

**ONTOLOJİ TABANLI ETKİNLİK VE ÖĞRENME NESNESİ
PAYLAŞIM SİSTEMİ**

**ONTOLOGY BASED ACTIVITY AND LEARNING OBJECT
SHARING SYSTEM**

HASAN TÜRKSOY

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2007

ONTOLOJİ TABANLI ETKİNLİK VE ÖĞRENME NESNESİ PAYLAŞIM SİSTEMİ

Hasan Türksoy

ÖZ

Bu çalışma, etkinlik geliştirme sürecinde üretilen öğrenme nesnelerinin paylaşımı ve tekrar kullanımını sağlayacak bir araç geliştirmeyi amaçlamaktadır. Üretilen öğrenme nesnelerinin paylaşımı ve tekrar kullanımı sayesinde, aynı amaca yönelik olarak tekrar tekrar öğrenme nesnesi üretilmesi için harcanan zamanın önüne geçilmiş olunacaktır.

Öğrenme nesneleri havuzunda eğitmenin ihtiyaç duyduğu özelliklere en uygun öğrenme nesnesine ulaşabilmesi için ontolojiler ve anlamsal web teknolojilerinden faydalanılmıştır. Uygulama amacı yeni olmasa da, ihtiyaca en uygun bilgiye erişme kısmı anlamsal web projelerinin en güçlü yanlarıdır. Örneğin, aynı veri sıradan teknolojiler ile yönetilmek istendiğinde, yapılan bir sorgu sonucunda kullanıcı bağlamıyla ilgisi olmayan sonuçların fazlalığı dikkat çeker. Anlamsal web altyapısı kullanılarak kullanıcı, bu tür gereksiz sonuçlarla uğraşma zahmetinden kurtarılmaya çalışılır. Kullanılan ontolojiler sayesinde, kullanıcı bağlamına en uygun verilere ulaşılmaya çalışılır. Örneğin; kullanıcı “ağaç” sorguladığında, bilgisayar bilimleri konusu olan “ikili ağaç veri yapıları” ile ilgili sonuçlarla, biyoloji konusu olan ağaçlarla ilgili sonuçlar bir arada gelir. Halbuki kullanıcı, biyoloji alanı ile ilgileniyorsa, “ağaç, biyoloji” şeklinde bir sorgu yaptığında, daha çok ve öncelikli olarak biyoloji alanı ile ilgili sonuçlarla karşılaşmalıdır. Anlamsal web’in bu gücünden faydalanma isteği, basit bir veri yönetimi uygulaması olmasına rağmen, yapılan çalışmanın da öncelikli hedefidir.

Kullanılan anlamsal web teknolojilerinin yeniliği sebebiyle önce kavramlar açıklanmıştır. Bu bağlamda önce anlamsal web kavramı açıklanmıştır. Bu bölümde, anlamsal web’in ne olduğu, geleneksel web’den farkları ve anlamsal web’e neden gerek duyulduğu konuları irdelenmiştir. Daha sonra ontoloji kavramı, farklı ontoloji tanımları, ontoloji ile bilgi tabanının farkı, ontolojilere neden gereksinim duyulduğu konuları açıklanmıştır. Bir sonraki bölümde ontolojilerin eğitim alanında hangi amaçlarla kullanıldığı, ne tür faydalar sağlayabileceği ve eğitsel ontolojilerin yönetimi ve işlenmesi ile ilgili yapılmış bazı çalışmalarla ilgili bilgi verilmiştir. Anlamsal web teknolojileri ile ilgili bilgilerin aktarılmasından sonra;

öğrenme nesnelere tanımı ve anlamsal web mimarisinde nasıl kullanıldıkları konusu irdelenmiştir.

İkinci bölümde, anlamsal web uygulamalarının altyapısını oluşturan ilgili teknolojiler incelenmiştir. Bu kısımda RDF teknolojisi, yapısı anlatılmış, RDF çizimleri örneklerle açıklanmıştır. Daha sonra OWL (Web Ontology Language) mimarisi anlatılmış. OWL mimarisine neden gereksinim duyulduğu örneklerle açıklanmış, bazı OWL örnekleri ile teknolojinin detayları aktarılmaya çalışılmıştır. Bu bölümün son kısmında ise; yapılmış anlamsal web uygulamaları hakkında bilgiler verilmiştir. Verilen örnekler daha çok eğitsel alanla ilgili çalışmalardır. İlk önce ontoloji geliştirme çalışmaları incelenmiş, daha sonra ise geliştirilen ontolojileri işleyerek anlamsal web'i eğitim alanında uygulamaya çalışan bir takım eğitsel uygulamalar incelenmiştir.

Üçüncü bölümde, geliştirilen uygulama ile ilgili yöntem ve teknolojiler hakkında bilgiler verilmiştir. Ontolojinin oluşturulması, bunun için kullanılan araçlar ve kullanım bilgileri aktarılmıştır. Bu aşamada, Protégé ontoloji geliştirme aracı tanıtılmış ve kullanımı, ekran görüntüleri ile desteklenerek açıklanmaya çalışılmıştır. Ayrıca, OWL sınıfları ve özelliklerinin oluşturulması işlemleri adım adım açıklanmıştır. Daha sonra, oluşturulan ontolojiyi işleyebilmek için gerekli araçlar ve bu araçların nasıl kullanıldığı hakkında bilgiler verilmiştir. Bu araçlardan Jena anlamsal web geliştirme aracı ve ontolojiden Java sınıfları üretmekte kullanılan Jastor aracı kullanım bilgileri örneklendirilerek açıklanmıştır. Bu aşamada, anlamsal web uygulaması geliştirirken karşılaşılan beklenmedik durumlar ve sorunlar, araçlar arasındaki uyumsuzluklar ve bulunan çözüm yolları da aktarılmıştır.

Son bölümde ise, geliştirilen uygulamanın kullanım durumları, özellikleri ve ekran görüntüleri verilmiştir.

Anahtar Kelimeler: Anlamsal web, ontoloji, etkinlik ve öğrenme nesnesi paylaşımı

Danışman: Prof. Dr. Petek AŞKAR, Hacettepe Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

ONTOLOGY BASED ACTIVITY AND LEARNING OBJECT SHARING SYSTEM

Hasan Türksoy

ABSTRACT

This study aims to develop a tool to share and reuse of learning objects created during activity development process. By sharing and reusing of learning objects, new learning objects' production time will decrease.

The study uses ontologies and some other semantic web technologies to search the learning object pool for the most appropriate learning objects that instructors required. Although the aim of the project is not new, reaching to the most appropriate data for the needs is the most powerful part of the semantic web projects. For instance, if we'd like to manage the same data by ordinary methods, result includes lots of data which are not related with the user's real context. Usage of semantic web technologies aims to protect the user from trying with such unrelated data. Reaching to the most convenient data is aimed by the usage of ontologies. For example; when the user queried "tree", results may include "binary tree data structures" of computer science domain together with the "tree" of biology domain. However, if the user needs results about biology domain, it should be enough to make a query like "tree, biology" to get them primarily. Although it is a simple data management application, benefiting from this power of semantic web technologies is the preferred objective of this study.

Some concepts explained because of the usage of new technologies. In this context, semantic web concept is explained first. Investigated some topics like what the semantic web is, its differences from traditional web and requirement for the semantic web. Later, ontology, different ontology definitions, differences of ontology and knowledge base and requirement for ontologies explained. At the following section, given some information about the usages of ontologies in educational field, advantages those can provide, and studies about management and processing of ontologies. After providing knowledge about semantic web, learning object definitions and their usage at the semantic web architecture were elaborated.

Second section examines some technologies about semantic web infrastructure. Firstly, the structure of RDF technology is introduced and given some RDF graph samples. After that, OWL (Web Ontology Language) structure explained. Need for OWL architecture and the details of OWL technology are explained by some examples. In the last part of this section, information about some semantic web studies, especially in the educational field, were given. These studies include ontology development studies, ontology management studies which tries to apply semantic web technology into some educational cases, etc...

In the third section, given information about the methods and technologies used during our application's development process. Described the creation of ontology and concepts about OWL. In this context, referred to Protégé ontology development tool and introduced its environment with the help of some sample screenshots. Also, creating OWL classes and properties is explained step by step. Later, introduced some required tools and explained their usage to process this created ontology. These tools include Jena semantic web framework, Jastor to generate Java classes from ontologies. While introducing some other alternative tools, quoted some problems along with disharmonies between semantic web development tools and unexpected issues faced while developing our semantic web application. Solutions to overcome those issues also provided.

In the last section, use cases, properties and some screen shots about developed application are provided.

Keywords: Semantic web, ontology, sharing of activity and learning object

Advisor: Prof. Dr. Petek AŞKAR, Hacettepe University, Department of Computer Education and Instructional Technology

TEŐEKKÜR

Sayın Prof. Dr. Petek AŐkar'a, alıŐmanın sonuca ulaŐtırılmasında ve karŐılaŐılan problemlerin giderilmesinde destek, ilgi ve bilgisini esirgemeyerek her tŸrlŸ yardımı yaptıđı ve en zor Őartlarda dahi gŸvenini ifade ederek, salt tez danıŐmanlıđının Ÿtesinde bir yŸn gŸsterici olduđu iin itenlikle teŐekkŸr ederim.

Sevgili eŐime, alıŐmamda her zaman yanımda olarak beni desteklediđi, bana gŸvendiđi ve iyi bir alıŐma ortamı oluŐturabilmek iin her tŸrlŸ fedakarlıđı gŸsterdiđi iin sonsuz teŐekkŸr ederim.

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZ	I
ABSTRACT.....	III
TEŞEKKÜR.....	V
İÇİNDEKİLER DİZİNİ	VI
ŞEKİLLER DİZİNİ	VIII
ÖRNEKLER DİZİNİ.....	IX
1. GİRİŞ	1
1.1. Anlamsal Web.....	3
1.2. Ontoloji.....	4
1.3. Eğitimde Ontolojiler	5
1.4. Öğrenme Nesneleri	7
1.5. Çalışmanın Amacı	8
2. İLGİLİ TEKNOLOJİLER VE ARAŞTIRMALAR	9
2.1 RDF Yapısı	9
2.2 OWL (Web Ontology Language) Yapısı	12
2.3 Anlamsal Web Uygulamaları	14
3. YÖNTEM	18
3.1 Ontolojinin Oluşturulması	18
3.2 Anlamsal Web Uygulaması Mimarisi	28
3.2.1 Ontolojinin geliştirilmesi	29
3.2.2 Ontolojinin içeriğini yönetmek için yardımcı Java sınıflarının üretilmesi... 37	
3.2.3 Oluşturulan ontoloji yönetim sınıfları kullanılarak ontolojinin sorgulanması ve yeni nesne örneklerinin kaydedilmesi	38
3.3 Geliştirilen Aracın Kullanımı.....	40
3.3.1. Etkinlik Giriş	41
3.3.2. Etkinlik Sorgulama	42

3.3.3. Öğrenme Nesnesi İşlemleri.....	43
4. TARTIŞMA VE YORUM.....	45
KAYNAKLAR.....	47
EKLER DİZİNİ.....	52
ÖZGEÇMİŞ	56

ŞEKİLLER DİZİNİ

Sayfa

ŞEKİL 2.1.1. “ERIC MILLER” İSMİNDE BİR KİŞİYİ BETİMLEYEN ÖRNEK RDF ÇİZGESİ.....	10
ŞEKİL 3.1.2. ETKİNLİK ONTOLOJİSİNİN PROTÉGÉ ARACI İLE GELİŞTİRİLMESİ.....	22
ŞEKİL 3.1.3. NESNE ÖRNEKLERİ ARASINDA İLİŞKİ TANIMLANMASI	24
ŞEKİL 3.1.4. TERS İLİŞKİ ÖRNEĞİ: ALTKONU, USTKONU ADINDA BİR TERS İLİŞKİYE SAHİP	25
ŞEKİL 3.1.5. FONKSİYONEL İLİŞKİ ÖRNEĞİ: “GELİŞTİREN”	25
ŞEKİL 3.1.6. GEÇİŞLİ İLİŞKİ ÖRNEĞİ: “ALTKONU”	26
ŞEKİL 3.1.7. SİMETRİK İLİŞKİ ÖRNEĞİ: “İLİSKİLİOGRENMEALANI”.....	26
ŞEKİL 3.2.1.1. ETKİNLİK GİRİŞİ MENÜSÜ.....	41
ŞEKİL 3.2.1.2. ETKİNLİK GİRİŞ EKRANI.....	41
ŞEKİL 3.2.2.1 ETKİNLİK SORGULAMA MENÜ ADIMI.....	42
ŞEKİL 3.2.2.2. ETKİNLİK SORGULAMA EKRANI.....	43
ŞEKİL 3.2.3.1. ÖĞRENME NESNESİ GİRİŞ MENÜSÜ.....	44
ŞEKİL 3.2.3.2. ÖĞRENME NESNESİ GİRİŞ EKRANI.....	44

ÖRNEKLER DİZİNİ

Sayfa

ÖRNEK 2.1.1. "ERİC MİLLER"İ BETİMLEYEN RDF/XML YAPISI.	11
ÖRNEK 3.1.1. ETKİNLİK ONTOLOJİSİNDEN BİR BÖLÜM.....	22
ÖRNEK 3.2.3.1. JENA API'LERİ KULLANARAK BİR ONTOLOJİ MODELİ OLUŞTURMA.....	38
ÖRNEK 3.2.3.2. JASTOR YARDIMCI SINIFLARI KULLANILARAK YENİ ONTOLOJİ SINIFI OLUŞTURMA.....	39

1. GİRİŞ

Bilgi teknolojileri, her alanı olduğu gibi eğitim alanını da ciddi boyutta etkilemiştir. Eğitimde kullanılan teknolojiler eğitim teknolojileri olarak özelleşme boyutuna ulaşırken, bu teknolojiler e-öğrenme vizyonu ile öğrenme sürecini farklılaştırmışlardır. Genel kanı olarak e-öğrenme dendiği zaman anlaşılan şey; web teknolojilerinin öğrenme/öğretme sürecinde daha aktif kullanımınıdır. Bu anlamda e-öğrenme, ders/etkinlik yönetimi, web tabanlı öğrenme veya video konferanslar gibi ağ kullanımı ve pedagojik içerik tasarımı ile ilgili konular üzerinde yoğunlaşmış olarak kullanılmaktadır.

Günümüzde yüksek oranda kabul görmüş olan yapılandırmacı eğitim yaklaşımı ve çoklu zeka kuramına uygun etkinliklerin geliştirilmesinde öğretmenler pek çok eğitsel içerik ya da kaynağı uygulamalarında kullanmaktadırlar. Bir etkinlik geliştirilirken, o etkinliği amacına uygun olacak şekilde destekleyecek öğrenme nesnelерinin edinilmesi, öğretmenler açısından çok zaman tüketen ve yorucu bir aşama olmaktadır. Bu süreci kolaylaştıracak en önemli etken ise, istenen özellikleri sağlayan hazır, tekrar kullanılabilir ve farklı ortamlarda kullanılacak öğrenme nesnelерine erişebilmektir. Fakat, gerek yerel ortamlarda (kişisel veya ortak çalışmalarla üretilen) gerekse web ortamında (çevrimiçi dersler, ders notları vb...) pek çok kaynak üretilmiş ve kullanılıyor olmasına rağmen bu kadar çok kaynak içerisinden öğretmenlerin ihtiyaç duyduğu niteliklere sahip kaynaklara erişim konusunda başlıca üç önemli problemle karşılaşmakta;

1. Kaynaklara erişimi sağlayacak sistemlerin eksikliği;

Bu sorun günümüzde, birtakım içerik yönetim sistemlerinin kullanılmasını gerektirmiştir. Bu tür sistemlere sahip olmayanlar için ve internet ortamındaki kaynaklara erişmek isteyenler için ise çeşitli kütüphanelerin taranması, web aramaları neticesinde ulaşılan pek çok sonuç içerisinden uygun olanların seçilmesi gibi seçenekler kalmaktadır.

2. Erişilen kaynakların aranan niteliklere sahip olmama oranının yüksekliği

Bu tür aramalar, zor da olsa, üretilen kaynakların tekrar kullanımı ve paylaşımıyla ilgili ihtiyacı genel anlamda karşılamaktadırlar. Fakat, bu sistemlerin kaynak içeriklerini, kaynaklar arası ilişkileri ya da çıkarımları

yapmadan doğrudan yaptıkları sorgulamalar, beklenen işletim etkinliğinin gerisinde sonuçlar elde edilmesine neden olmaktadır. Bunun en önemli nedeni; günümüz standartlarında kaynakların, eğitsel anlam ifade edecek şekilde, yani Anlamsal Web için tasarlanmamış olmasıdır. Bu noktada gereksinim duyulan anlamsal web teknolojileri, bahsedilen eksiklikleri giderebilmek için sunulmuştur.

3. Erişilen ya da üretilen kaynakların paylaşımı

Bir eğitmen, gereksinim duyduğu kaynaklara yukarıda bahsedilen zahmetli süreçler neticesinde ulaşmış olsa da, aslında temelde problem tamamen çözülmüş olmamaktadır. Zira, aynı konuda etkinlik çalışması yapmak isteyen başka bir eğitmenin de, aynı süreçlerden geçmesi gerekecektir. Çalışmamız sonucu geliştirilen “Etkinlik Yönetim Sistemi” ise bu eksikliğı gidermeyi amaçlamaktadır.

1.1. Anlamsal Web

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 (Herman, 2007b). Anlamsal web fikrini ortaya atan ilk makale Scientific American dergisinde 2001 yılında yayınlanmıştır (Berners-Lee et al., 2001). Bu makale, anlamsal web'i, şu anda kullanılan web'den ayrı bir web oluşumu değil, onun bir uzantısı olarak tanımlar. Anlamsal web'in halihazırda kullanılan web ortamını geliştirmesinin ilk aşamasının, web ortamındaki veri ve dökümanları makinelerin de işleyip "anlayabilmesi" olacağı belirtilmiştir.

Web'in evrenselliği ve kaynaklar arasındaki bağlantıların asıl gücü; "herşeyin herşeye bağlantı içerebileceği" üzerine kuruludur. Bunun sonucunda web teknolojisi, olgunlaşmamış taslak döküman ile işlenmiş ve son hali verilmiş bir dökümanı, ticari bilgi ile akademik bilgiyi, kültür ve dil farklılıklarını ayırt edememektedir. İşte, anlamsal web'in şu andaki web ortamından ayrılma noktası, bu ayrımları yapabilme yeteneğidir. Bu ayırım, web'deki verinin anlamsal yapılarının kullanılması sonucu makinelerin işleyebileceği verinin oluşturulmasıyla sağlanacaktır. Neticede daha karmaşık ve daha iyi sonuç veren sorgular yapılabilecek, insanlar kendi durumlarına ve gereksinimlerine en uygun veriye ulaşabilecektir. Veri ve bilginin bu işletimi ise, birtakım özelleşmiş otomatize ajanlar tarafından sağlanacaktır.

Günümüzde, sıradan yöntemler uygulanarak geliştirilen uygulamalarda, yapılan bir sorgu sonucunda kullanıcı bağlamıyla ilgisi olmayan sonuçların fazlalığı dikkat çeker. Anlamsal web altyapısı kullanılarak kullanıcı, bu tür gereksiz sonuçlarla uğraşma zahmetinden kurtarılmaya çalışılır. Kullanılan ontolojiler sayesinde, kullanıcı bağlamına en uygun verilere ulaşılmaya çalışılır. Örneğin; kullanıcı "ağaç" sorguladığında, bilgisayar bilimleri konusu olan "ikili ağaç veri yapıları" ile ilgili sonuçlarla, biyoloji konusu olan ağaçlarla ilgili sonuçlar bir arada gelir. Halbuki kullanıcı, biyoloji alanı ile ilgileniyorsa, "ağaç, biyoloji" şeklinde bir sorgu yaptığında, daha çok ve öncelikli olarak biyoloji alanı ile ilgili sonuçlarla karşılaşmalıdır.

Şu anda kullanılan web teknolojileri genellikle döküman alışverişi esasına dayanmakta, fakat, anlamsal web, çeşitli dökümanlardan elde edilen verinin bütünleştirilmesi ve birleştirilmesi için gerekli olan ortak yapılar üzerinde durmaktadır. Bu yapıların hem yukarıda bahsedildiği gibi verinin anlamsal yapısını tutabilme hem de çeşitli dökümanlardan elde edilen verinin bütünleştirilmesi ve birleştirilmesi için gerekli ortak yapıyı sağlama yeteneği olmalıdır.

1.2. Ontoloji

Bu çalışmayı ve konusunu tam olarak anlayabilmek için öncelikle ontolojinin ne olduğunun bilinmesinde fayda vardır. Ontolojilerle ilgili, çoğu kabaca aynı olsa da, pek çok tanım bulunabilir.

Ullrich (2004)'e göre ontoloji, insanlar veya sistemler arasındaki iletişimin temel aldığı bir bilgi alanının, ilgili sistemler tarafından tanınan ortak anlamını ifade eder.

Gruber (2007) ise, ontolojinin bilgi sistemleri alanında kullanımını gözönüne alarak daha yalın bir tanım kullanmış ve ontolojiyi; “kavramların detaylandırılması” olarak tanımlamıştır. Daha açık bir ifadeyle, ontoloji; bir ajan yada ajan topluluğunun sahip olabileceği kavramların ve ilişkilerin tanımıdır.

Nelerin ontoloji olarak kabul edilebileceği konusunda pek çok tartışmalar olmakla birlikte, tüm yaklaşımlarda temel olarak bulunan iki unsur vardır (Ushold and Gruninger, 2004):

- Bir bilgi alanından seçilen şeyleri karşılayan terimler kümesi
- Bu terimlerin anlamlarını, belirli bir mantık çerçevesinde, detaylandırılması.

Özellikle bilgi tabanı (knowledge base) ile ontoloji kavramları sıklıkla karıştırılan kavramlardır. Ontoloji ve bilgi tabanlarının ortak özellikleri; ikisinin de bilgiyi betimlemesidir. Ancak, bilgi tabanı, ontolojiden farklı olarak, kavramlar ve bunların anlamlarından oluşan terminolojilere ek olarak bu kavramların örneklerini de içerir. Yani, eğer bilgi tabanından kavramların örnekleri çıkarılırsa, ontoloji elde edilmiş olur.

Genel anlamda ontolojiler, kavramları betimleyen terimler kümesi, bunların anlamlarının detaylı tanımları ve bu kavramlar arasında varolan ilişkilerin

birleşiminden oluşur. Bu kavram ve ilişkilerin yönetimini sağlayabilecek birtakım kurallar ve kısıtlarda kullanılarak, ilgili terimle ilgili olası yorum sayısı sınırlanabilir. Bunun sonucunda terimin anlamsal yapısı kısıtlanmış olacağı için, istenen bilgiye ulaşım daha kolaylaşacaktır.

Bu özelliği ile ontolojiler, gerçekleştirmelerinden bağımsız olarak farklı sistemlerin bilgi düzeyinde entegrasyonunu ve dolayısıyla bilgi paylaşımını sağlayacak altyapıyı sağlarlar. Ontolojiler, kavramları ve bunların anlamlarını öz ve belirsizlik içermeyen bilgi formunda sunabilmelidir. Ancak böyle bir bilgi, farklı ajanlar (insan ve/veya makineler) arasında paylaşılıp tekrar kullanılabilir. Ontolojiler temelde iki şekilde kullanılmaktadır. Birincisi; bir alanla ilgili kavramların sunulması. İkincisi ise; belirli alanlarla/sistemlerle ilgili geliştirilmiş olan ontolojiler arasındaki ilişkileri tanımlayarak farklı ontolojileri birbirine bağlayan ontolojilerdir.

Özellikle ikinci türden ontoloji mimarileri, farklı sistemlerdeki bilgileri eşleştirmeyi/birleştirmeyi amaçlayan uygulamalar için çok kritiktirler. Bu tür uygulamalarda, formatında anlaşma sağlanmış XML DTD'leri yada XML şemalarının kullanımı, ontolojilerdeki mantıksal yapıları sağlayamadıkları için, yapılacak işin gereken güvenilirlikte yapılmasını engellemektedir.

1.3. Eğitimde Ontolojiler

Geleneksel etkinlik tasarımı, bilgi alanı ve öğrencilerin yerine getirmeleri beklenen becerilerle ilintili olarak oluşturulan beklentileri veya öğrenme hedeflerini temel almaktadır. Halbuki, bilgi alanı ve beceriler arasındaki ilişkilerin doğrusal ve hiyerarşik oluşu, e-öğrenme ortamlarının kişiselleştirilmesini zorlaştırmaktadır. Ontolojiler, bilgiyi, iyi tanımlanmış anlamsal yapılarla birlikte açık ve anlamlı bir şekilde sunmamızı sağlayan bilgi betimleme araçlarıdır (Kızılkaya et al., 2007).

Uygulama açısından bakıldığında, ontolojiler uzun zaman bilgi bankası görevi gören teknik birer araç olarak algılandı. Ontolojilerin bilişsel araçlar olarak anlamlandırıldığı alanların başında eğitim alanı gelmektedir. Bunun en önemli sebepleri; yapılandırmacı yaklaşımın yayılışı ve öğretim amacıyla kullanılan kavram haritaları, zihin haritaları gibi bilgi teknolojilerinin geniş kullanımınıdır (Dicheva et al., 2005).

Eđitimde, paylaşılan öđretimsel tanımların üretilmesi, hem öđrenciler hem de öđretmenler için faydalar içermektedir. Öđrenme nesneleriyle ilgili arama sonuçlarının daha tutarlı olması daha iyi düzeyde bir paylaşım sağlayacağı gibi aynı konu ile ilgili birden fazla gereksiz çalışmayı azaltacaktır. Sonuçta da etkinlikler daha hızlı geliştirilecek ve öđretimsel olarak uygun materyallere erişim sonucunda da birtakım bilgi eksiklikleri daha etkin bir şekilde giderilebilecektir (Ullrich, 2004). Öđrenme nesnelерinin daha etkin paylaşımını amaçlayan diđer bir çalışmada (Mohammed and Mohan, 2005) ise öđrenme nesnelерinin, ilişkilendirilen ontolojileri kullanarak, kendi bağlam ve karakteristik özellikleri hakkında yorum yapabilme yeteneđine sahip olması sağlanmıştır. Bunun sonucunda, yapılan öđrenme nesnesi istekleri sonucunda, sadece ilgili özelliklere sahip olan ve istenen bağlama uyan öđrenme nesneleri dönecek ve sorgu sonucu dönen kayıt sayısı etkin bir şekilde sınırlanmış olacaktır.

Örneđin, eđitsel içerik için tasarlanmış bir ontoloji, bir materyalin öđretimsel fonksiyonunu tanımlamada kullanılabileceđi gibi, ek olarak, etkinliđin temel aldığı eđitsel kuramları belli oranda tanımlamada da kullanılabilir. Böyle bir ontolojinin tanımlanabilmesi için ilgili öđretimsel fonksiyonun ne olduđu, özellikleri vs.'de tanımlanmak zorundadır. Aynı şekilde, etkinliđin temel aldığı kuramın belirlenebilmesi için de, eđitim kuramlarının ve özelliklerinin tanımlanması gerekir. Bu sayede, bu ontolojiyi paylaşan her sistem, geliştirdiđi etkinliđin dayandırıldığı kuramın özellikleri konusunda hemfikir olacaktır. Bu ontolojiyi paylaşan sistemler genişledikçe, eđitim kuramları konusunda ortak payda da o kadar büyüyecek ve kuram ile ilgili standardartlaşma sağlanabilecektir.

Ontolojiler, birtakım çıkarım araçları ve genel bilgi sunum teknikleriyle kullanıldığında, kütüphane, bilgi paylaşımı ve tekrar kullanımını sağlayan yazılım sistemlerinin geliştirilmesi ile ilgili çalışmalarda da kullanılabilir. Örneđin O4E (Ontologies for Education) (Dicheva et al., 2005), sayısı giderek artan ontolojileri tek bir portal altında toplayarak hem geliştirilmiş olan ontolojilerle ilgili arama, paylaşım ortamı sunmak hem de bu paylaşımın bir sonucu olarak yapılan ontolojilerin tekrar tekrar baştan oluşturulması yerine varolan ontolojilerin geliştirilmesini sağlamak amacıyla geliştirilmiş bir anlamsal web portalı çalışmasıdır.

Ontolojilerin en aktif kullanıldığı alanlardan birisi de uyarlanabilir e-öğrenme ortamlarının oluşturulmasıdır. Holohan ve diğerleri, çalışmalarında (Holohan et al., 2005), tanımlanan ontolojileri kullanarak uyarlanabilir (farklı öğrenme/öğretme stratejilerine göre, öğrenci profiline göre) öğrenme nesnelерinin yarı otomatik olarak üretilmesini sağlamışlardır. Bu amaçla yapılan diğer bir çalışma kapsamında geliştirilen AdaptWeb (de Freitas et al., 2002) sistemi, seçilen öğrenci profiline ve programlara göre uyarlanabilen web tabanlı ders içerikleri hazırlayabilmektedir. Materyal içeriği; karmaşıklık, sıradüzeni, örnek ve destek materyallerinin kullanılma durumlarına göre değiştirilerek üretilebilmektedir.

1.4. Öğrenme Nesneleri

Öğrenme nesneleri ile ilgili pek çok önemli tanım (Barrit et al.,1999; Richards, 2002; Wiley, 2000) arasında en genel olanı (Duval, 2002): Bir öğrenme nesnesi, teknoloji destekli öğrenim süreci boyunca kullanılabilir, referans verilebilecek yada tekrar kullanılabilir, dijital yada dijital olmayan her şeydir. Bu tanım aynı zamanda LOM (Learning Object Metadata) için IEEE LSTC (Learning Technology Standards Committee) standardında da kullanılmıştır. Bu fazlasıyla geniş ve sınırları belirsiz tanıma ek olarak, öğrenme nesneleri farklı bağlamlardaki farklı nesnelere olan anlamsal ilişkileri ifade etmekte yoksundur. Bu durum ise, IMS öğrenim tasarımı tanımlamasının (IMS-LD-IM, 2003) geliştirilmesini gerektirmiştir.

Bir sonraki aşama geliştirilen öğrenme nesneleri yada öğrenim tasarımlarının farklı bağlamlarda yeniden kullanılabilmesi ve paylaşılabilmesidir. Bu nesnelere paylaşılabilmesi için ilk akla gelen altyapı XML'dir. Ancak XML, bu nesnelere anlamsal yapılarını içermekten yoksundur. Bu noktada, öğrenme nesneleri ve öğrenim tasarımlarının paylaşımı ve tekrar kullanımı için ontolojiler ve anlamsal web teknolojileri tam bir çözüm sunmaktadırlar.

Bir bilgisayar destekli etkinlik tasarımında, eklenen öğrenme nesneleri (resim, metin, animasyon, vb...) ve öğrenmeyi destekleyici birtakım servisler (mesajlaşma, sohbet ve tartışma ortamları, çoktan seçmeli testler, vb...) kullanılarak, içeriğin hangi sırada, nasıl sunulacağı, öğrencilere nasıl aktarılacağı gibi durumlar düzenlenebilir.

Öğrenme nesnelerinin farklı bağlamlarda tekrar kullanımının sağlanabilmesi için bir havuzda toplanmış olduğunu varsayarsak, bir eğitmenin pek çok öğrenme nesnesi arasından kendi ihtiyaçlarına en uygun olanı bulabilmesi için gerekli arama yeteneklerinin sağlanmış olması gerekmektedir. Bunun için, oluşturduğumuz ontolojide, öğrenme nesnelerinin diğer nesnelere ya da kavramlarla ilişkilerini belirlemek için gerekli metaveriler ilişkilendirilmiştir. Bir sonraki bölümde daha detaylı açıklanacağı gibi, çalışmamızın asıl amacı; bu anlamsal mimariyi kullanarak, eğitmenlere, kendi bağlamları için en uygun öğrenme nesnesine ulaşabilmek için gerekli yapıyı sağlamaktır.

1.5. Çalışmanın Amacı

Bu çalışma, eğitmenlerin etkinlik geliştirme sürecinde yüzleştikleri materyal sağlama güçlüğüne hafifletebilmek için bir araç geliştirmeyi amaçlamaktadır. Geliştirilecek araç sayesinde eğitmenler geliştirdikleri etkinliklerle ilgili öğrenme nesnelerini paylaşımına açacaklar, başka bir eğitmen ise, etkinlik geliştirme sürecinde aracın sağladığı arayüzleri kullanarak konu ile ilgili mevcut kaynaklar arasında arama yapabilecektir. Bu sayede aynı amaca yönelik geliştirilmiş olan bir öğrenme nesnesinin tekrar tekrar farklı eğitmenler tarafından üretilmesinin ve bunun için harcanan zamanın önüne geçilmiş olacaktır.

Geliştirilen öğrenme nesnelerinin paylaşılabilmesini sağlamanın yanında, bu paylaşımın etkinliğini artırabilmek için, etkinlik arama mantığının basit sorgulardan öte anlamsal web teknolojileri kullanılarak kullanıcının isteğine en yakın sonuçları sağlayabilecek yeteneğe sahip olması gereklidir. Bu sayede kaynak arayan bir kullanıcı, amaç ve bağlamına daha uygun kayıtlar arasından seçim yapabilecektir.

Uygulama sonunda, etkinlik geliştirme süresinin kısalması, bilgi paylaşımının artması ve paylaşılan öğrenme nesnelerinin farklı bağlamlarda kullanılmasıyla olgunlaşması sağlanmış olacaktır.

2. İLGİLİ TEKNOLOJİLER VE ARAŞTIRMALAR

Veriyi sunabilmek ve üzerinde bir takım çıkarımsal işlemler yapabilmek, bu veriyi farklı sistemlerin paylaşabilmesi ve sözdizim ve anlamsal olarak anlayıp işleyebilmesi için, hem veri hem de bu veri ile ilgili birtakım kuralların girilebileceği ortak bir dil gereklidir. Bu ortak dil, günümüz teknolojilerinden XML (eXtensible Markup Language) ve RDF (Resource Description Framework) altyapısı kullanılarak oluşturulmaktadır.

XML sözdizimi ile geliştiriciler, uygulamaya özel etiketler (tag) kullanarak veriyi belli bir yapıda sunabilirler. Bu şekilde etiketlenmiş ve belli bir formatta sunulan veriyi uygulamalar pek çok şekilde ve değişik amaçlarla kullanabilirler. Ancak, bir uygulamanın bir XML'i kullanabilmesi için, ilgili XML'i geliştiren kişinin, etiketleri hangi amaçla kullanılmak üzere oluşturduğunun bilinmesi gerekir. Başka bir deyişle, XML teknolojisi bize verinin yapısal olarak sunulmasını sağlar ancak anlamsal olarak ne ifade ettiği ile ilgili bilgi içermez.

Bu anlamsal ifade eksikliği WWW konsorsiyum'unun Şubat 2004'te sunduğu RDF teknolojisi ile giderilmiştir (Manola, 2004).

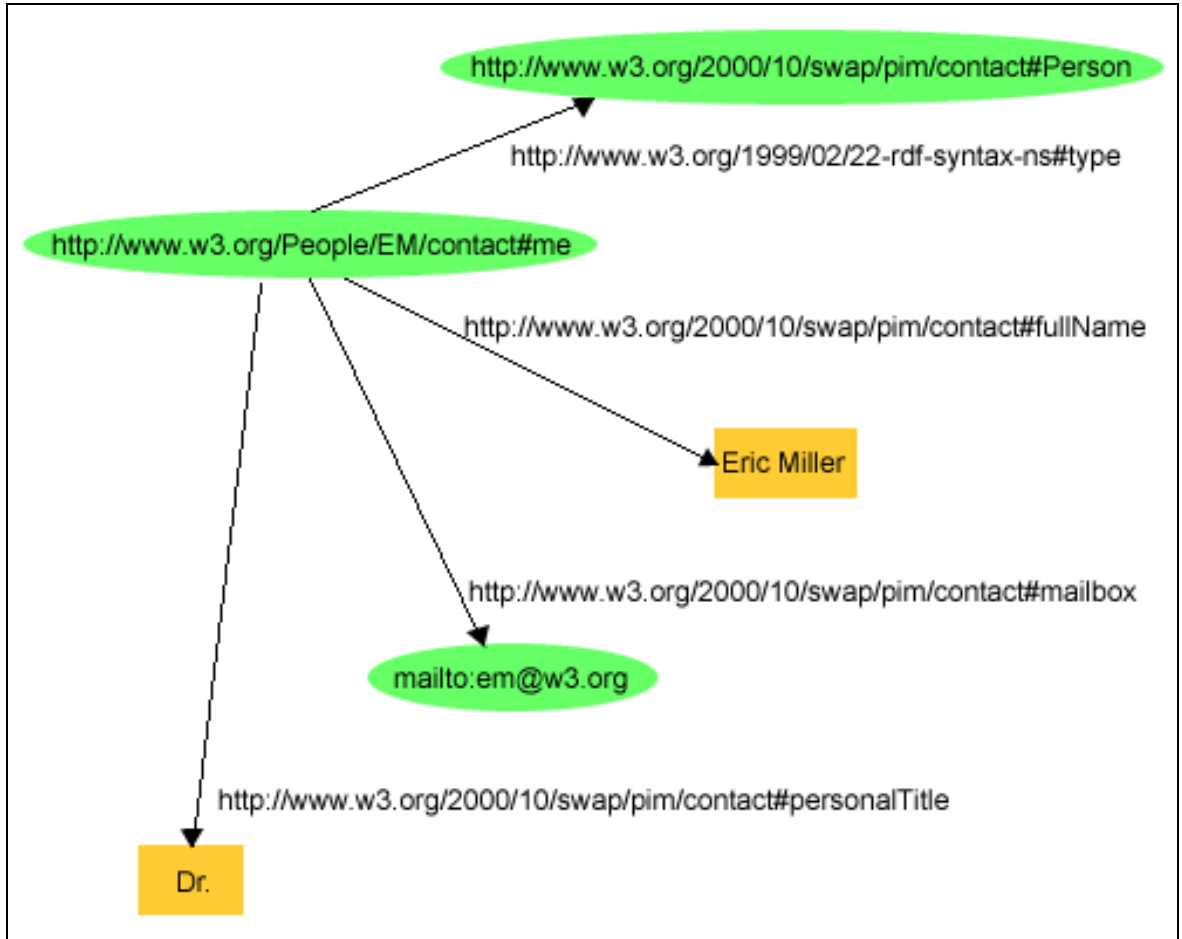
OWL (Web Ontology Language) ise, bilgi yönetimini, yazılım ajanlarını ve ileri düzeyde web aramalarını destekleyen, ontoloji denem yapıların yayımlanması ve paylaşımı için kullanılmaktadır. Her iki veri alış-verişi ve anlamlandırma formatı da zaten var olan XML (eXtensible Markup Language) teknolojisi temeline dayanmaktadır.

2.1 RDF Yapısı

RDF, bilgiyi sunmak ve web'de paylaşımını sağlamak için kullanılır. RDF formatı XML teknolojisi temeline dayanır. RDF yapısında anlamlar RDF üçlüleri kullanılarak ifade edilir. Herbir üçlü bir özne, eylem ve nesne (subject-verb-object) birleşiminden oluşur. Daha öncede belirtildiği gibi bu üçlüler XML etiketleri ve yapısı kullanılarak oluşturulabilir. Bir RDF dökümanında, birtakım şeylerin (özneler=insanlar, web sayfaları,vb..) bazı özelliklerinin (property="-in kızkardeşi", "-in yazarı", vb..) sahip olduğu bazı değerler (başka bir insan, web sayfası, vb..) ifade edilebilir. Aslında bu yapı, bilgisayarlarda işlem gören verilerin büyük çoğunluğunu ifade etmenin doğal yoludur. Özne ve nesnelerin herbiri, tıpkı bir web

sayfasında kullanılan bağlantılar gibi, URI (Universal Resource Identifier)'lar ile tanımlanır. Bu bağlantılara URL (Uniform Resource Locators) denir ve bunlar aslında URI'ların en yaygın halidir. Eylemler de URI'lar aracılığıyla tanımlanır. Bu sayede herkes web üzerinde bir kavram ya da fiili bir URI ile tanımlayarak oluşturabilir (Berners-Lee et al., 2001).

Şekil 2.1.1.'de, "Eric Miller" isminde bir kişiyi betimleyen örnek bir RDF çizgesi görülmektedir. Çizge, W3C RDF Primer (<http://www.w3.org/TR/rdf-primer/>) sitesinden alınmıştır.



Şekil 2.1.1. "Eric Miller" isminde bir kişiyi betimleyen örnek RDF çizgesi.

Bu RDF çizgesi, şu şekilde ifade edilebilir; `http://www.w3.org/People/EM/contact#me` ile tanımlanmış ve ismi "Eric Miller" olan

bir kiři var. Bu kiřinin e-posta adresi “em@w3.org” ve ünvanı “Dr.”. Yukarıdaki gibi düğümler ve bunlar arasındaki baęlantılardan oluřan bir RDF çizgesi oluřturulurken, herbir düğüümü ve düğümler arasındaki iliřkileri belirleyen baęlantıları tanımlayan URI referansları kullanılır. Sonuç olarak Őekil 2.1.1.’deki RDF çizgesi, URI referansları kullanarak (Manola, 2004);

- bireyleri, (örę. “Eric Miller”, <http://www.w3.org/People/EM/contact#me> ile tanımlandı)
- nesnelerin türlerini, (örę. Kiři, <http://www.w3.org/2000/10/swap/pim/contact#Person> ile tanımlandı)
- nesnelerin özelliklerini, (örę. e-posta, <http://www.w3.org/2000/10/swap/pim/contact#mailbox> ile tanımlandı)
- bu özelliklerin deęerlerini, (örę. e-posta deęeri olarak “mailto:em@w3.org”) (RDF aynı zamanda, “Eric Miller” gibi karakter dizilerini ve integer, date gibi dięer veri türlerini de destekler)

tanımlamakta kullanılır.

RDF, bu çizgelerin kaydedilmesi ve deęiř tokuř’u için XML tabanlı bir sözdizimi saęlar. Ařaęıda görölen Örnek 2.1.1., Őekil 2.1.1.’deki çizgenin bir kısmına karřılık gelen XML dizilimidir (RDF/XML) (Manola, 2004):

Örnek 2.1.1. “Eric Miller”ı betimleyen RDF/XML yapısı.

```
<?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>
```

2.2 OWL (Web Ontology Language) Yapısı

Anlamsal web’de sınıflar ve web dökümanlarında kullanılan özellikleri biçimsel olarak tanımlamak için gerekli olan bir sonraki öge; OWL (Web Ontology Language)’dir. Bilgisayarların, bu dökümanlar üzerinde işe yarar çıkarımlar yapabilmesi için, bir önceki bölümde bahsettiğimiz RDF yapısının basit semantiğinden fazlasına ihtiyaç vardır. OWL yapısı, nesne ve ilişki tanımlamalarında daha fazla anlam içermek isteyenler için oluşturulmuştur. OWL dili ile özellikler, nesne ve veri tipi olmak üzere ikiye ayrılmış, küme işlemleri getirilmiş, özelliklerin alabileceği değerler üzerinde kısıtlamalar koyabilme ve eşleme ilişkileri tanımlayabilme sağlanmıştır (Herman, 2007a).

W3C, ifade etme gücüne göre ölçeklendirilmiş, OWL Full, OWL DL ve OWL Lite olmak üzere üç OWL versiyonu önermiştir.

OWL fikrinin temelini, karar verme yöntemlerine uygun ontolojilerin en etkili şekilde sunumu oluşturmaktadır. Ontolojiler dağıtık olabilir ve OWL, ontolojilerin, farklı ontolojilerde bulunan terimleri ya da kavramları kullanabilmesine olanak sağlar. Bunu, RDF tarafından sunulan ilişkilendirme özelliğini farklı sistemlere dağıtmış ontolojiler arasında bağlantı kurarak sağlar. Bu sayede, ontolojiler, farklı sistemlerde dağılmış durumda olan bilgileri birleştirmek ya da aramak isteyen uygulamalar için kritik yetenekler olmuşlardır.

Aslında, XML DTD’ler ve XML şemaları tanımlarda önceden anlaşmış farklı sistemler arasında veri alış verişinde kullanılabilir. Ancak, bu yapının semantik eksikliği, bu sistemlerin veri alış verişinde farklı XML sözcük hazinesine sahip makineler arasındaki iletişimi güvenli şekilde sağlamalarını engeller. Örneğin, aynı terim farklı bağlamlarda farklı anlamlara gelebilir. Aynı zamanda, farklı terimler, aynı şeyi ifade etmede kullanılıyor olabilir. Bu sebeple, RDF ve RDF Şemalarına, tanımlayıcılarla ilişkilendirilmiş basit semantikler eklenmesi mümkün kılınmıştır. En genel anlamda sınıflar ve özelliklerin özne ve nesne bilgilerine ek olarak bunlar

arasında birtakım hiyerarşik ilişkiler tanımlanabilmesi sağlanabiliyor. Bu anlamda, RDF şema da aynı zamanda bir ontoloji dilidir. Ancak, çok sayıda ve otonom olarak geliştirilen ve yönetilen şemanın birlikte çalışabilmesi için daha zengin semantik yapılar gereklidir. Örneğin, RDF şemaları kullanılarak, Kişi ve Araba sınıflarının birbirini dışladığını veya bir grubun tam olarak dört elemanı olması gerektiğini belirtemeyiz (Heflin, 2004).

Ontolojiler, taksonomik bilgilere ek olarak bir takım kural kümelerine de sahip olabilir. Bu çıkarım kuralları, dilin ifade gücünü artırır. Bilgisayarlar, bu kuralları ontolojilerle birlikte kullanarak, terimleri ve kavramları kullanıcılar için daha anlamlı ve kullanışlı olacak şekilde kullanabilirler.

OWL dili kendi içinde birtakım çıkarımları sağlayabilecek şekilde tasarlanmıştır – genellikle sınıflandırma ve hiyerarşik dizilim ile –. Fakat, aynı zamanda, otomatik çıkarım araçları da kullanılabilir. Farklı alanlarda bulunan bilgilerin tamamını kapsayabilecek yorum yeteneğine sahip bir tasarım biçimi sunmak zor olacağından, web'deki bilgilerden çıkarım yapabilecek farklı yaklaşımlar ortaya çıkmıştır. Bu, kurallar ve çıkarım yeteneği ihtiyacını ortaya çıkaran sebeptir. Bu ihtiyaç, farklı kural-temelli formatların birlikte çalışabilmeleri ve desteklenmeleri için RIF (Rule Interchange Format - www.w3.org/2005/rules) adında bir formatın belirlenmesi sonucunu doğurmuştur.

Öte yandan, web'de ontoloji kullanan sayfalar arttıkça, ortaya çıkan bir takım terminoloji problemlerine çözüm üretme gerekliliği de ortaya çıkmıştır. Önceki gibi, adres bilgisi içeren bir ontolojiyi kullanmak istediğimizde, bir kullanıcı ya da uygulama servisi zip kodu içeren bir adres tanımlarken, başka bir kullanıcı ya da uygulama servisi aynı adresin posta kodu içeren halini tanımlayabilir. Bu durumda, elimizde bulunan iki adres bilgisinin de aslında aynı adresi belirttiğini anlayabilmek için, ontolojilerin bir takım eşitlik ilişkilerini içermesi gerekmektedir. Yani, tanımlanan ontolojilerden biri yada her ikisi, bir servisin içerdiği zip kodu bilgisinin diğer servisin içerdiği posta kodu bilgisine eşit oluşunu tanımlamalıdır. (Berners-Lee et al., 2001).

Sonuç olarak, ontolojiler, kullanıldıkları alana veya ortamdaki kavram ve tanımların netliğine göre daha fazla veya daha az kullanışlı olabilir. Anlamsal web teknolojisi

geliştikçe ve kullanımı yaygınlaştıkça, bazı ontolojiler elenirken, bazıları standart ve kabul görme yolunda ilerleyeceklerdir. Tabii bu arada insanlar, hayatta kalabilen ontolojilerin içerdiği terimler arasındaki ilişkileri de oluşturabilmelidirler.

2.3 Anlamsal Web Uygulamaları

Anlamsal web ile ilgili çalışmalar amaçları açısından temel olarak üç alanda incelenebilir;

- Bir alanla ilgili ontoloji geliştirme çalışmaları
- Geliştirilen ontolojileri işleyen araçlar ve sistemler geliştirmeyi amaçlayan çalışmalar
- Hem ontoloji geliştirme sürecini hemde ontolojik verileri işleyecek sistemleri geliştirme sürecini kolaylaştırmak için bir takım araçlar geliştirmeyi amaçlayan çalışmalar.

Günümüz anlamsal web teknolojilerinin temelini oluşturan ontolojilerin geliştirilebilmesi için çok farklı alanlarda pek çok ontoloji geliştirilmiştir. Genel amaçlı ontolojilere örnek olarak, iş modellerinin tanımlanması için ortak bir ontoloji çalışması (Andersson, 2006) ve biomedikal araştırmalarla ilgili kullanılan materyaller, üretilen veriler ve bu veriler üzerinde yapılan analizlerle ilgili bilgileri tanımlayan ontoloji çalışması (<http://obi.sourceforge.net/index.php>) verilebilir. Bunlara ek olarak, Farrar et al. (2002), dilbilimsel verilerin tanımlanması ve paylaşımı ile bu veriler üzerinde analizler yapabilecek uzman anlamsal web uygulamalarına yardımcı olabilecek, dilbilimsel kavramlar için bir genel amaçlı ontoloji çalışmasıdır. Eğitim alanında da bu tür genel amaçlı ontolojiler üretilmiştir. Örneğin; OntoEdu (Guangzuo et al., 2004), kullanıcı adaptasyonu, gerekli görevlerin kullanıcı isteklerine göre otomatik üretilmesi, eğitim ontolojisi, servis ve içerik modülleri gibi parçalardan oluşan esnek bir e-öğrenme mimarisi önermektedir. Bir diğer çalışmada ise Knight et al. (2005), öğretim tasarımları ile öğrenme nesnelерinin entegrasyonu ve yeniden kullanılabilirliğini maksimize edecek bir kavramsal çerçeve sunmaktadır. Bunun için, öğretim tasarımları ve öğrenme nesneleri arasındaki ilişkiyi sağlayacak bir “öğrenme nesnesi bağlamı” önermektedir. Ek olarak, önerilen çerçeve için üç ontoloji geliştirilmiştir: öğrenme

nesneleri için ALOCoM ontolojisi, IMS-LD için LOCO ontolojisi, öğrenme nesnesi bağlamları için LOCO-Cite ontolojisi.

Bu bahsedilen genel amaçlı ontolojiler yanında, genel amaçlı referans ontolojilerin alt kümesi olarak algılanabilecek, belirli bir bilgi alanına yönelik ontolojiler de geliştirilmektedir. Aşkar et al. (2007), e-öğrenme ortamlarında uygulayabilmek için öğrenme süreçleri ve kavramlarını çıkararak bir K-12 eğitim ontolojisi oluşturmayı amaçlamaktadır. Çalışmada, Türkçe'nin dil yapısından dolayı (imalı) bir anlamsal model kullanılmıştır. Geliştiren POLE eğitsel ontolojisi, çeşitli öğrenme süreçleri ve kavramlarının bir arada kullanımı sonucu ortaya çıkmıştır. POLE, öğretmenlerin, hangi öğrenme nesnelерinin hangi sırada görüntüleneceğini belirleyerek, kendi öğrenme paketlerini kendilerinin hazırlamasına imkan verir. Öğrencilere, uygulanacak bir takım testlere verdikleri cevaplardan yararlanarak yapılan akıllı çıkarımlar sonucu, hangi öğrenme nesnelерini çalışmalarını gerektiği önerilir.

Aşağıda, eğitsel ontolojiler kullanılarak yapılan birkaç örnek çalışma hakkında bilgi verilecektir.

Geliştirilen bir döküman ontolojisi ile yüksek öğretimde kullanılan kaynakların kullanım etkinliğini artırmayı amaçlayan çalışmada (Dinos and Vega Riveros, 2004), ajan temelli bir mimari benimsenmiştir. Geliştirilen döküman ontolojisindeki yazarları tanımlayabilmek için FOAF (Friend Of a Friend) projesinden faydalanılmıştır. Ek olarak, özerk çalışma birimleri, asenkron işlemler gibi ajan temelli mimari özelliklerinden faydalanılmıştır. Ajanların gerçekleştiriminde, JADE (Java Agent Development Framework) (Bellifemine, Poggi and Rimasa, 1999) kullanılmıştır.

Ullrich (2003), geliştirdiği etkinlik geliştirme aracı ile, öğrencinin bilgi düzeyi, öncelikleri, hedefleri ve yeteneklerini de hesaba katarak öğrenme nesnelерini bir araya getirir.

Öğretmenlerin ders tasarlamasına yardımcı olabilmek için yapılmış olan bir çalışmada (Kasai et al., 2005), birden fazla ontoloji entegre bir şekilde kullanılmıştır. Bilgi teknolojisi dersi geliştirmede yardımcı olabilmek için geliştirilmiş olan araç, bilgi teknolojisi eğitimi hedeflerini içeren ontolojileri kullanarak, hem yeni öğrenme nesnesi geliştirmede geliştirilen kaynakların bu hedeflere yönelik

metaveri ile işaretlenmesinde, hem de varolan kaynakların istenen hedeflere uygun olarak sorgulanmasında kolaylıklar sağlamıştır.

Holohan ve diğerlerinin geliştirdikleri OntAWare (Holohan et al., 2005) aracı, anlamsal web teknolojilerini temel alarak uyarlanabilir e-öğrenme içeriği üretmektedir. OntAware, verilen ontoloji bilgisini kullanarak standart e-öğrenme ve diğer öğrenme nesnelerini yarı otomatik olarak üretebilmektedir. Bunun sonucunda belirli bir standarda uyan ders materyalleri ve öğrenme nesneleri üretilmiş olmaktadır. OntAWare'in temel yeteneklerinden birisi de uyarlanabilirliğidir. Eğitimci, girdi olarak kullandığı ontolojiyi değiştirerek, farklı öğretme/öğrenme stratejilerine göre öğrenme nesnelerinin üretilmesini sağlayabilir. Ayrıca, üretilen öğrenme nesneleri ile ilgili konfigürasyon değişikliği yapılarak, üretilen öğrenme nesnesinde öğrenciye nasıl bir gezinim ortamı sağlanacağı belirlenebilir. Örneğin, çok katı (öğrencinin akışa müdahale edemediği) gezinimler üretilebileceği gibi, öğrencinin serbestçe gezinebileceği öğrenme nesneleri de üretilir.

Öğrenme nesnelerinin kullanım ve paylaşımını kolaylaştırmak amacıyla yapılan diğer bir çalışmada (Mohammed and Mohan, 2005) ise öğrenme nesneleri, yazılım ajanları şeklinde ele alınmıştır. Bunun için, bağlama duyarlı öğrenme nesneleri oluşturulmuştur. Bu bağlama duyarlı ve ajan temelli oluşturulan öğrenme nesnesi, varolan öğrenme nesnesi ile birlikte, onunla ilgili karakteristikleri bir ontoloji yapısında tutmaktadır. Bunun sonucunda ortaya çıkan yeni öğrenme nesnesi, ABLO (Agent Based Learning Object), bağlama duyarlı, kendisi hakkında bilgi sahibi olan, gelen isteklere olumlu/olumsuz cevap verebilecek ve kendisiyle ilişkili diğer öğrenme nesnelerini bilecek yeteneklere sahip olmaktadır. Öğrenme nesnesi ile ilgili ontoloji üzerinden çıkarımlar yapma yeteneğine de sahip olan yeni öğrenme nesnesi, aynı isimle kullanılmış olsa bile farklı olan kavramları ayırt edebilmektedir. Örneğin; elimizde "Ağaç" hakkında, biri bilgisayar bilimlerindeki ağaç veri yapısını diğeri bitki olan ağacı betimleyen iki öğrenme nesnesi olsun. "Ağaç, Veri Yapıları" şeklinde yapılan bir sorgu sonucunda bitki olan ağacı içeren öğrenme nesnesi bu isteğe olumsuz yanıt verecek ve sonuçta sadece istenen bağlama uygun öğrenme nesnesi olumlu yanıt verecektir.

Farklı sistemlerde kullanılan ontolojileri eşleyen ve entegre bir şekilde kullanılmalarını amaçlayan çalışmada ise (Gašević and Hatala, 2005), öğrenme

nesnelerinin birden fazla ontolojiden araştırılması desteklenmiştir. Çalışma, farklı ontolojileri temel alan e-öğrenme ortamlarının birlikte çalışabilirliğine de çözüm önerileri sunmaktadır. Temel yaklaşım olarak; SKOS (Simple Knowledge Organization System)'in (Miles and Brickley, 2005) bir parçası olan eşleme ontolojisi kullanılmıştır. Sonuçta, SKOS çerçevesi kullanılarak bir arama algoritması geliştirilmiş ve çözüm, kullanıcıların kendi bağlamlarındaki yerel ontolojilerini tanımlayarak, ilgili öğrenme nesnelerini aramalarını sağlayacak bir sistem olarak sunulmuştur.

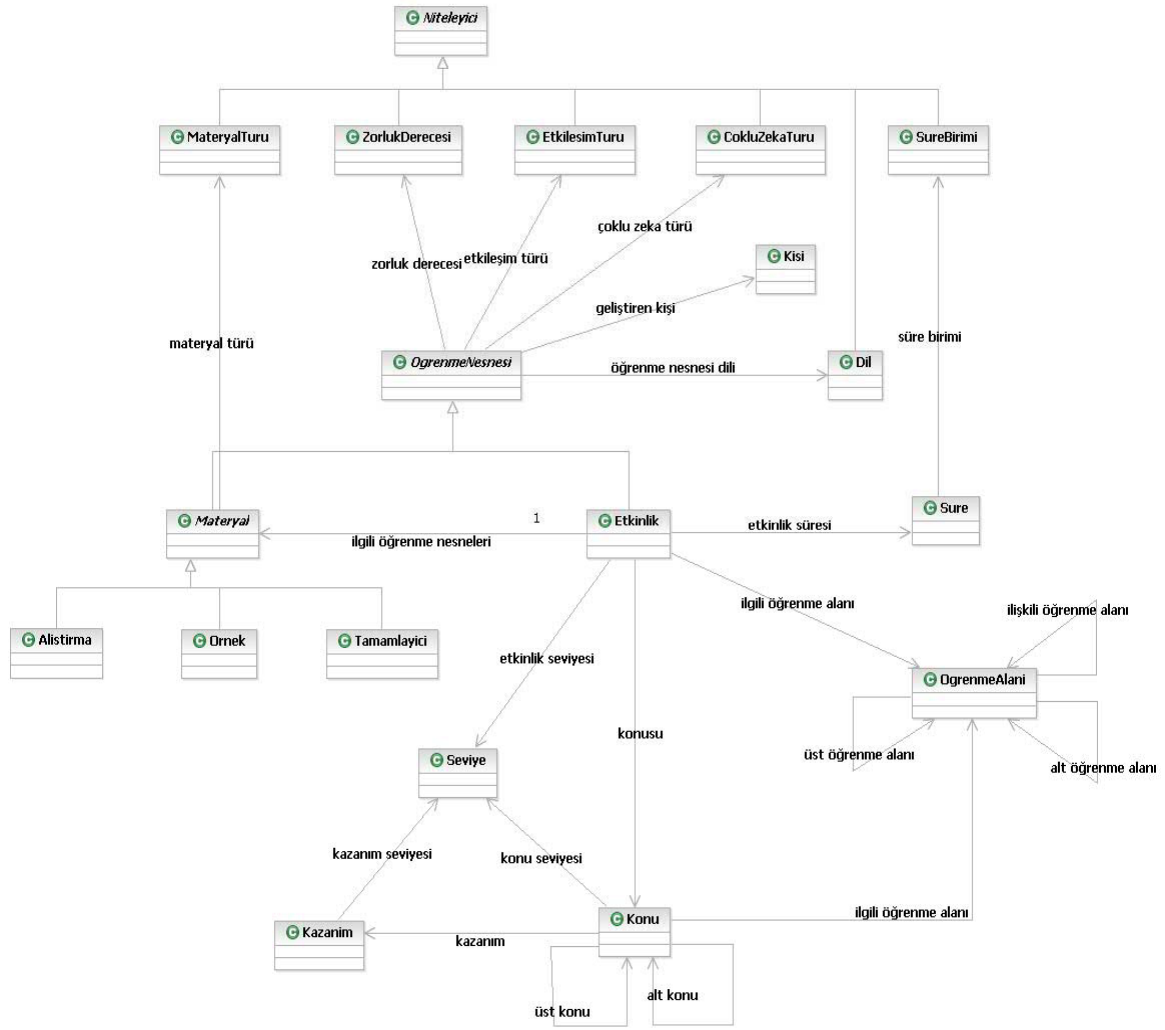
3. YÖNTEM

Sistemin ilk adımı, uygulama gereksinimlerini karşılayabilecek bir ontolojinin oluşturulmasıdır. Zorunlu olmamakla birlikte, oluşturulacak ontolojinin bir takım çıkarımlara da izin verebilmesi, anlamsal web teknolojisinin ve faydalarının uygulamada kullanıcı tarafından daha iyi farkedilebilmesine olanak sağlayacaktır. İkinci adım oluşturulan ontoloji üzerinde istenen işlemleri yapabilecek bir altyapının oluşturulmasıdır. Son adım ise oluşturulan bu altyapıyı kullanarak ontoloji üzerinde gerekli işlemleri yapabilecek kullanıcı arayüzlerinin oluşturulmasıdır.

3.1 Ontolojinin Oluşturulması

Sistemde tüm işlemler, öngörülen etkinlik verilerinin tutulmasına ve bu veriler üzerinde birtakım işlemlerin yapılabilmesine olanak sağlayacak şekilde, gerekli kavramları, kaynakları vb... tanımlayan bir ontoloji üzerinden gerçekleştirilecektir.

Oluşturulan etkinlik ontolojisi, etkinlik adı, ilgili ders bilgileri, süre, seviye, zorluk derecesi, dil, kazanımlar, ilişkili olduğu öğrenme alanları, ilgili materyaller ve hedeflenen zeka türleri gibi bilgileri içeriyor. Veri yapıları ve bunlar arasında tanımlanmış olan ilişkiler Şekil 3.1.1.'de gösterilmiştir. Şekil 3.1.2.'de ise, tanımlanmış yapılarla ilgili sınıf diyagramı verilmiştir.



Şekil 3.1.2. Etkinlik Yönetimi sınıf diyagramı.

Yukarıda görülen ontoloji, şu şekilde yorumlanabilir; öğrenme nesnelерinin geliřtiren kiři, dil, materyal türü, çoklu zeka türü, zorluk derecesi gibi özellikleri vardır. Etkinlik bir öğrenme nesnesidir ve öğrenme nesnelерine ek olarak öğrenme alanı, konu gibi özelliklere sahiptir. Her konunun bir önceki konusu ve bir sonraki konusu olabilir ve her konu birtakım kazanımlara sahip olabilir. Etkinlik ontolojisi daha sonraki bölümlerde daha ayrıntılı olarak irdelenecek ve daha geniş haliyle sunulacaktır.

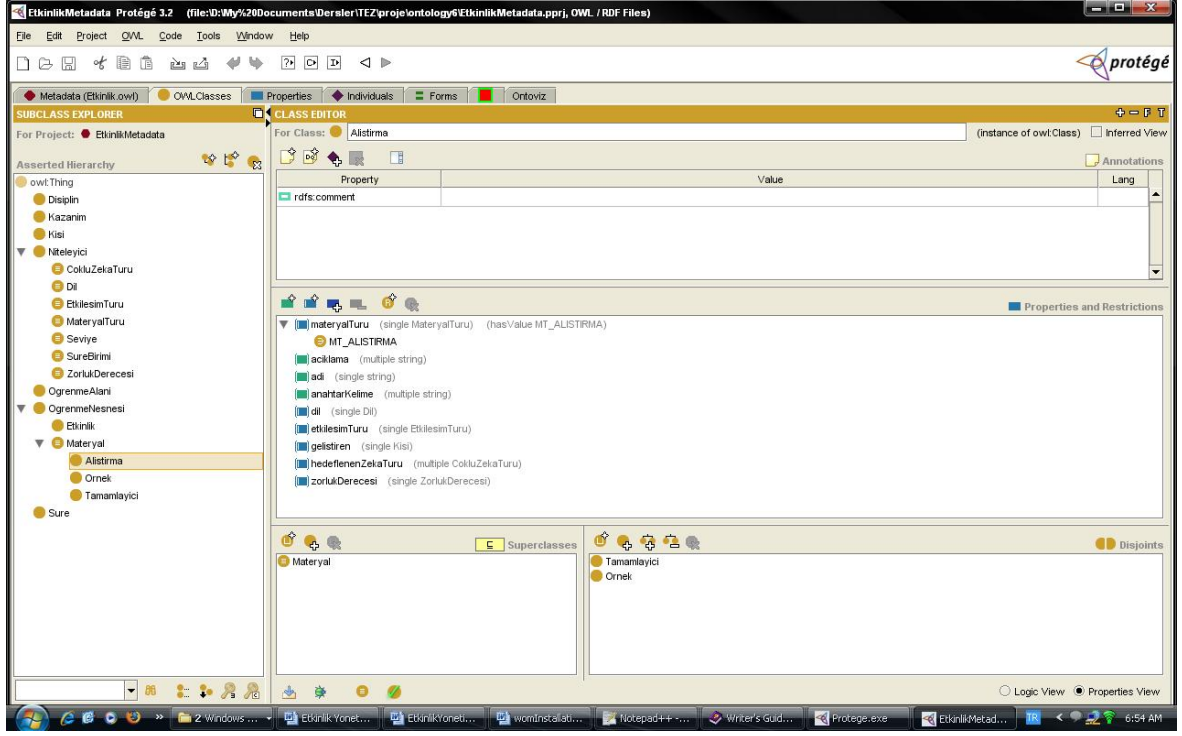
Ontolojiler, Web Ontology Language (OWL) (Herman, 2007a) dili kullanılarak geliřtirilir. Belirli sözdizim özelliklerine sahip olan OWL dilinin temelini XML yapısı oluşturur. Ontolojiler belirlenen sözdizime uyan XML yapısını elle oluşturarak da geliřtirilebilir. Ancak ontoloji elemanları ve bunlar arasındaki ilişkilerin elle takibi çok zordur. Bu yüzden ontoloji geliřtirmeyi kolaylařtıracak bazı programlar üretilmiřtir. Bu programlara örnek olarak; Protégé, OntoEdit (Sure et al., 2002), OilEd (Bechhofer et al., 2001), WebODE (Corcho et al., 2002) ve Ontolingua (Farquhar et al., 1996) verilebilir. Bu çalışmada, etkinlik ontolojisinin geliřtirilmesinde Protégé ontoloji geliřtirme editörü kullanıldı. Ontoloji geliřtirmede Protégé aracı;

- çok aktif bir geliřtirici ve kullanıcı topluluğuna sahip olması,
- daha geliřmiş ve kullanıcı dostu bir kullanım arayüzüne sahip olması,
- daha yetkin ve alternatif ontoloji görüntüleme eklentilerine sahip olması,
- ontoloji geliřtirme, sorgulama, çıkarım vb.. işlemler için geliřtirilmiş pek çok yardımcı anlamsal web araç ve teknolojiyle (Jena, Jess, SPARQL, ARQ, SWRL (Horrocks, 2004), Fact++ (Tsarkov and Horrocs, 2006), RacerPro, Pellet vs..) uyumlu çalışabilmesi,
- yeterli derecede dökümantasyona sahip olması,

gibi özellikleri nedeniyle tercih edilmiştir. (Shankar et al., 2002), (Knublauch et al., 2004) ve (Crubezy et al., 2005) çalışmalarında Protégé aracının seçimi ve kullanımı ile ilgili bilgiler verirken, (Noy and Musen, 2002) ve (Eriksson et al.,

1999), Protégé aracıyla diğer ontoloji geliştirme araçlarının çeşitli açılardan karşılaştırmasını yapmışlardır.

Şekil 3.1.3.'de etkinlik ontolojisini içeren Protégé aracının ekran görüntüsü verilmiştir.



Şekil 3.1.3. Etkinlik ontolojisinin Protégé aracı ile geliştirilmesi.

Yukarıda ekran görüntüsü verilen ontolojinin XML formatında oluşan OWL dosyasının bir kısmı aşağıdaki gibi olacaktır;

Örnek 3.1.1. Etkinlik ontolojisinden bir bölüm.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://www.owl-ontologies.com/Etkinlik.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

```



```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

xmlns:owl="http://www.w3.org/2002/07/owl#"

xml:base="http://www.owl-ontologies.com/Etkinlik.owl">

<owl:Ontology rdf:about=""/>

<owl:Class rdf:ID="OgrenmeAlani"/>

<owl:Class rdf:ID="Disiplin"/>

<owl:Class rdf:ID="Kisi"/>

<owl:Class rdf:ID="OgrenmeNesnesi"/>

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

  <rdfs:subClassOf rdf:resource="#OgrenmeNesnesi"/>

</owl:Class>

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

  <rdfs:range rdf:resource="#Kazanim"/>

  <rdfs:domain rdf:resource="#OgrenmeAlani"/>

</owl:ObjectProperty>

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

  <rdfs:domain rdf:resource="#OgrenmeNesnesi"/>

  <rdfs:range rdf:resource="#CokluZekaTuru"/>

</owl:ObjectProperty>

<owl:Class rdf:ID="CokluZekaTuru" />

<owl:Class rdf:ID="Kazanim"/>

</rdf:RDF>
```

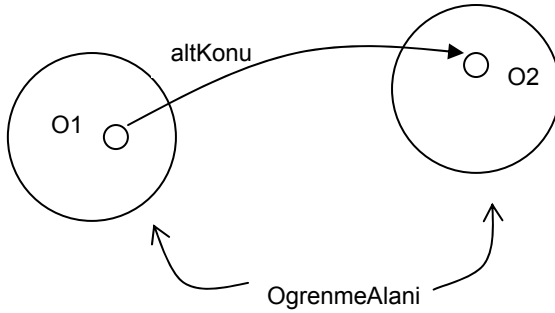
Ontolojiler temel olarak üç ana unsur içerir;

1. Nesne Örnekleri
2. Sınıflar
3. Özellikler

Nesne örnekleri, ilgilendiğimiz konu alanındaki nesnelere temsil ederler.

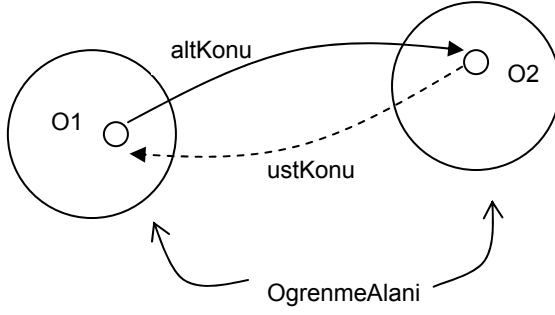
Sınıflar, nesne örneği kümeleri olarak tanımlanabilir. Her bir sınıf, kendi örneklerinin sahip olması gereken özellikleri tanımlar. Yani, her bir öğrenme alanı nesne örneğinin sahip olabileceği özellikler, bir öğrenme alanı sınıfı ile tanımlanır. Ontolojiler geliştirilirken oluşturulan sınıf yapıları da nesneye yönelik programlama tasarımlarında olduğu gibi bir takım hiyerarşik organizasyonlara sahip olabilirler (üst sınıf-alt sınıf ilişkisi gibi).

Özellikler, nesne örnekleri arasındaki ikili ilişkilere dir. Örneğin, bir “OgrenmeAlani” nesne örneği olan “O1”, “altKonu” özelliği ile başka bir “OgrenmeAlani” nesne örneği olan “O2”ye bağlanabilir. (Bkz.: Şekil 3.1.4)



Şekil 3.1.4. Nesne örnekleri arasında ilişki tanımlanması

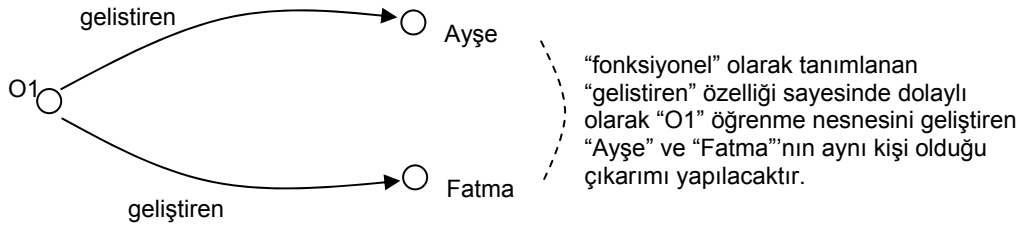
Özellikler, ters yönde de geçerli olabilirler. Örneğin, “altKonu” özelliği, ters yönde geçerli olan bir “ustKonu” özelliğine sahip olabilir. (Bkz.: Şekil 3.1.5)



Şekil 3.1.5. Ters ilişki örneği: altKonu, ustKonu adında bir ters ilişkiye sahip

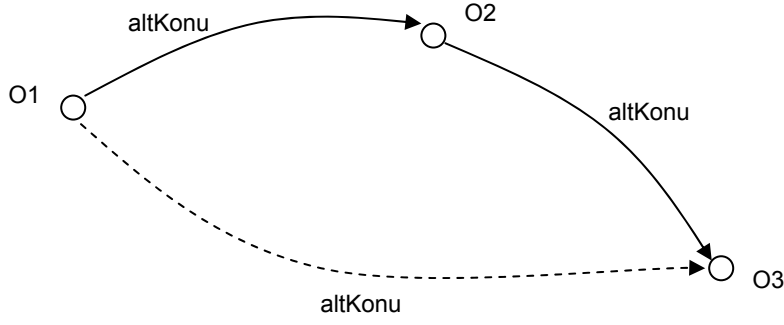
Ayrıca özellikler, “fonksiyonel (functional)”, “geçişli (transitive)” ve “simetrik (symmetric)” karakteristiklerine de sahip olabilir.

Fonksiyonel Özellik: “fonksiyonel” bir özellik, ilgili nesne örneği için, sadece bir tek değere sahip olabilir. Yani, ilgili nesne örneği ile bu özellik üzerinden ilişkili sadece bir nesne örneği olabilir. (Bkz.: Şekil 3.1.6)



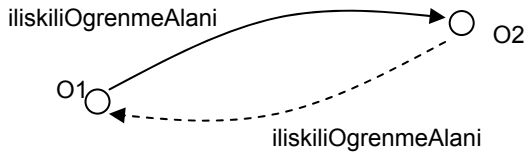
Şekil 3.1.6. Fonksiyonel ilişki örneği: “geliştiren”

Geçişli Özellik: “a” nesne örneğini “b”ye, “b”yi de “c”ye bağlayan bir özellik eğer “geçişli” olarak tanımlanırsa, “a”nın bu özellikle “c” ile ilişkilendirildiği çıkarımını yapabiliriz. Örneğin, bir öğrenme alanı “O1”, “altKonu” özelliği ile “O2”ye, “O2”de, altKonu özelliği ile “O3” ile ilişkilendirilmiş ise; “altKonu” özelliğinin geçişli tanımlanması durumunda “O1”in de bu özellik ile “O3” ile ilişkilendirilmesi çıkarımı yapılabilir. (Bkz.: Şekil 3.1.7)



Şekil 3.1.7. Geçişli ilişki örneği: “altKonu”

Simetrik Özellik: “a” nesne örneği ile “b”yi ilişkilendiren bir özellik, simetrik olarak tanımlanması durumunda, aynı zamanda “b” nesne örneği ile “a” arasında da ilişkinin tanımlanmasına sebep olacaktır. (Bkz.: Şekil 3.1.8)



Şekil 3.1.8. Simetrik ilişki örneği: “iliskiliOgrenmeAlani”

Ontolojinin oluşturulmasından sonraki adım; bu ontolojiyi işleyecek (sorgulama, kayıt, vb...) altyapının oluşturulmasıdır. Aslında Protégé aracı bir bütün olarak ontoloji oluşturma, oluşturulan metaveriye uygun kayıtların girilmesi, sorgulanması, grafik ekranlarda ilişkilerin görsel olarak sunulması gibi bir anlamsal web projesine giriş için hemen her türlü yeteneği sağlıyor. Ancak, bu çalışmada Protégé sadece ontolojinin oluşturulması ve görsel olarak ilişki ağacının sunulması amacıyla kullanıldı. Ontoloji üzerinde yapılacak her türlü işlem için Jena aracından faydalanıldı. Bu seçimde, Protégé'nin de Jena API'lerini temel alarak arayüz sağlamasına ek olarak, Jena API'lerinin kullanımıyla ilgili daha fazla bilgi ve döküman bulunuşu etkili oldu. Özet olarak; ontoloji işleme için de birden fazla araç seçeneği (KAON, IBM Web Ontology Manager) olmasına rağmen Jena'nın seçilmesinde etkili olan sebepler;

- Jena kullanıcı topluluğunun ve forumlarının diğerlerine oranla daha canlı ve gelişmiş olması,
- Jena'nın sağladığı API'lerin daha sağlam ve test edilmiş olduğunun görülmesi,
- Diğerlerinden daha iyi dökümantasyona sahip olması,
- Pek çok anlamsal web geliştirme aracının (Kazuki, Jastor, RDFReactor, vb...) Jena ontoloji modelini ve API'lerini temel alarak geliştirilmiş olması,

olarak açıklanabilir.

Yapılan çalışmada geliştirilen ontoloji, Protégé aracı ile geliştirilmiştir. Geliştirilen ontoloji(ler) Jena aracı kullanılarak işlenmiştir. Bu mimari kullanılmaya çalışıldığında teoride çalışan bu yapının, pratikte ciddi bir takım zorlukları görülmüş ve birtakım yeni araç ve katmanlara ihtiyaç duyulmuştur.

Örnek 3.1.1'deki ontoloji kodu incelendiğinde, elimizdeki metaverinin aslında URI'lerle tanımlanan bir takım kaynaklar (resource) topluluğu olduğu görülecektir. Bunun anlamı, bir veri yapısını (örneğin bir etkinliği) programatik olarak okuduğumuzda, elimize o veri yapısının tamamı (içerdiği diğer özelliklerle birlikte) yerine sadece ilgili URI ile tanımlanan bir kaynak (OWLResource) gelecektir. Bu durumda, bizim için anlamlı olacak bir etkinlik veri yapısını oluşturabilmek için, ontoloji modelindeki kaynaklar arasında URI'leri kullanarak teker teker gezinmek gerekir. Yani, bir Etkinlik nesnesini programatik olarak adım adım inşa etmemiz gerekir. Etkinlik sınıfının içerisinde hiyerarşik olarak başka sınıfların da olduğunu düşünürsek, yapılacak şey bir URI'yı okuyup, ilgili nesnemizin ilgili alanına kurmak ve bunu özyineli bir şekilde tüm Etkinlik nesnemiz oluşana kadar yapmaktır.

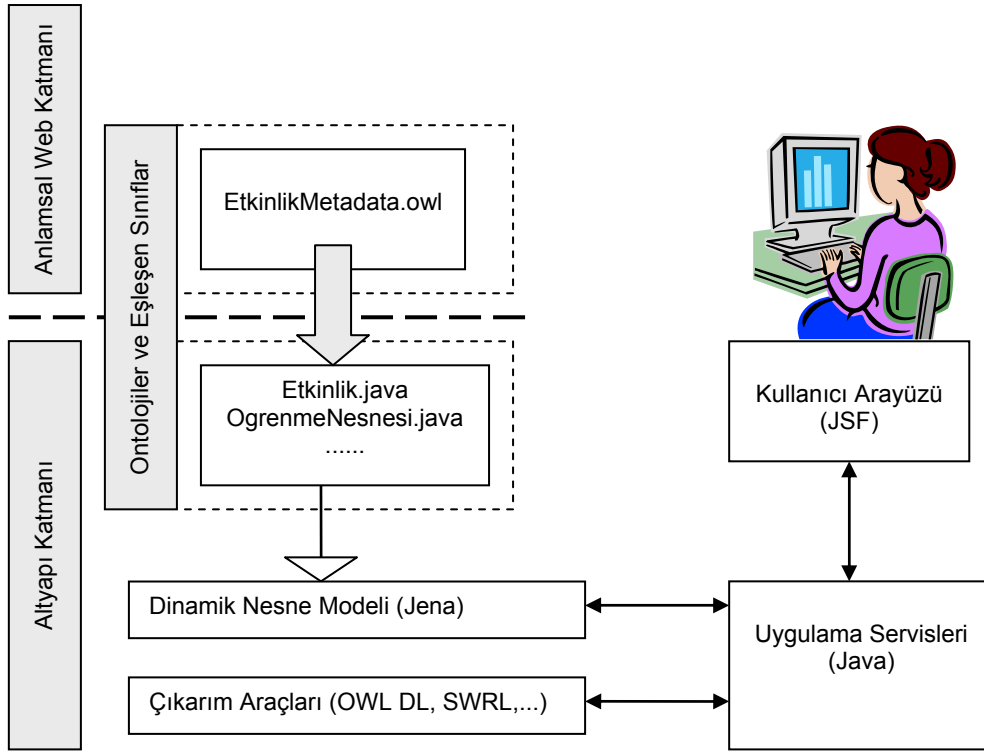
Bu uygulama güçlüğünü aşabilmek için üretilmiş araçları kullanmak, nesneye dayalı bir mimari için iş yükünü ciddi oranda düşürecek bir adım olacaktır. Aslında bu araçlar da bizim yukarıda bahsettiğimiz işlemi yapmakta fakat sonuç olarak programcıyı bu iş yükünden kurtarmaktadırlar. Yapılan araştırmalar sonucunda OWL'den Java sınıflarına çevrim yeteneği sağlayan Kazuki, Jastor, RDFReactor gibi araçlar bulundu. Yapılan araştırmalarda bu konuda ciddi teknoloji eksikliği

olduđu görüldü. OWL yapısından Java sınıflarına çevrim sađlayan araç sayısı az ve hepsinde de ciddi dökümantasyon eksiklikleri ve hatalı durumlar oluşmakta. Örneđin, yukarıda bahsedilen her üç araç da Jena ontoloji modeli üzerinde çalışacak şekilde tasarlanmış. Ancak, çalışmanın yapıldığı tarihte, bu araçlar kullanılmaya çalışıldığında, Jena'nın son versiyonu olan 2.5 versiyonuyla uyumlu olmadıkları görüldü. Kazuki, yeteneklerini içeren bir Eclipse aracı eklentisine sahip, fakat, çalışmanın yapıldığı tarihte bu eklentinin Eclipse uygulama geliştirme arayüzü üzerinde çalışmadığı görüldü.

Ontoloji üzerinde yapılması gereken işleri yürütecek servis katmanı oluşturulduktan sonra sıra bu servisleri kullanarak ontoloji üzerinde kullanıcının işlem yapmasını sađlayacak kullanıcı arayüzlerinin geliştirilmesine geldi. Etkinlik ontolojisini sunacak kullanıcı arayüzü web tabanlı, JSF (JavaServer Faces) teknolojisini temel alan bir uygulama olarak geliştirildi. Web tabanlı kullanıcı arayüzü gereksinimlerini daha kolay karşılayabilmek için, güçlü JSF gerçekleştirmelerinden olan Apache MyFaces projesi kullanıldı. MyFaces projesinin seçilmesinin sebepleri; çok aktif bir kullanıcı topluluđuna sahip olması, iyi dökümantasyon ve gelişmiş pek çok web bileşenini bir arada sunması olarak sıralanabilir.

3.2 Anlamsal Web Uygulaması Mimarisi

Bu bölümde, etkinlik paylaşım uygulaması örnek alınarak, bir anlamsal web uygulaması mimarisi ve geliştirme süreci açıklanmaya çalışılacaktır. Örneklenen anlamsal web uygulaması, temel olarak, geliştirilen bir ontolojinin çeşitli araçlar kullanılarak yönetilmesini amaçlamaktadır. Örnek bir anlamsal web uygulaması mimarisi Şekil 3.2.1'de verilmiştir.



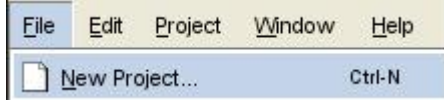
Şekil 3.2.1. Örnek Bir Anlamsal Web Uygulaması Mimarisi

Bu bağlamda, geliştirme süreci başlıca aşağıdaki adımlardan oluşmaktadır;

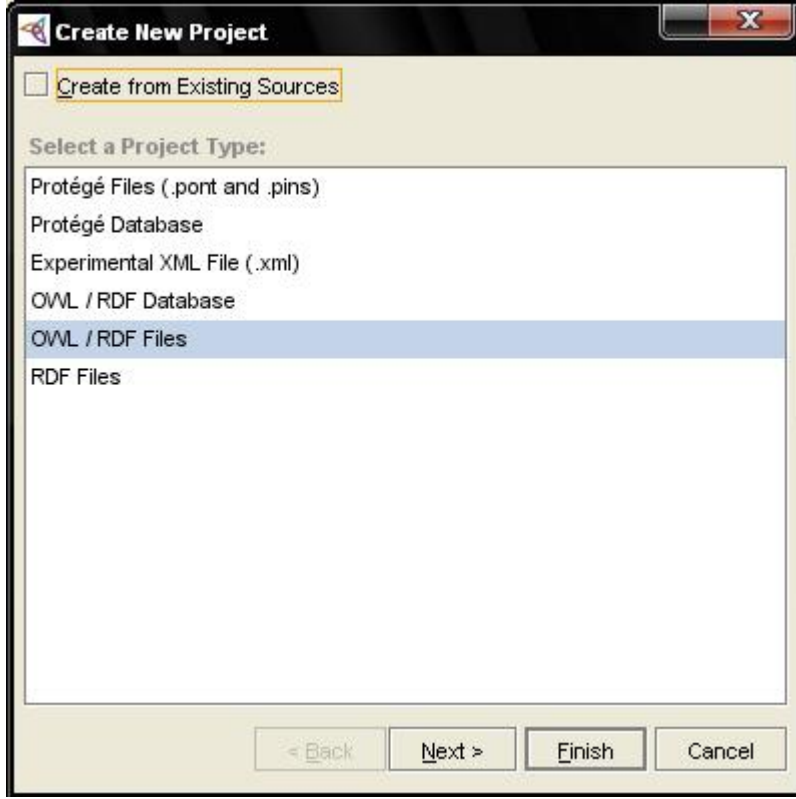
1. Ontolojinin geliştirilmesi,
2. Ontolojinin içeriğini yönetmek için yardımcı Java sınıflarının üretilmesi,
3. Oluşturulan ontoloji yönetim sınıfları kullanılarak ontolojinin sorgulanması ve yeni nesne örneklerinin kaydedilmesi,

3.2.1 Ontolojinin geliştirilmesi

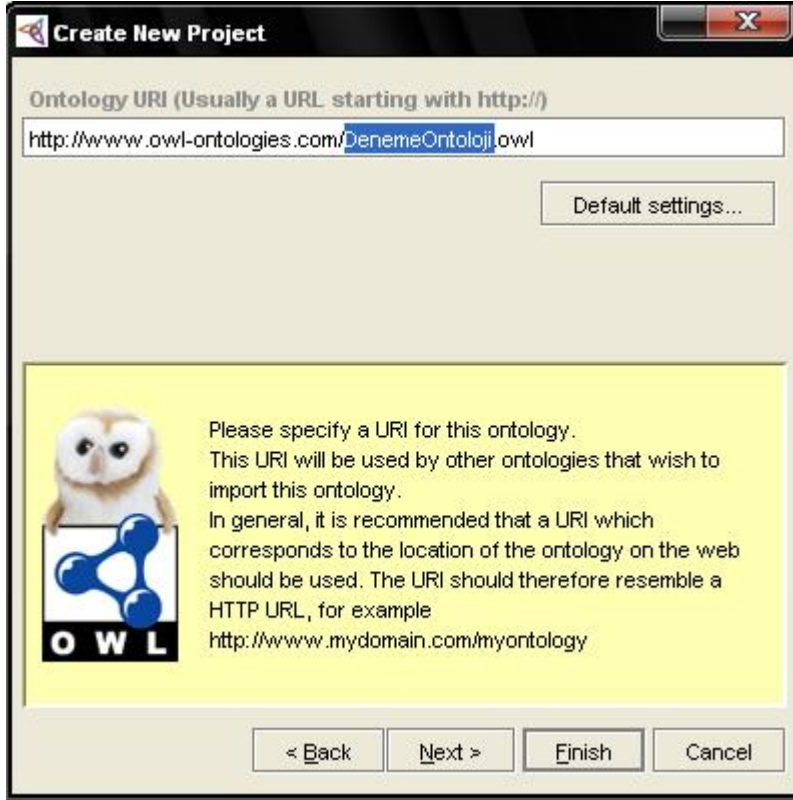
Ontoloji, Protégé aracı kullanılarak geliştirilmiştir. Bunun için Protégé'de yeni bir OWL/RDF projesi oluşturulur (Bkz.: Şekil 3.2.1.1), (Bkz.: Şekil 3.2.1.2) , (Bkz.: Şekil 3.2.1.3).



Şekil 3.2.1.1. Protégé: Yeni Proje Oluşturma Menüsü.

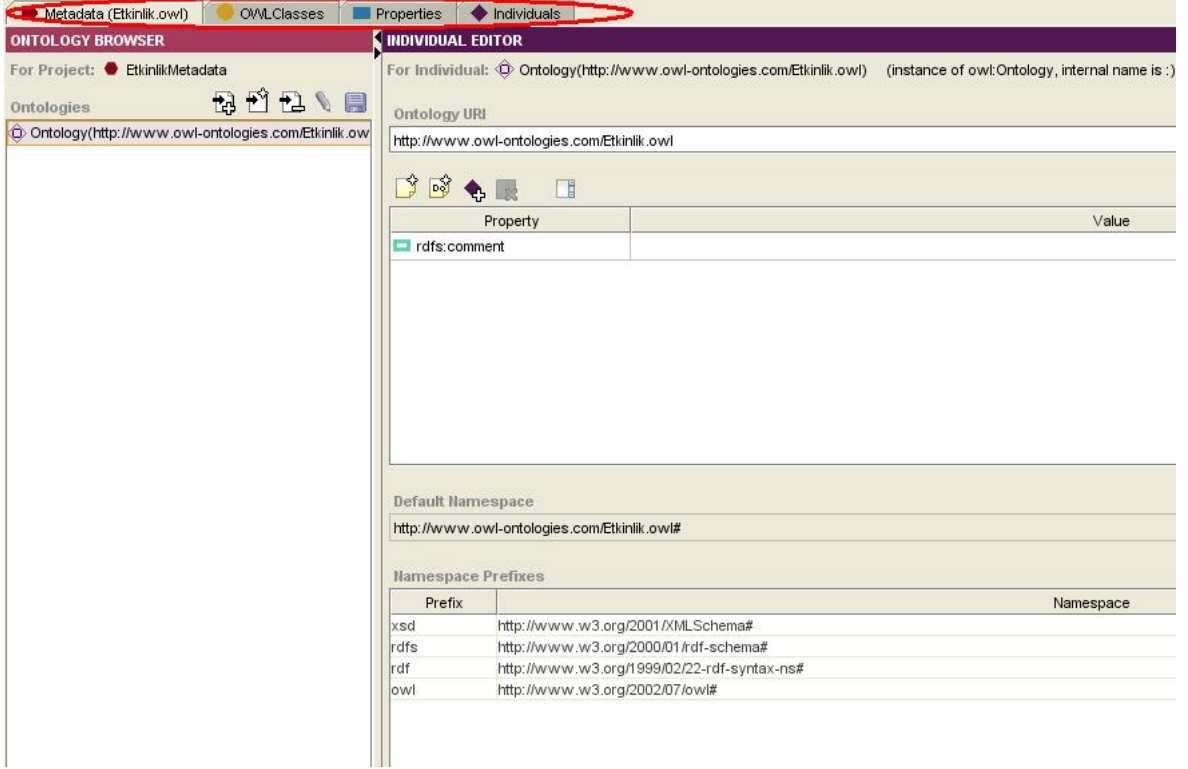


Şekil 3.2.1.2. Protégé: Yeni Proje Oluşturma – Proje Tipi Seçimi.



Şekil 3.2.1.3. Protégé: Yeni Proje Oluşturma – Ontoloji URI Belirtme.

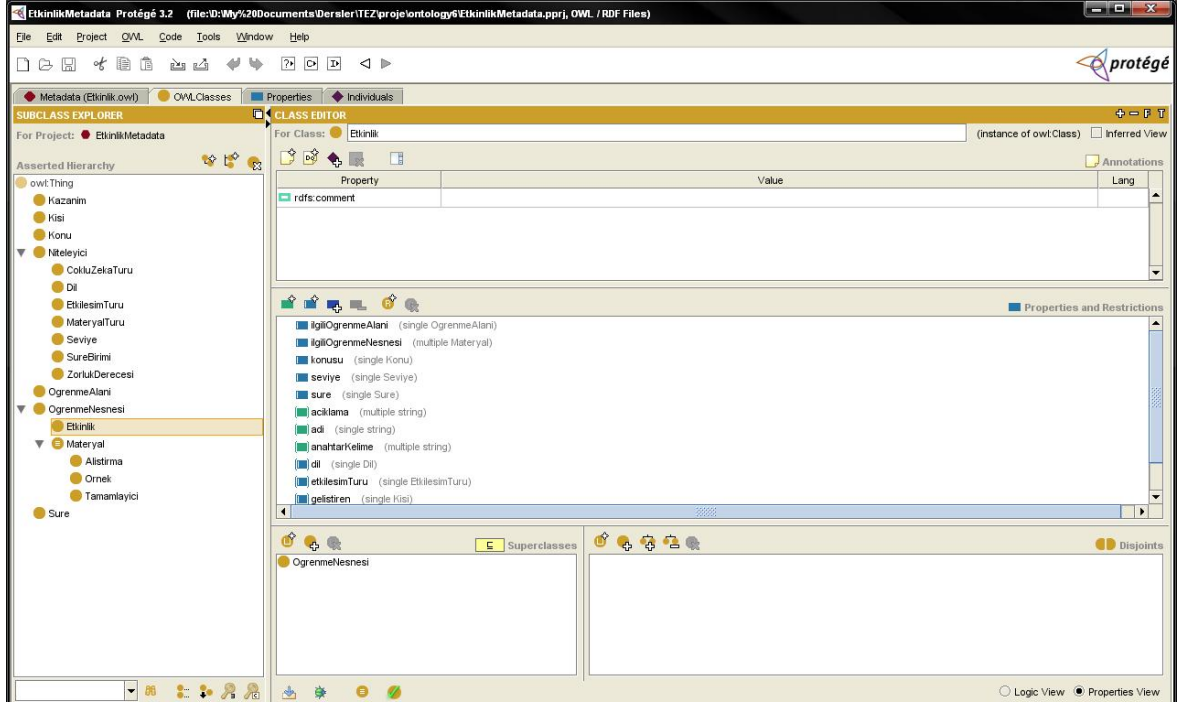
Bu akış sonunda, Şekil 3.2.1.4'deki OWL projesi ekranı görüntülenecektir. Bu ekran, Metadata, OWLClasses, Properties ve Individuals olmak üzere, başlıca dört sekme içermektedir.



Şekil 3.2.1.4. Protégé: Proje Sekmeleri – Metadata sekmesi.

Metatada sekmesinde, o projede kullanılan ontolojiler görülür. Bir ontoloji seçildiğinde, ilgili ontolojinin tanımlayıcı URI bilgisi ve içeriğinde kullandığı dış ontolojilerle ilgili tanımlar görülür.

OWLClasses sekmesinde, ontolojinin içerdiği sınıf tanımları, sınıflar arasındaki hiyerarşi ve herbir sınıfın sahip olduğu özellikler görülür (Bkz.: Şekil 3.2.1.5).



Şekil 3.2.1.5. Protégé: Proje Sekmeleri – OWLClasses sekmesi.

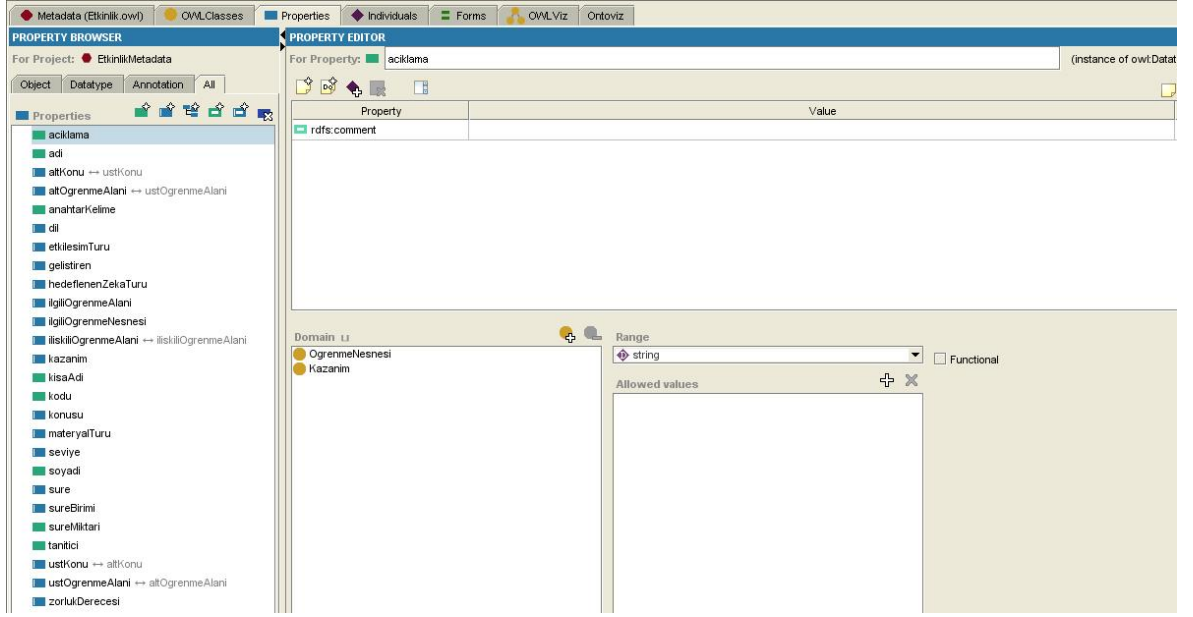
Her bir OWL sınıfı bir “nesnelere kümesi” olarak yorumlanabilir. Sınıf hiyerarşisinde en üstte otomatik olarak eklenmiş olan owl:Thing sınıfı, tüm nesnelere içeren kümeyi temsil eder. Bu sebepten dolayı da, diğer tüm sınıflar owl:Thing sınıfının alt sınıflarıdır. Sınıf hiyerarşisini oluşturabilmek için kullanılan panelin detaylı görüntüsü, Şekil 3.2.1.6’da görülebilir.



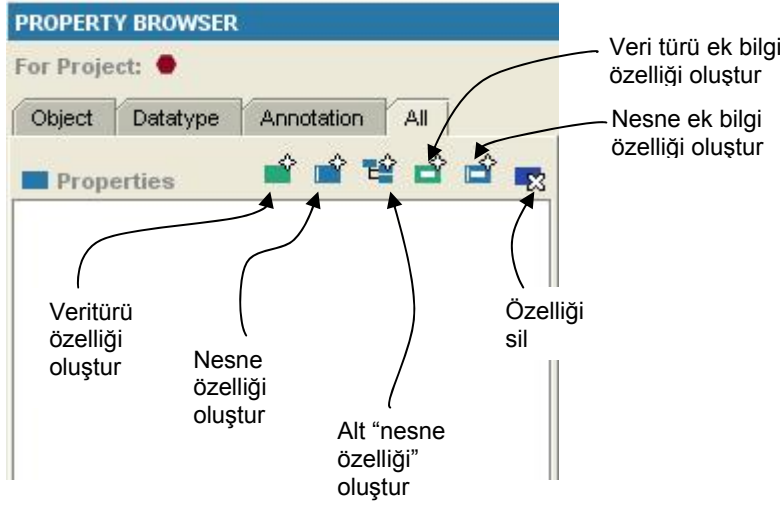
Şekil 3.2.1.6. Protégé: Sınıf hiyerarşisi paneli.

Sınıf özellikleri (property), iki nesne arasındaki ilişkiyi betimlerler. İki ana tür de özellik bulunmaktadır; Nesne özellikleri (Object properties) ve Veritürü özellikleri (Datatype properties). Nesne özellikleri, iki nesneyi birbirine bağlar. Veritürü özellikleri ise, bir nesneyi, bir XML Veritürü'ne (XML Schema Datatype – rdf literal) bağlar. Üçüncü bir özellik türü de ek açıklama (annotation) özellikleridir. Bu özellikler, sınıflara, nesnelere ve nesne/veri türü özelliklerine ek bilgi eklemekte kullanılır.

Hiyerarşi ağacından bir ontoloji sınıfı seçildiğinde, o sınıfın tüm özellikleri OWL Classes sekmesinde görülebilir durumdadır. Ancak, ilgili ontolojide tanımlanmış olan tüm özellikleri görebilmek için Properties sekmesi (Bkz.: Şekil 3.2.1.7) kullanılmalıdır. Şekil 3.2.1.8, Properties sekmesinde bulunan ve OWL özellikleri tanımlamakta kullanılan düğmeleri göstermektedir.



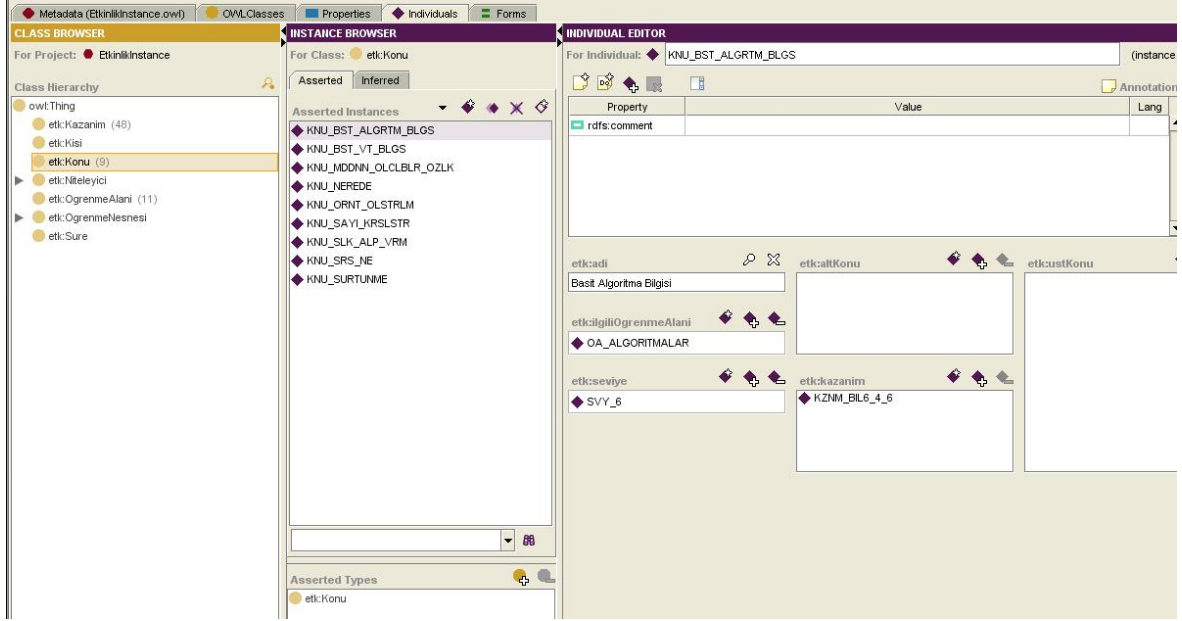
Şekil 3.2.1.7. Protégé: Özellikler sekmesi.



Şekil 3.2.1.8. Protégé: Özellikler paneli.

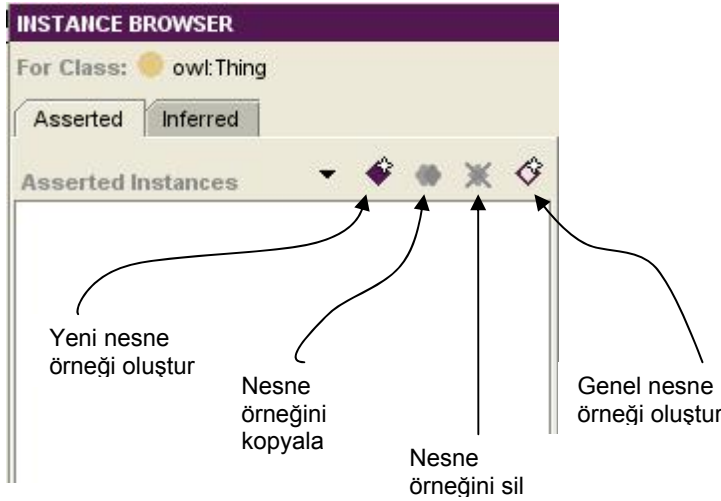
Özellik karakteristikleri ve ilişki türleri ile ilgili gerekli bilgi 3.1. Ontolojinin Oluşturulması bölümünde verilmiştir.

OWL'de istenen tiplerde nesne örnekleri de tanımlanabilir. Oluşturulan nesne örnekleri, aynı zamanda, belli değere sahip sonuçları elde etmek için oluşturulan kısıtlar ve kurallarda da kullanılabilir. Protégé'de nesne örnekleri Individuals sekmesi kullanılarak oluşturulur (Bkz.: Şekil 3.2.1.9).



Şekil 3.2.1.8. Protégé: Nesne örnekleri sekmesi.

Nesne örnekleri panelinde bulunan ve nesne örneği ekleme, silme, kopyalama, vb... işlemler için kullanılan düğmeler Şekil 3.2.1.9'da açıklanmıştır.



Şekil 3.2.1.9. Protégé: Nesne örnekleri paneli.

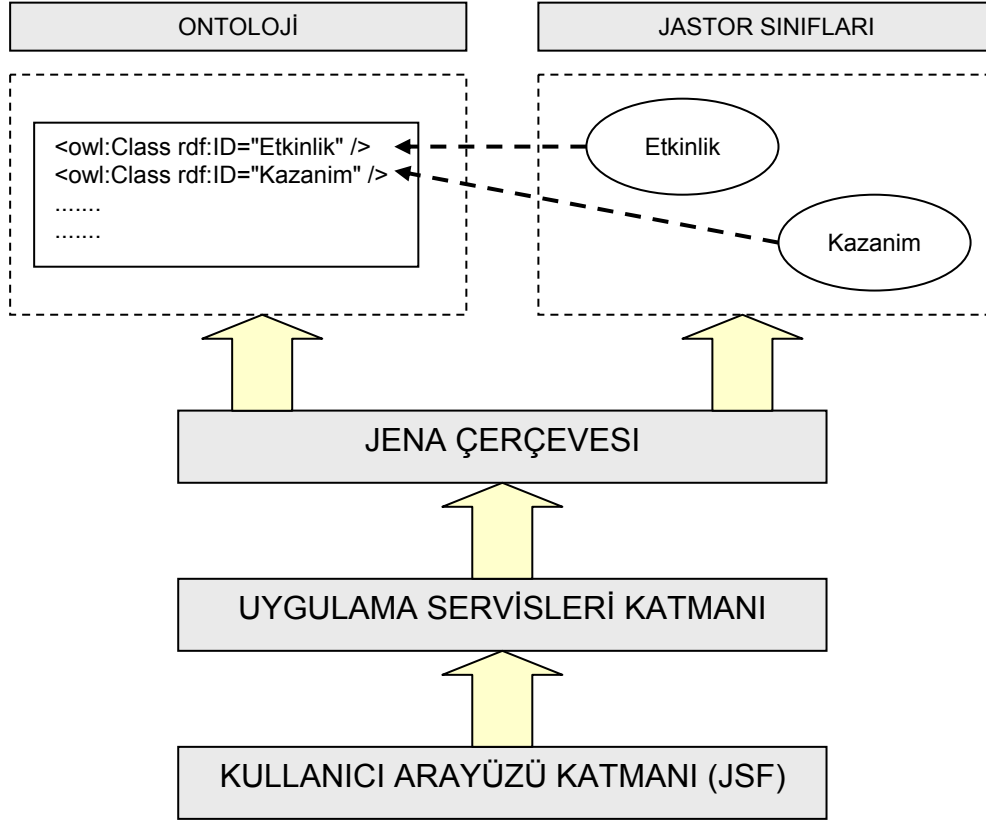
3.2.2 Ontolojinin içeriğini yönetmek için yardımcı Java sınıflarının üretilmesi

Uygulama, nesne yönelimli bir programlama dili olan Java kullanılarak geliştirilmiştir. Bu yüzden, ontolojide bulunan yapıların Java nesnelere şeklinde ifade edilebilmesi ve kullanılabilmesi gereklidir. Jena, OWL sınıflarının, özelliklerin ve nesne örneklerinin, OntClass, Individual gibi Java sınıfları ile ifade edilebilmesini sağlayan bir dinamik nesne modeli sunar. Bu sınıflar bir dereceye kadar soyutlama sağlasa da, ontoloji nesnelere ile etkileşim sadece URI'ler kullanılarak sağlandığı için ontoloji yönetimi hala zorluklar içermektedir.

İdeal durum, ontoloji içerisinde tanımlanmış yapılara karşılık gelecek fakat bu yapılara URI ile erişim güçlüğünden de kurtaracak bir sistemin oluşturulmasıdır. Bunun için, Jastor aracı kullanılmıştır. Jena, OntClass, Individual gibi genel Java sınıfları ile ontoloji yönetimini bir yere kadar kolaylaştırırken, Jastor'un da kullanımıyla, ontoloji içerisinde tanımlanmış tüm yapılara karşılık gelecek Java sınıfları üretilmektedir. Jastor'un ürettiği Java sınıfları Jena ile tam uyumlu olduğu için, Jena API'leri ile entegre bir şekilde rahatlıkla kullanılabilir.

İlk aşamada ontoloji oluşturuldu, ikinci aşamada ise Jastor kullanılarak ontolojideki yapılara karşılık gelen Java sınıfları üretildi. Son aşamada ise, geliştirilen uygulama servislerinin, Jena API'lerini kullanarak, Jastor sınıfları üzerinden

ontolojiyi yönetmesi sağlanmıştır. Oluşan uygulama mimarisinin bir gösterimi Şekil 3.2.2.1.'de verilmiştir.



Şekil 3.2.2.1. Uygulama mimarisi katmanları.

Ontolojideki yapılara karşılık olan Java sınıflarını üretebilmek için kullanılan örnek bir Jastor uygulama kodu EK 1'de verilmiştir.

3.2.3 Oluşturulan ontoloji yönetim sınıfları kullanılarak ontolojinin sorgulanması ve yeni nesne örneklerinin kaydedilmesi

Ontoloji üzerinde işlem yapabilmek için, ilk olarak, Jena API'leri kullanılarak, ilgili ontolojiyi içeren bir ontoloji modeli oluşturulmalıdır. Bir ontoloji modelinin oluşturulması için kullanılan kod örneği Örnek 3.2.3.1'de verilmiştir.

Örnek 3.2.3.1. Jena API'leri kullanarak bir ontoloji modeli oluşturma.

```
// model nesnesi olustur
OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM,
null);

model.getDocumentManager().addAltEntry(EtkinlikMetadata.getURI(),
```



```
EtkinlikMetadada.getOntologyFilePath());

// URI ile ontolojiyi iliskilendir
model.getDocumentManager().addAltEntry(EtkinlikMetadada.getInstanceURI(),
EtkinlikMetadada.getInstanceFilePath());

// ontolojiyi model nesnesine yukle
model.read(EtkinlikMetadada.getInstanceURI());
```

Ontoloji modeli oluşturulduktan sonra, ontolojide bulunan yapılar için oluşturulan sınıfları yönetebilmek için Jastor aracı tarafından üretilen yardımcı sınıflar kullanılabilir (Bkz.: Örnek 3.2.3.2). Ayrıca, doğrudan Jena API metodları da kullanarak ilgili ontoloji modelinin yönetimi sağlanabilir.

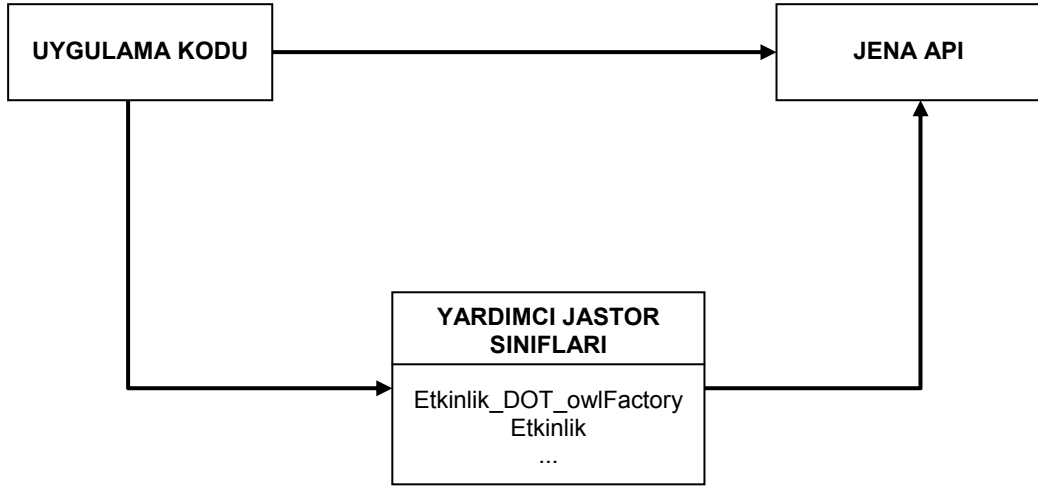
Örnek 3.2.3.2. Jastor yardımcı sınıfları kullanılarak yeni ontoloji sınıfı oluşturma.

```
// yeni bir etkinlik nesnesi olustur.
Etkinlik etkinlik = Etkinlik_DOT_owlFactory.createEtkinlik(
    EtkinlikMetadada.getInstanceURI() + "#Etkinlik_" + uid, model);

// etkinlik nesnesinin alanlarini ekranda girilen degerlerle doldur.
etkinlik.setAdi(etkinlikView.getEtkinlikAdi());
etkinlik.setIlgiliOgrenmeAlani(findOgrenmeAlaniModelObject(etkinlikView.
getOgrenmeAlani()));
etkinlik.setSure(getSure(etkinlikView.getSure(), model, uid));
etkinlik.setSeviye(etkinlikView.getSeviye());
etkinlik.setZorlukDerecesi(etkinlikView.getZorlukDerecesi());
etkinlik.setDil(etkinlikView.getDil());

// model'i yeni haliyle kaydet.
model.write(new
FileOutputStream(EtkinlikMetadada.getInstanceFilePath()));
```

Örnek 3.2.3.2'de de görüldüğü gibi, Jastor kullanarak ürettiğimiz “Etkinlik_DOT_owlFactory” sınıfını kullanarak yeni bir etkinlik nesnesi oluşturuluyor. Bu nesnenin ilgili alanlarını doldurduktan sonra da Jena modelini yeni haliyle kaydediyoruz. Aslında Jastor'un sunduğu yardımcı sınıflar da (Etkinlik_DOT_owlFactory, Etkinlik, vb..) arka planda Jena altyapısını kullanmaktadır. Fakat bu haliyle geliştiriciyi, Jena API'leriyle doğrudan uğraşma zahmetinden kurtarmış olmaktadır. Bu yapı kısaca Şekil 3.2.3.1'deki gibi özetlenebilir.



Şekil 3.2.3.1. Uygulama kodu üzerinden Jastor ve JENA API kullanımı.

Etkinlik sorgulama işlemi için, kullanıcı tarafından girilen sorgu kriterlerinin Jena tarafından modeli sorgulamakta kullanılacak sorgulara dönüştürülmesi gerekmektedir. Bu işlem için bir sorgu kriteri inşa edici yardımcı sınıf üretilmiştir. Bu sınıf, ekranda kullanıcı tarafından girilen kriterleri alarak, Jena'nın ontoloji modelini sorgulaması için gerekli olan sorgulara dönüştürmektedir. Ekranda girilen kriterlerden ontoloji modelini sorgulamak için gereken sorguyu üretmekte kullanılan örnek bir kod yapısı EK 2'de verilmiştir.

Sorgu kriterlerinden etkinlik sorgulama işleminde de, Jena ve Jastor yapılarından faydalanılmıştır. Sorgulama işlemi, Jena API'leri kullanılarak yapılmış, ontolojide bulunan etkinlik yapılarının Java sınıflarına çevrimini sağlamak için ise Jastor sınıfları kullanılmıştır. Jena API'leri kullanılarak verilen sorguyu yapan ve elde edilen OWL sınıflarını Java sınıflarına çevirerek dönen bir sorgulama metodu örneği EK 3'te verilmiştir.

3.3 Geliştirilen Aracın Kullanımı

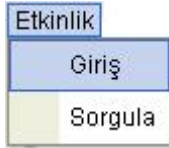
Geliştirilen araç başlıca iki kullanım senaryosu içermektedir. Birincisi; kullanıcı herhangi bir etkinlikten bağımsız öğrenme nesnesi girişi ya da sorgusu yapabilir. Örneğin, bir kullanıcı, geliştirdiği bir öğrenme nesnesini bu senaryo üzerinden sisteme kaydederek, diğer kullanıcıların paylaşımına sunabilir. İkinci kullanım durumu ise etkinlik girişi ya da sorgusu yapılmasıdır. Bu senaryoda, geliştirilen bir

etkinlikle ilgili gerekli bilgiler, öğrenme nesneleri vs.. sağlanarak, bu etkinliğin diğer kullanıcıların paylaşımına açılması sağlanır.

Bahsedilen senaryolardan da anlaşılacağı gibi, uygulamanın gerçekleştirimi, hem öğrenme nesnelere hem geliştirilen etkinlikleri en üst düzeyde paylaşım ve tekrar kullanıma açabilmek üzerine kurulu. İlerleyen bölümlerde bu senaryoların ve ekranların kullanıcı tarafından işletimi için gerekli bilgiler verilecektir.

3.3.1. Etkinlik Giriş

Etkinlik giriş işlemi, ekranın üst kısmındaki Etkinlik menüsünün Giriş alt menüsü seçilerek başlatılır (Bkz.: Şekil 3.3.1.1).



Şekil 3.3.1.1. Etkinlik girişi menüsü.

Bu menünün seçilmesi sonucu Şekil 3.3.1.2.'de görülen etkinlik giriş ekranı görüntülenecektir.

A screenshot of the 'Etkinlik Giriş' (Activity Entry) screen. The screen is divided into two main sections. The left section contains several input fields: 'Etkinlik Adı' (Activity Name), 'Ders' (Lesson) with a dropdown menu showing 'Ders1', 'Öğrenme Alanı' (Learning Area) with a dropdown menu, 'Süresi' (Duration) with a dropdown menu, 'Seviye' (Level) with a dropdown menu, 'Zorluk Derecesi' (Difficulty Level) with a dropdown menu, 'Dil' (Language) with a dropdown menu, and 'Açıklama' (Description) with a text area. The right section contains four list boxes: 'Kazanımlar' (Outcomes), 'İlişkili Olduğu Diğer Öğrenme Alanları' (Related Other Learning Areas), 'İlgili Öğrenme Nesneleri' (Related Learning Objects), and 'Hedeflenen Zeka Türleri' (Targeted Intelligence Types). Each list box has a '+' button to add items and a '-' button to remove items. At the bottom of the screen, there are two more input fields: 'Anahtar Kelimeler' (Keywords) and 'Etkinlik Detay Dökümanı' (Activity Detail Document).

Şekil 3.3.1.2. Etkinlik Giriş ekranı – ilk açılış.

Kullanıcı bu ekranı kullanarak oluşturulacak etkinliğin adını girebilir, ilgili ders, öğrenme alanı, süre, seviye, zorluk derecesi, dil seçimlerini yapabilir. Girilen etkinliğin kazanımları, hedeflenen zeka türleri, ilişkili olduğu öğrenme alanları ve öğrenme nesnelelerini seçebilir. Etkinlik girişinde atanan anahtar kelimeler ile de, etkinlik sorgusunun başarı yüzdesinin artmasına katkı sağlanabilecektir.

Örnek verileri seçilmiş bir etkinlik giriş ekran görüntüsü Şekil 3.3.1.3.'te görülebilir. Gerekli veriler girildikten sonra “Kaydet” düğmesi kullanılarak etkinliğin ontolojiye kaydedilmesi sağlanabilir.

The screenshot shows a web-based form for entering activity details. The form is divided into several sections:

- Activity ID:** A text input field.
- Activity Name:** A text input field containing "10'a kadar sıralama".
- Subject:** A dropdown menu set to "Matematik".
- Learning Area:** A dropdown menu set to "Doğal Sayılar".
- Duration:** A dropdown menu set to "2" and a unit dropdown set to "Saat".
- Level:** A dropdown menu set to "1. Sınıf".
- Difficulty Level:** A dropdown menu set to "Kolay".
- Language:** A dropdown menu set to "Türkçe".
- Keywords:** A text input field containing "Sıra sayılarını gösteren kartlar kullanılabilir".
- Keywords:** A text input field.
- Keywords:** A text input field.
- Buttons:** "Temizle" and "Kaydet" buttons.

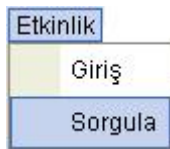
On the right side, there are several sections for selecting related content:

- Kazanımlar (Learning Objectives):** A list of objectives with a search bar and a dropdown menu. The selected objective is "Sıra bildiren ilk on sayıyı kullanır".
- İlişkili Olduğu Diğer Öğrenme Alanları (Related Learning Areas):** A list of learning areas with a search bar and a dropdown menu. The selected area is "Sayılar".
- İlgili Öğrenme Nesneleri (Related Learning Objectives):** A list of learning objectives with a search bar and a dropdown menu. The selected objective is "Deneme Etkinlik1".
- Hedeflenen Zeka Türleri (Targeted Cognitive Types):** A list of cognitive types with a search bar and a dropdown menu. The selected types are "Mantıksal/Matematiksel Zeka", "İçsel Zeka", "Görsel/İzamsal (Mekânsal) Zeka", and "Müziksel Zeka".

Şekil 3.3.1.3. Etkinlik Giriş ekranı – örnek veri girişi.

3.3.2. Etkinlik Sorgulama

Kayıtlı etkinlikleri sorgulamak için kullanılan ekrandır. Ekranı açabilmek için Şekil 3.3.2.1'deki menü adımı kullanılır.



Şekil 3.3.2.1 Etkinlik sorgulama menü adımı.

Etkinlik sorgulamak için kullanılacak ekran görüntüsü Şekil 3.3.2.2.'deki gibidir.

Etkinlik Adı	Ders	Öğrenme Alanı	Süresi	Seviye	Zorluk Derecesi	Dil	Kazanımlar	İlişkili Olduğu Diğer Öğrenme Alanları	İlgili Öğrenme Nesneleri	Hedeflenen Zeka Türleri
--------------	------	---------------	--------	--------	-----------------	-----	------------	--	--------------------------	-------------------------

Şekil 3.3.2.2. Etkinlik Sorgulama Ekranı – ilk açılış.

Etkinlik sorgulama ekranı kullanılarak, istenen kriterlere uygun etkinlikler sorgulanabilir. Sorgulama için, öğrenme alanı, seviye, zorluk derecesi, gerekli kazanımlar, hedeflenen zeka türleri gibi kriterler kullanılabilir.

Örnek sorgu kriterleri seçilerek yapılmış bir sorgu sonucunda elde edilen ekran görüntüsü Şekil 3.3.2.3.'te görülebilir.

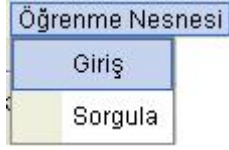
Etkinlik Adı	Ders	Öğrenme Alanı	Süresi	Seviye	Zorluk Derecesi	Dil	Kazanımlar	İlişkili Olduğu Diğer Öğrenme Alanları	İlgili Öğrenme Nesneleri	Hedeflenen Zeka Türleri
10'a kadar sıralama	Doğal Sayılar	Doğal Sayılar	2.0 Saat	1. Sınıf	Kolay	TR				

Şekil 3.3.2.3. Etkinlik Sorgulama Ekranı – örnek sorgulama işlemi.

3.3.3. Öğrenme Nesnesi İşlemleri

Öğrenme nesneleri etkinliklerden bağımsız olarak kaydedilebilir. Böylece bir öğrenme nesnesi, gerekli metaveriler eşliğinde paylaşılacaktır. Şekil 3.3.3.1.'de

Öğrenme Nesnesi giriş menüsü, Şekil 3.3.3.2.'de ise Öğrenme Nesnesi Giriş ekran görüntüsü sunulmuştur.



Şekil 3.3.3.1. Öğrenme Nesnesi giriş menüsü.

A screenshot of the 'Öğrenme Nesnesi Giriş' screen. It features four input fields: 'Öğrenme Nesnesi Türü' (dropdown), 'Dil' (dropdown), 'Zorluk Derecesi' (dropdown), and 'Hedeflenen Zeka Türleri' (text area).

Şekil 3.3.3.2. Öğrenme Nesnesi Giriş ekranı.

4. TARTIŞMA VE YORUM

Bu çalışmada, kullanıcıların girdiği bir takım bilgileri ontoloji tabanlı bir sistem ile kontrol ederek, kullanılan anlamsal web teknolojileri sayesinde daha etkin bir paylaşım ve sorgulama ortamı oluşturulması amaçlanmıştır.

Çalışma süreci iki aşamada değerlendirilebilir. Birincisi; ontoloji geliştirme süreci, diğeri ise geliştirilen ontolojinin anlamsal web araçları ile işleme süreci.

Ontoloji geliştirme süreci, ontoloji ile ilgili kavramların öğrenilmesi, ontoloji geliştirme, geliştirilen ontolojinin kullanımı (geliştirilmesi, yönetilmesi, vb...) için gerekli araçların seçilmesi gibi adımlardan oluşmaktadır. Ontoloji geliştirme aşamasında hissedilen en önemli eksiklik, bu alanda yapılmış pek çok uygulama, araştırma vs.. olmasına rağmen, bir anlamsal web uygulaması geliştirme aşamalarını yeni geliştiricilerin faydalanabileceği düzeyde pratik olarak açıklayan kaynakların eksikliği oldu. Ontoloji ve ilgili kavramlarla ilgili pek çok tanım bulabilmek mümkün. Ontolojinin ne olduğu ve nasıl geliştirilebileceği ile ilgili yeterli kaynak mevcut. Ancak ontoloji geliştirildikten sonra o ontolojinin nasıl kullanılacağı konusunda yapılan çalışma sayısı fazla olmasına rağmen ulaşılabilecek kaynak ve bilgi konusunda sıkıntılar yaşandı. Karşılaşılan pek çok forum, kullanıcı grubu vb.. ortamlarda, bu konuda çok fazla kişinin arayış içerisinde olduğu, ontolojiyi geliştirdikten sonra bu ontolojiyi nasıl kullanmaları gerektiğini bilmedikleri görüldü. Geliştirilen uygulamada da en çok zaman kaybı, basit de olsa, geliştirilen bir ontolojinin uygulama ortamında nasıl kullanılacağı konusunun araştırılması aşamasında yaşandı.

İkinci aşamada ise, uzun uğraşlar sonunda kararlaştırılan bir anlamsal web uygulama geliştirme ortamının ayağa kaldırılması gerekti. Anlamsal web uygulamaları geliştirmeye yardımcı pek çok araç mevcut. Ancak, bu araçların hiçbiri tüm uygulama gereksinimlerini bir tek çatıda toplayabilmiş değildir. Tüm araçlar faydalı bir takım işlevler üstlenmişler ancak bir anlamsal web uygulaması geliştirebilmek için birden fazla aracın entegre edilerek çalıştırılabilmesi gerekli. Bu aşamada karşılaşılan en büyük problem; entegre olarak kullanılması gereken farklı araçların bazı versiyonlarında birbirleri ile uyumsuzluklar yaşanması oldu.

Yapılan arařtırmalar sonucu hangi aralardan, anlamsal web uygulaması mimarisinin hangi katmanında faydalanılabileceđi belirlenmiřtir öncelikle. Bu ařamada da her ara için ayrı ayrı bir takım kullanım bilgilerine ulařılabilmiř olsa da, araların bir arada nasıl kullanılabileceđi konusunda doyurucu bilgiye ulařmak güç olmuřtur. Örnek olacak bir anlamsal web uygulamasını bařtan sona tanımlayarak, ilgili araları tanıtarak ve kullanım/entegrasyon bilgilerini vererek öğretmeyi amalayan alıřmalara ulařılamamıřtır. Bu nedenle, yapılan alıřmanın amalarından biri de; bir anlamsal web uygulaması mimarisini, geliřtirme ařamalarını, kullanılan araları ve bu araların entegrasyonu için yapılması gerekenleri adım adım sunarak, bu alanda yapılacak alıřmalara yardımcı bir kaynak olabilmektir.

Geliřtirilen uygulama, etkinlik ve öğrenme nesnelere ile ilgili ontolojiyi de, kullanıcının girdiđi veriyi de dosya sisteminde tutmaktadır. Fakat, ideal durum, ontoloji dosya sisteminde tutulsa bile, verinin veritabanı yapılarında tutulmasıdır. Bu alıřmanın bir sonraki adımı, girilen etkinlik ve öğrenme nesnesi verisini veritabanında tutarak, ontoloji ile uyumlu alıřabilmesini sađlamak olacaktır. alıřmada, ontoloji ve bu ontoloji ile ilgili oluřturulan nesne örnekleri farklı ontoloji dökümanlarında tutularak, nesne örnekleri geniřleyip artarken metaveri'nin bozulmaması sađlanmıřtır.

Yapılan alıřma, arařtırma ve uygulama olmak üzere iki aıdan geliřtirilme ihtiyacı içermektedir.

Arařtırma konusunda; geliřtirilen etkinlik ve öğrenme nesnesi ontolojisinin, bu bilgi alanının standart ontolojileri ile uyumlu hale getirilmesi gerekmez. Bunun için ontoloji, LOM(Learning Object Metadata) (Duval, E., 2002) ve IMS öğrenim tasarımı (IMS-LD-IM, 2003) standart tanımlarını temel almalıdır. Ek olarak, ontolojinin geçerliliđi internette bulunan uygun bir ontoloji kontrol aracı ile test edilmelidir.

Uygulama konusunda ise; yapılan alıřmanın deđerlendirilmesidir. Bunun için ise, geliřtirilen uygulamanın bir eđitmen grubu üzerinde test edilmesi ve fayda sađlayıp sađlamadıđı, kullanıřlılıđı, vs... test edilmelidir.

KAYNAKLAR

- Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P., Grégoire, B., et al. (2006). Towards a Common Ontology for Business Models. Proceedings of the Open INTEROP Workshop On Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP), Luxembourg, 2006.
- Aroyo, L. & Dicheva, D. (2004). The New Challenges for E-learning: The Educational Semantic Web. *Educational Technology & Society*, 7 (4), 59-69.
- ARQ. (2005). A SPARQL Processor for Jena (Version 1.1) [Computer Software]. Retrieved December 15, 2005, from <http://jena.sourceforge.net/ARQ/>
- Aşkar, P., Kalinyazgan, K., Altun, A. & Pekince, S.S. (2007). An Ontology driven model for e-learning in K-12 education. In T. Kidd & H. Song (Eds.). *Handbook of Research on Instructional Systems and Technology*. (pp. 105-114). Idea Group Reference.
- Barrit, C., Lewis, D. & Wieseler, W. (1999). CISCO Systems Reusable Information Object Strategy Version 3.0. See also: <http://www.cisco.com>
- Bechhofer, J., Horrocks, I., Goble, C. & Stevens R., OilEd: a Reasonable Ontology Editor for the Semantic Web. Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna. Springer-Verlag LNAI Vol. 2174, pp 396--408. 2001.
- Bellifemine, F., Poggi, A. & Rimassa, G. (1999). JADE A FIPA-compliant agent framework. Proceedings of PAAM'99 (pp. 97-108). London, UK.
- Berners-Lee, T., Hendler, J. & Lassila, O., "The Semantic Web," *Scientific Am.*, May 2001, pp. 34-43.
- Corcho, Ó., Fernández-López, M., Gómez Pérez, A., Vicente, Ó., WebODE: An Integrated Workbench for Ontology Representation, Reasoning, and Exchange, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. *Ontologies and the Semantic Web*, p.138-153, October 01-04, 2002
- Crubezy, M., O'Connor, M. J., Buckeridge, D. L., Pincus, Z. S. & Musen., M. A., Ontology-Centered Syndromic Surveillance for Bioterrorism. *IEEE Intelligent Systems*,20(5):26-35. 2005.
- de Freitas, V., Marçal, V.P., Gasparini, I., Amaral, M.A., Proença Jr., M.L., Brunetto, M.A.C., Pimenta, M.S., Ribeiro, C.H.F.P., de Lima, J.V. & de Oliveira, J.P.M. (2002). AdaptWeb: an Adaptive Web-based Courseware. *ICTE - International Conference On Information And Communication Technologies In Education*, 2002. p. 131-134.
- Dicheva, D., Sosnovsky, S., Gavrilova, T., Brusilovsky, P. (2005). Ontological Web Portal for Educational Ontologies. Workshop on Applications of Semantic

Web in E-Learning (SW-EL@AI-ED'05), in conjunction with 12th International Conference on Artificial Intelligence in Education, AI-ED 2005, July 18-22, 2005, Amsterdam, The Netherlands.

- Dinos, J. L. & Vega Riveros, J. F. (2004). A document ontology and agent-based RDF metadata retrieval. Semantic Web Personalization Workshop (pp. 13-17). San José, California, USA.
- Duval, E. (2002), editor. 1484.12.1 IEEE Standard for learning Object Metadata. November 2002.
- Eclipse. (2007). Eclipse development platform (Version 3.3.0) [Computer Software]. Retrieved May 3, 2007, from <http://www.eclipse.org>
- Eriksson, H., Ferguson, R. W., Shahar, Y. & Musen, M. A., Automatic Generation of Ontology Editors. Twelfth Banff Knowledge Acquisition for Knowledge-based systems Workshop, Banff, Alberta, Canada, 1999.
- Farrar, S., Lewis, W. & Langendoen, T. (2002). A Common Ontology for Linguistic Concepts. In Proceedings of the Knowledge Technologies Conference, Seattle, Washington, March 10-13, 2002.
- Farquhar, A., Fikes, R. & Rice., J., The Ontolingua server: a tool for collaborative ontology construction. In Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, 1996.
- FOAF. (2007). The Friend of a Friend (FOAF) Project. Retrieved July 18, 2007, from <http://www.foaf-project.org/>
- Gašević, D. & Hatala, M. (2005). Searching context relevant learning resource using ontology mappings. Proceedings of the 3rd International Workshop on Applications of Semantic Web Technologies for E-Learning (at the 3rd International Conference on Knowledge Capture) (pp. 45-52). Banff, Canada
- Gruber, T. (n.d.). What is an ontology?. Retrieved June 15, 2007, from <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
- Guangzuo, C., Fei, C., Hu, C. & Shufang, L. (2004). OntoEdu: A Case Study of Ontology-based Education Grid System for e-Learning. Global Chinese Computer Journal, 12, 59-72.
- Heflin, J. (2004, February 10). OWL Web Ontology Language Use Cases and Requirements. Retrieved June 12, 2007, from World Wide Web Consortium Web site: <http://www.w3.org/TR/webont-req/>
- Herman, I. (2007, April 24). Web Ontology Language (OWL). Retrieved June 12, 2007, from World Wide Web Consortium Web site: <http://www.w3.org/2004/OWL/>
- Herman, I. (2007, May 25). 3C Semantic Web Activity. Retrieved June 12, 2007, from World Wide Web Consortium Web site: <http://www.w3.org/2001/sw/>

- Holohan, E., Melia, M., McMullen, D. & Pahl, C., "Adaptive e-Learning Content Generation based on Semantic Web Technology", in Proc. of the International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL 2005), held in conjunction with the 12th International Conference on Artificial Intelligence in Education (AIED 2005), July 2005
- Horrocks, I. (2004, May 21). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Retrieved June 12, 2007, from World Wide Web Consortium Web site: <http://www.w3.org/Submission/SWRL/>
- IBM Web Ontology Manager. (2007). A Web-based system for managing Web Ontology Language (OWL) ontologies. Retrieved March 22, 2007, from IBM's web site: <http://www.alphaworks.ibm.com/tech/wom>
- IMS-LD-IM (2003). IMS Learning Design Information Model. Version 1.0 Final Specification, revision 20. IMS Global Consortium. Accessed online on 25 March 2005 at: http://www.imsglobal.org/learningdesign/ldv1p0/imslld_infov1p0.html
- Jastor. (2007). Jastor Typesafe, Ontology Driven RDF Access from Java (Version 1.0.4) [Computer Software]. Retrieved May 28, 2007, from <http://jastor.sourceforge.net/>
- Jena. (2007). Jena Semantic Web Framework (Version 2.5.2) [Computer Software]. Retrieved April 15, 2007, from <http://jena.sourceforge.net/>
- Jess. (2007). Jess the rule engine for the Java platform (Version 7.0-trial), <http://herzberg.ca.sandia.gov/jess/>
- JSF. (2006). JavaServer Faces Technology (Version 1.1). Retrieved July 11, 2006, from <http://java.sun.com/javaee/javaxserverfaces/>
- KAON. (2006). The Karlsruhe Ontology and Semantic Web Tool Suite (Version 1.2.9) [Computer Software]. Retrieved March 18, 2006, from <http://kaon.semanticweb.org/>
- Kasai, T., Yamaguchi, H., Nagano, K., Mizoguchi, R. (2005). A Semantic Web System for Helping Teachers Plan Lessons Using Ontology Alignment. Workshop on Applications of Semantic Web in E-Learning (SW-EL@AIED'05), pages 9–17, July 18-22, 2005, Amsterdam, The Netherlands.
- Kazuki. (2007). Kazuki toolkit (Version 1.5) [Computer Software]. Retrieved May 21, 2007, from <http://projects.semwebcentral.org/projects/kazuki/>
- Kızılkaya, G., Dikbaş-Torun, E., Aşkar, P. (2007). Restructuring e-learning with ontologies. Proceedings of Fifth International Conference on Computational Science and Applications, Kuala Lumpur-Malaysia, 26-29 August 2007, p: 161-165.
- Knight, C., Gašević, D. & Richards, G. (2005). Ontologies to integrate learning design and learning content. Journal of Interactive Media in Education 2005(07).

- Knublauch, H., Dameron, O. & Musen., M. A., Weaving the Biomedical Semantic Web with the Protégé OWL Plugin. First International Workshop on Formal Biomedical Knowledge Representation, Whistler, Canada, 2004.
- Manola, F. (2004, February 10). RDF Primer. Retrieved June 12, 2007, from World Wide Web Consortium Web site: <http://www.w3.org/TR/rdf-primer/>
- Martins, A., Pinto, H. S., Oliveira, A. L., Towards Automatic Learning of a Structure Ontology For Technical Articles, Proceedings of the Workshop on the Semantic Web at SIGIR'04, Jul. 2004.
- Miles, A. & Brickley, D. (2005, November 2). SKOS Core Guide. Retrieved August 15, 2007, from World Wide Web Consortium Web site: <http://www.w3.org/TR/swbp-skos-core-guide/>
- Mohammed, P., Mohan P. (2005). Agent Based Learning Objects on the Semantic Web. Workshop on Applications of Semantic Web Technologies for e-Learning (SW-EL@ AIED'05), Amsterdam, The Netherlands.
- MyFaces. Apache MyFaces Project (Version 1.1.5). Retrieved February 14, 2007, from <http://myfaces.apache.org/>
- Noy, N. F., Musen, M. A., Evaluating Ontology-Mapping Tools: Requirements and Experience. 2002.
- Pellet. (2007). Pellet OWL-DL Reasoner (Version 1.4) [Computer Software]. Retrieved March 17, 2007, from <http://pellet.owldl.com/>
- Protégé. (2006). The Protégé project (Version 3.2) [Computer Software]. Retrieved November 7, 2006, from <http://Protégé.stanford.edu/>
- RacerPro. (2007). RacerPro OWL reasoner and inference server for the Semantic Web (Version 1.9) [Computer Software]. Retrieved April 20, 2007, from Racer systems web site: <http://www.racer-systems.com>
- RDFReactor. (2007). RDFReactor (Version 4.4.8) [Computer Software]. Retrieved August 13, 2007, from <http://rdfreactor.ontoware.org/>
- Richards, G. (2002). Editorial: The Challenges of the Learning Object Paradigm. Canadian Journal of Learning and Technology, 28, (3), 3-10.
- Shankar, R. D., Tu, S. W. & Musen, M. A., Use of Protégé-2000 to Encode Clinical Guidelines. 2002.
- SPARQL. (2006). SPARQL Query Engine for Java (Version 0.8) [Computer Software]. Retrieved December 16, 2006, from <http://sparql.sourceforge.net/>
- Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, F. & Wenke, D., OntoEdit: Collaborative ontology development for the Semantic Web, The Semantic Web - ISWC 2002, First International Semantic Web Conference, Sardinia,

Italy, June 9-12, 2002, Proceedings (Ian Horrocks and James A. Hendler, eds.), Lecture Notes in Computer Science, vol. 2342, Springer, 2002.

- Tsarkov, D. and Horrocks, I. Fact++ Description Logic Reasoner: System Description. In Proc. of the International Joint Conference on Automated Reasoning (IJCAR 2006), 2006.
- Ullrich, C. (2004). Description of an Instructional Ontology and its Application in Web Services for Education. Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning SW-EL'04, 17-23.
- Ullrich, C., An instructional component for dynamic course generation and delivery. In R. Tolksdorf and R. Eckstein, editors, Proceedings of Berliner XMLTage 2003, pages 467–473, 2003.
- Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. Retrieved August 16, 2007, from <http://reusability.org/read/chapters/wiley.doc>

EKLER DİZİNİ

EK-1 JASTOR İLE ONTOLOJİDEN JAVA SINIFI ÜRETME KODU

EK-2 KULLANICI TARAFINDAN GİRİLEN KRİTERLERDEN ONTOLOJİ SORGUSUNUN ÜRETİLMESİ

EK-3 BİR JENA ONTOLOJİ MODELİNİN SORGULANMASI VE SONUÇLARIN JAVA SINIFLARINA ÇEVİRİMİ

EK-1 JASTOR ile ONTOLOJİDEN JAVA SINIFI ÜRETME KODU

```
private static final String METADATA_DIR = "ConfigSource/metadata";

private static final String ETKINLIK_ONTOLOGY_FILE =
METADATA_DIR+"/EtkinlikMetadata.owl";

private final static String ETKINLIK_ONTOLOGY_URI = "http://www.owl-
ontologies.com/Etkinlik.owl";

private static final String TARGET_MODEL_PACKAGE =
"hsn.bote.etkinlikyonet.metamodel";

JastorContext ctx = new JastorContext();

ctx.addOntologyToGenerate(new FileInputStream(ETKINLIK_ONTOLOGY_FILE),
ETKINLIK_ONTOLOGY_URI, TARGET_MODEL_PACKAGE);

JastorGenerator gen = new JastorGenerator(new
File("gensrc").getCanonicalFile(), ctx);

gen.run();
```

EK-2 KULLANICI TARAFINDAN GİRİLEN KRİTERLERDEN ONTOLOJİ SORGUSUNUN ÜRETİLMESİ

```
public String sorguKriteriOlustur() {

    // SORGULANACAK ONTOLOJİLERİN TANITILMASI
    StringBuffer queryString = new StringBuffer(
        "PREFIX metadata: <" + EtkinlikMetadada.getURI() + "#>\n" +
        "PREFIX instance: <" + EtkinlikMetadada.getInstanceURI() + "#>\n" +
        "SELECT ?e " +
        "WHERE {");

    boolean conditionExists = false;
    // VARSA, ETKİNLİK ADI KRİTERİ EKLE
    if (StringUtils.isNotEmpty(getEtkinlikAdi())) {
        if (conditionExists) queryString.append(" . ");
        queryString.append(" ?e metadata:adi ?etkinlikAdi");
        conditionExists = true;
    }

    // VARSA, SEVİYE KRİTERİ EKLE
    if (StringUtils.isNotEmpty(getSeviye())) {
        sayisalKriterEkle("seviye", "tanitici", getSeviye(),
            queryString, conditionExists);
        conditionExists = true;
    }

    // VARSA, ZORLUK DERECESESİ KRİTERİ EKLE
    if (StringUtils.isNotEmpty(getZorlukDerecesi())) {
        sayisalKriterEkle("zorlukDerecesi", "tanitici",
            getZorlukDerecesi(), queryString, conditionExists);
        conditionExists = true;
    }

    // VARSA, DİL KRİTERİ EKLE
    if (StringUtils.isNotEmpty(getDil())) {
        sayisalKriterEkle("dil", "tanitici", getDil(), queryString,
            conditionExists);
        conditionExists = true;
    }
    queryString.append(" }");

    // OLUSAN SORGU DİZİSİNİ GONDER...
    return queryString.toString();
}

// sayisal bir alan icin sorgu ureten yordimci sinif...
private void sayisalKriterEkle(String alanIsmi, String sorguSutunuAdi,
    String alanDegeri, StringBuffer queryString, boolean
    conditionExists) {

    if (conditionExists) queryString.append(" . ");
    queryString.append(" ?e metadata:").append(alanIsmi).append("
?").append(alanIsmi);
    queryString.append(" . ");
    queryString.append(" ?").append(alanIsmi).append("
metadata:").append(sorguSutunuAdi).append(" ").append(alanDegeri);
}
}
```


EK-3 BİR JENA ONTOLOJİ MODELİNİN SORGULANMASI VE SONUÇLARIN JAVA SINIFLARINA ÇEVİRİMİ

```
public List etkinlikSorgula(EtkinlikCriteria etkinlikCriteria) {
    List sorguSonucu = new ArrayList();
    String sorgu = // KRITERLERDEN SORGU OLUSTURULMASI-BKZ:EK2

    // JENA SORGU NESNESİ OLUSTUR
    Query query = QueryFactory.create(sorgu);
    OntModel model = EtkinlikMetadata.getInstanceModel();
    QueryExecution qe = QueryExecutionFactory.create(query, model);
    try {
        // SORGUYU CALISTIR VE SONUCLARI AL...
        ResultSet results = qe.execSelect();

        // OWL SONUCLARININ JAVA SINIFLARINA CEVRIMI...
        for (; results.hasNext();) {
            QuerySolution soln = results.nextSolution();
            RDFNode node = soln.get("e");
            String etkinlikURI = node.toString();

            // JASTOR YARDIMIYLA JAVA SINIFINA DONUSTUR
            Etkinlik etkinlik =
Etkinlik_DOT_owlFactory.getEtkinlik(etkinlikURI, model);
            logger.debug("Bulunan Etkinlik : " +
etkinlik.getAdi());
            sorguSonucu.add(new EtkinlikView(etkinlik));
        }
    } catch (Exception e) {
        logger.error("etkinlik sorgulanırken hata oluştu : " +
e.getMessage(), e);
    } finally {
        // DIKKAT: SORGUYU CALISTIRMAK ICIN ALINAN KAYNAKLARIN
MUTLAKA BOSALTILMASI GEREKIR!
        qe.close();
    }
    return sorguSonucu;
}
```

ÖZGEÇMİŞ

Adı Soyadı : Hasan Türksoy

Doğum Yeri : Yozgat

Doğum Yılı : 1980

Medeni Hali : Evli

Eğitim ve Akademik Durumu:

Lise 1994-1997 Yozgat Lisesi

Lisans 1997-2002 Hacettepe Üniversitesi

Bilgisayar Bilimleri Mühendisliği

Yabancı Dil: İngilizce

İş Tecrübesi:

2002-2004 Yazılım Mühendisi

Likom Yazılım

2004- Yazılım Mühendisi

STM Savunma Teknolojileri Mühendislik ve Tic. A.Ş.