Semantic Recommendation Strategy in Programming Tutoring System, *Expert System with Applications*, 39, 12229-12246.
12. Gladun A., Rogushina J. (2007), An Ontology-Based Approach To Student Skills in MultiAgent E-Learning Systems, *International Journal*, 1, 219-225.

13. Oguejiofor E., Kicing R., Popovici E., Arciszewski T. and De Jong K. (2004), Intelligent Tutoring Systems: An Ontology-Based Approach, *International Journal of IT in Architecture*, 2, 115-128.
14. Jacinto A.S., Parente de Oliveira J. M. (2008), An ontology-based architecture for Intelligent Tutoring System, *Scientia-Interdisciplinary Studies in Computer Science*, 19(1), 25-35.
15. Escudero H., Gonzalez R. F. (2010), Generating Ontologies from Intelligent Tutoring System Courses. A Generic Approach, *IOS Press*, 1, 1-9.
16. Cheng Y.T. (2007), *Algorithm and Intelligent Tutoring System Design for Ladder Logic Programming*, Master's Thesis, Texas A&M University, Texas.
17. Duo S., Ying C. (2012), Personalized E-learning System Based on Intelligent Agent, *Physics Procedia*, 24, 1899-1902.
18. Mitrovic A. (2004), Devedzic V., A Model of Multitutor Ontology-Based Learning Environments, *Int. J.Cont. Engineering Education and Lifelong Learning*, 14(3), 229-245.
19. Barros H., Silva A., Costa E., Bittencourt I. I., Holanda O. and Sales L. (2011), Steps, techniques and Technologies for the Development of Intelligent applications Based on Semantic Web Services: A Case Study in E-learning Systems, *Engineering Applications of Artificial Intelligence*, 24, 1355-1367.
20. Waalkens M., Alevin V. and Taatgen N. (2013), Does Supporting Multiple Student Strategies Lead to Greater Learning and Motivation? Investigating a Source of Complexity in the Architecture of Intelligent Tutoring system, *Computers & Education*, 60, 159-171.
21. Latham A., Crockett K. and McLean D. (2014), An Adaptation Algorithm for an Intelligent Natural Language Tutoring System, *Computers & Education*, 71, 97-110.
22. Özyurt Ö., Özyurt H. and Baki A. (2013), Design and Development of an Innovative Individualized Adaptive and Intelligent, E-learning System for Teaching-Learning of Probability Unit: Details of UZWEBMAT, *Expert Systems with Applications*, 40, 2914-2940.
23. Allahverdi N. (2002). *Uzman Sistemler*, 12. İstanbul: Atlas, 3-4, 15-22, 43-60.
24. Yu X., Wei Y., Huang D., Jiang Y, Liu B. and Jin Y. (2011), Intelligent Switching Expert System for Delayed Coking Unit Based on Iterative Learning Strategy, *Expert Systems with Applications*, 38, 9023-9029.
25. Alpaydın E., Yapay Öğrenme, 1, *Boğaziçi Üniversitesi*, İstanbul, 153-169 (2011).

26. İnternet: Bilgisayar Kavramları, Karar Ağacı Öğrenmesi. URL: <http://bilgisayarkavramlari.sadievrenseker.com/2012/04/11/karar-agaci-ogrenmesi-decision-tree-learning/>, Son Erişim Tarihi: 08/12/2015.
27. Silahtaroğlu G. (2013), *Veri Madenciliği*, 2, İstanbul: Papatya, 12, 67-74.
28. Elmas Ç. (2011), *Yapay Zeka Uygulamaları*, 2, Ankara: Seçkin, 41-43, 88, 137.
29. Öztemel E. (2012), *Yapay Sinir Ağları*, 3, İstanbul: Papatya, 29, 91-93.
30. Moise G. (2010), *Usage of the Artificial Neural Networks in the Intelligent Tutoring System*, Paper presented at the first The 5th International Conference on Virtual Learning ICVL, Romania.
31. Üstün O., Yıldız İ. (2009), Geri-Yayımlı Öğrenme Algoritmasındaki Öğrenme Parametrelerinin Genetik Algoritma ile Belirlenmesi, *SDU International Technologic Science*, 1(2), 61-73.
32. Doğan B. (2006), *Zeki Öğretim Sistemlerinde Veri Madenciliği Kullanılması*, Doktora Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
33. Blagojevic M., Micic Z. (2013), A Web-Based Intelligent Report E-Learning System Using Data Mining Techniques, *Computers and Electrical Engineering*, 39, 465-474.
34. İnternet: Vizyon 21.Yüzyıl Semantik Web. URL: http://vizyon21yy.com/documan/Genel_Konular/Bilisim/Semantik_Web.pdf, Son Erişim Tarihi: 08/12/2015.
35. Berners-Lee T., Hendler J. and Lassila O. (2001), A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, *Scientific American*.
36. Türksoy H. (2007), *Ontoloji Tabanlı Etkinlik ve Öğrenme Nesnesi Paylaşım Sistemi*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
37. İnternet: W3C Linked Data. URL: <http://www.w3.org/standards/semanticweb/data>, Son Erişim Tarihi: 08/12/2015.
38. İnternet: Corporate Semantic Web. URL: http://bortecin.com/Semantic_Web.pdf, Son Erişim Tarihi: 08/12/2015.
39. Kazandır Z. B. (2011), *Anlamsal Web Tabanlı Kütüphane Bilgi Sistemi*, Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü, Ankara.
40. Gruber T. R. (1995), Toward principles for the design of ontologies used for knowledge sharing , *Int. J. Human- Computer Studies*, 43, 907-928.
41. Rani M., Nayak R. and Vyas O. P. (2015), An ontology-based adaptive personalized learning system, assisted by software agents on cloud storage , *Knowledge - Based Systems*, 90, 33-48.

42. Kara G., Cömert Ç. (2011), *Ulusal Konumsal Veri Altyapısı için Semantik Veri Tanımlama*, Tmmob Coğrafi Bilgi Sistemleri Kongresinde sunuldu, Antalya.
43. İnternet: W3C RDF Primer. URL: <http://www.w3.org/TR/rdf-primer/>, Son Erişim Tarihi: 08/12/2015.
44. İnternet: W3C OWL Web Ontology Language URL: <http://www.w3.org/TR/owl-features/>, Son Erişim Tarihi: 08/12/2015.
45. İnternet: The University of Manchester, Protege OWL Tutorial. URL: <http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/>, Son Erişim Tarihi: 08/12/2015.
46. Van Zwanenberg N., Wilkinson L. J. (2000), Felder and Silverman's Index of Learning Styles and Honey and Mumford's Learning Styles Questionnaire: how do they compare and do they predict academic performance?, *Educational Psychology*, 20(3), 365-380.
47. Felder R. M., Spurlin J. (2005), Application, Reliability and Validity of the Index of Learning Styles, *International Journal Engineering Education*, 21(1), 103-112.
48. Scott E., Rodriguez G., Soria A. and Campo M. (2014), Are learning styles useful indicators to discover how students use Scrum for the first time? *Computers in Human Behavior*, 36, 56-64.
49. Latham A., Crockett K., McLean D. and Edmonds B. (2012), A conversational intelligent tutoring system to automatically predict learning styles, *Computer & Education*, 59, 95-109.
50. Çağiltay K. (2011), *İnsan Bilgisayar Etkileşimi ve Kullanabilirlik Mühendisliği: Teoriden Pratiğe, 1*, Ankara: ODTÜ Yayıncılık, 67-68.
51. Zhong L., Zheng M., Yuan J. and Jin J. (2012), The Jena-Based Ontology Model Interface and Retrieval Application, *Scientific Research*, 4, 157-160.

EKLER

EK-1(a). Uzman değerlendirme cevapları

UZMAN DEĞERLENDİRME ANKETİ

Eğitmenin;

Ad Soyadı	: Uzman#1
Doğum Yılı	: 1981
Eğitmenlikte çalıştığınız süre (yıl)	: 13
İngilizce eğitmenliğinde çalıştığınız süre (yıl)	: 13
Kaç yıldır bilgisayar kullanıyorsunuz?	: 20

1) Uygulama amacına uygun mu?

Evet

2) Uygulama kullanışlı ve anlaşılır mı?

Oldukça kullanışlı ve sade olduğu içinde kolay anlaşılır.

3) Uygulama tasarımını sizce nasıldır?

Kullanıcı dostu bir arayüz olduğunu söyleyebilirim.

4) Uygulama erişimi ve kullanımı kolay mı?

Evet.

5) Uygulamanın güçlü yönleri nelerdir?

Yönlendirmeleri olması, sözel ve görsel veri kullanıyor olması, seviyeye uygun arayüz sunuyor olması uygulamanın artıları.

6) Uygulamanın zayıf yönleri nelerdir?

Uygulama hedefleri dahilinde değerlendirilirse çok zayıf noktaları olduğunu düşünmüyorum. Sadece kelimeler için cümleler hazırlanmış olması daha yararlı olabilirdi. Böylece kelimelerin kullanımı içinde görmekte mümkün olabilirdi. Ayrıca, oluşturma sayıları ve çeşitleri artırılabilir.

7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Kelime öğretimi konusunda hem sözel hem de görsel yaklaşımı kullanıyor olması önemli ancak görsel yöntemin öğretilmesi zor kelimelerde problem yaşanabilir.

EK-1(b). Uzman değerlendirme cevapları

UZMAN DEĞERLENDİRME ANKETİ

Eğitmenin;

Ad Soyadı	: Uzman#2
Doğum Yılı	: 1982
Eğitmenlikte çalıştığınız süre (yıl)	: 8
İngilizce öğretmenliğinde çalıştığınız süre (yıl)	: 8
Kaç yıldır bilgisayar kullanıyorsunuz?	: 16

1) Uygulama amacına uygun mu?

Uygun

2) Uygulama kullanışlı ve anlaşılır mı?

Gayet açık, programın kullanımı ve öğrenimi gayet kolaydır.

3) Uygulama tasarımını sizce nasıldır?

Kullanımı keyifli, öğrenme stilleri ve 5 duyuyu merkezine alan güzel bir bilgisayar destekli kelime öğrenme programı olmuş.

4) Uygulama erişimi ve kullanımı kolay mı?

Evet

5) Uygulamanın güçlü yönleri nelerdir?

Öğrenme stillerine yönelik hazırlanmış. 5 duyu organına hitap ediyor. Eş anlamlı ve zıt anlamlı kelime öğrenimi sağlıyor. Böylelikle daha az zamanda daha çok kelime öğrenimini kolaylaştırıyor. Terimlerin ve kelimelerin anlamlarını onların kullandığı alanlarda veriyor.

6) Uygulamanın zayıf yönleri nelerdir?

Cümle içinde kullanımına örnek yok. Kelimelerin farklı türlerini (sıfat, zarf, fiil vb) vermiyor.

7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Program sınav İngilizcesi gruplarında kullanılabilir. Bu tür programların kelimeleri kullanıldıkları bağlamdan yoksun olarak vermesi İngilizce öğrenimini ve üretimini azaltabilir.

EK-1(c). Uzman değerlendirme cevapları

UZMAN DEĞERLENDİRME ANKETİ

Eğitmenin;

Ad Soyadı	: Uzman#3
Doğum Yılı	: 1961
Eğitmenlikte çalıştığınız süre (yıl)	: 24
İngilizce öğretmenliğinde çalıştığınız süre (yıl)	: 24
Kaç yıldır bilgisayar kullanıyorsunuz?	: 14

1) Uygulama amacına uygun mu?

Evet. Öğrenme stilleri çok iyi. Flaş kart ve eşleştirmeyi beğendim.

2) Uygulama kullanışlı ve anlaşılır mı?

Evet. Rahat kullanabildim. Her şey gözle görünür.

3) Uygulama tasarımını sizce nasıldır?

Tasarım rahat. Yapılacak işlemler net belirlenmiş.

4) Uygulama erişimi ve kullanımı kolay mı?

İlk defa kullanmama rağmen kullanımı kolaydır. Açıklayıcı bilgi veriyor.

5) Uygulamanın güçlü yönleri nelerdir?

Flaş kart, eşleştirme, oyun ve değerlendirmeler çok iyi. Kelimeyi çok yönlü öğretiyor. Öğrenciye göre öğrenme stillerinin belirlenmesi çok faydalı.

6) Uygulamanın zayıf yönleri nelerdir?

Açıklayıcı bilgiler yeterli ama biraz daha net olabilirdi.

7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Avantajı öğrenme stillerinin belirlenmesi.
Dezavantajı kelime sayısı fazlaştırılmalı.

EK-1(d). Uzman değerlendirme cevapları

UZMAN DEĞERLENDİRME ANKETİ

Eğitmenin;

Ad Soyadı	: Uzman#4
Doğum Yılı	: -
Eğitmenlikte çalıştığınız süre (yıl)	: 18
İngilizce eğitmenliğinde çalıştığınız süre (yıl)	: 18
Kaç yıldır bilgisayar kullanıyorsunuz?	: 15

1) Uygulama amacına uygun mu?

Uygulama amacına uygun olarak hazırlanmış.

2) Uygulama kullanışlı ve anlaşılır mı?

Uygulama kullanışlı ve anlaşılır olmuştur.

3) Uygulama tasarımını sizce nasıldır?

Uygulama tasarımı anlaşılır ve nettir.

4) Uygulama erişimi ve kullanımı kolay mı?

Uygulama erişimi ve kullanımı kolay olmasına dikkat edilmiştir.

5) Uygulamanın güçlü yönleri nelerdir?

ILS testi ile öğrencinin öğrenme stilinin belirlenmesi uygulamanın güçlü yönüdür.

6) Uygulamanın zayıf yönleri nelerdir?

-

7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Kelimenin aynı uygulama içerisinde anlamı, eş anlamlısı, öğrenilmiş kelimelerin sisteme dahil edilebilmesi avantajlıdır. Belirlenen bir dezavantaj bulamadım. Uygulamadan memnun kaldım.

EK-1(e). Uzman değerlendirme cevapları

UZMAN DEĞERLENDİRME ANKETİ

Eğitmenin;

Ad Soyadı	: Uzman#5
Doğum Yılı	: 1980
Eğitmenlikte çalıştığınız süre (yıl)	: 14
İngilizce eğitmenliğinde çalıştığınız süre (yıl)	: 14
Kaç yıldır bilgisayar kullanıyorsunuz?	: 10

1) Uygulama amacına uygun mu?

Evet

2) Uygulama kullanışlı ve anlaşılır mı?

Evet

3) Uygulama tasarımını sizce nasıldır?

Anlaşılır

4) Uygulama erişimi ve kullanımı kolay mı?

Evet

5) Uygulamanın güçlü yönleri nelerdir?

Her öğrencinin öğrenme stiline göre hitap etmesi yönünden kullanışlı.

6) Uygulamanın zayıf yönleri nelerdir?

İnternet ve bilgisayara sahip olmayanlar için kayıp.

7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Avantajı kullanışlı olması ve her stile hitap etmesidir. Dezavantaj olarak bir şey göremedim.

EK-1(f). Uzman değerlendirme cevapları

UZMAN DEĞERLENDİRME ANKETİ

Eğitmenin;

Ad Soyadı	: Uzman#6
Doğum Yılı	: 1971
Eğitmenlikte çalıştığınız süre (yıl)	: 20
İngilizce öğretmenliğinde çalıştığınız süre (yıl)	: 20
Kaç yıldır bilgisayar kullanıyorsunuz?	: 20

1) Uygulama amacına uygun mu?

Uygundur. Amacına uygun olarak program hazırlanmış

2) Uygulama kullanışlı ve anlaşılır mı?

Uygulama kullanışlı ve anlaşılır olduğu görüldü.

3) Uygulama tasarımını sizce nasıldır?

Tasarım kullanımını kolay ve anlaşılır.

4) Uygulama erişimi ve kullanımını kolay mı?

Uygulama erişimi ve kullanımını kolay olduğu tespit edildi.

5) Uygulamanın güçlü yönleri nelerdir?

Değerlendirme olması programın etkisini ölçmekte yardımcı. Her öğrenciye göre farklı olması seviye belirlemede kolaylık sağlıyor. ILS testi ile öğrencinin öğrenme stilinin belirlenmesi çok yararlı.

6) Uygulamanın zayıf yönleri nelerdir?

Yok.

7) Mevcut uygulamalarla karşılaştırdığınızda avantaj ve dezavantajları nelerdir?

Uygulamanın her türlü öğrenmeye açık olduğu, öğrenme çeşitlerine göre hazırlandığı için öğrenci sıkılmadan tekrar yapabilir. Ne kadar öğrendiklerini ölçtükleri için basamak atlayabiliyor. Belirlenen bir dezavantaj bulamadım.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : AKDEMİR, Emre
 Uyuğu : TC
 Doğum tarihi ve yeri : 25/06/1982 Ankara
 Telefon : 0 (505) 727 98 70
 e-posta : emre.akdemir@gazi.edu.tr

Eğitim Derecesi	Okul/Program	Mezuniyet yılı
Yüksek Lisans	Gazi Üniversitesi / Bilgisayar Bilimleri	Devam Ediyor
Lisans	Gazi Üniversitesi / Bilgisayar Sistemleri Öğrt.	2005
Lise	Elmadağ Lisesi	1999

İş Deneyimi, Yıl	Çalıştığı Yer	Görev
2015-devam ediyor	MEB, Bilgi İşlem Dairesi	Öğretmen
2011-2015	Gazi Teknik ve Anadolu Meslek Lisesi	Öğretmen
2006-2011	Alaca Mesleki ve Teknik Eğitim Merkezi	Öğretmen
2005-2006	Elmadağ Mesleki ve Teknik Eğitim Merkezi	Öğretmen

Yabancı Dili

İngilizce

Hobiler

Bilgisayar Teknolojileri, Basketbol, Yüzme, Kitap okuma, Masa tenisi



GAZİ GELECEKTİR..