

**ÇEVİRİMİÇİ KİŞİSELLEŞTİRİLEBİLİR SEMANTİK WEB  
ONTOLOJİ GELİŞTİRME ORTAMI**

**AHMET ÖMERCİOĞLU**

**YÜKSEK LİSANS TEZİ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**OCAK 2012**

**ANKARA**

Fen Bilimleri Enstitü onayı

---

Prof. Dr. Ünver Kaynak  
Fen Bilimleri Enstitü Müdürü

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

---

Doç. Dr. Erdoğan DOĞDU  
Anabilim Dalı Başkanı

Ahmet ÖMERCİOĞLU tarafından hazırlanan ÇEVİRİMİÇİ  
KİŞİSELLEŞTİRİLEBİLİR SEMANTİK WEB ONTOLOJİ GELİŞTİRME  
ORTAMI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

---

Doç. Dr. Erdoğan DOĞDU  
Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Çağdaş Evren GEREDE \_\_\_\_\_

Üye : Doç. Dr. Erdoğan DOĞDU \_\_\_\_\_

Üye : Yrd. Doç. Dr. Bülent GÜMÜŞ \_\_\_\_\_

## **TEZ BİLDİRİMİ**

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Ahmet ÖMERCİOĞLU

<b>Üniversitesi</b>	: TOBB Ekonomi ve Teknoloji Üniversitesi
<b>Enstitüsü</b>	: Fen Bilimleri Enstitüsü
<b>Anabilim Dalı</b>	: Bilgisayar Mühendisliği
<b>Tez Danışmanı</b>	: Doç. Dr. Erdoğan DOĞDU
<b>Tez Türü ve Tarihi</b>	: Yüksek Lisans – Şubat 2012

**Ahmet ÖMERCİOĞLU**

## **ÇEVİRİMİÇİ KİŞİSELLEŞTİRİLEBİLİR SEMANTİK WEB ONTOLOJİ GELİŞTİRME ORTAMI**

### **ÖZET**

Klasik anlamdaki web’de veri web sayfası dokümanları üzerine gömülmüştür ve sadece insanlar tarafından okunup anlaşılabilir. Web 3.0 olarak da tanımlanan, Semantik Web (SW), klasik web’in genişletilmiş bir versiyonudur. SW, bilginin açık bir şekilde tanımlanmış bir anlama sahip olduğu, web sayfalarının anlam ifade eden içeriğinin tanımlanmış bir yapısı olduğu ve yazılım sistemlerinin de, web sayfalarını dolaşarak, normalde insanların yaptıkları işlemleri gerçekleştirebildikleri bir web olarak düşünülebilir.

Semantik Web’in yapı taşları sayılabilecek olan ontolojiler, bir bilgi alanını ifade etmeye yarayan kavram ve ilişkileri tanımlarlar. Ontolojiler, belirli bir uygulamada kullanılan terimleri sınıflandırmada, olası ilişkileri tanımlamada ve bu ilişkileri kullanırken konulacak sınırlandırmada kullanılır.

Yazılım sistemlerinin işleyeceği veri, web üzerinde dağınık halde bulunmaktadır. Dağınık veri modellerini bir araya getirmek ve ağ üzerinden diğer yazılım sistemleriyle paylaşabilmek oldukça zorlu bir iş olup, yazılım bütünleştirme uygulaması ve çerçeve programları geliştirilmesi yolunda oldukça fazla çaba harcanmaktadır.

Son zamanlarda sıkça kullanılmaya başlayan ontoloji geliştirme araçları, bu alanda yardımcı olabilecek araçlardandır. Çalışmamızda incelediğimiz ontoloji geliştirme araçları, birçok yönden oldukça gelişmiş olmalarına rağmen, son kullanıcının kendini rahat hissedeceği ve aşına olduğu bir kullanıcı deneyimi sunmak yerine, her kullanıcı için sabit, çoğunlukla kişiselleştirilemeyen bir arayüz ve deneyim sunmaktadırlar.

Bu çalışmamızda, OWL ontolojilerinin oluşturulmasını, ortaklaşa geliştirilmesini ve kullanıcıların kendi çalışma alanlarını kişiselleştirmesini sağlayan bir web ontoloji geliştirme ortamı gerçekleştirilmiştir. Geliştirilen araç mevcut benzerleri ile detaylı bir şekilde karşılaştırılmıştır.

Geliştirilen aracın ontoloji görünümünü kişiselleştirilebilmesi, bu görünümü asıl-detay ilişkisi (master-detail relationship) içinde birbirine bağlayabilmesi ve kişiselleştirilen bu çalışma alanlarının sunucu üzerinde saklanıp, istendiğinde geri çağrılabilmesi işlemlerini gerçekleştirebilmesi özellikleri ile diğerlerinden farklı olduğu gösterilmiştir.

**Anahtar Kelimeler:** Semantik Web, Ontoloji, OWL, SPARQL

**University** : TOBB University of Economics and Technology  
**Institute** : Institute of Natural and Applied Sciences  
**Science Programme** : Computer Engineering  
**Supervisor** : Associate Professor Dr. Erdoğan DOĞDU  
**Degree Awarded and Date** : M.Sc. – February 2012

**Ahmet ÖMERCİOĞLU**

**ONLINE CUSTOMIZABLE ONTOLOGY DEVELOPMENT  
ENVIRONMENT FOR SEMANTIC WEB**

**ABSTRACT**

In the current web, data is embedded in web page documents which are only readable and understandable by human beings. Semantic Web (SW), also known as Web 3.0, is an enhanced version of the classic web. SW can be thought of as a web in which, the information should have a clearly defined meaning, the content of web pages has a clearly defined structure and that software systems can traverse web pages and perform those operations that are normally performed by human beings.

Ontologies, which is considered the building blocks of the Semantic Web, define the concepts and relationships that are used to express an information domain. Ontologies classify the terms which are used in a given application, define the possible relationships and the restrictions for using these relations.

The data that software systems process are distributed on the web. Interconnecting and sharing these distributed data with other software systems over the network is a very challenging task. A lot of effort is spent on developing software integration systems and application development frameworks.

Ontology-development tools which recently began to be widely used can help in this area. Ontology development tools examined in this study, although they are quite advanced in many ways, provide fixed, mostly non-customizable user interfaces and user experiences, instead of providing a user experience that feels comfortable and familiar with.

In this work, we have implemented an online ontology development environment that allows the users to develop OWL ontologies jointly and at the same time personalize their own workspaces. We have conducted a detailed comparison of our system with the similar existing tools.

We have shown that our system is different from the others in the sense that it can personalize the ontology views, connect these views with master-detail relationships and store the workspace on the server and reuse this workspace when needed.

**Keywords:** Semantic Web, ontology, OWL, SPARQL

## **TEŐEKKÖR**

Çalıőmalarım boyunca kıymetli tecrübelerinden faydalandığım, yardım ve katkılarıyla beni yönlendiren hocam sayın Doç. Dr. Erdoğan DOĐDU' ya ve TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliđi Bölümü öğretim üyelerine teşekkürü bir borç bilirim.



## ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4-1: Ontoloji geliştirme araçları teknik özellikleri karşılaştırması	43
Çizelge 4-2: Ontoloji geliştirme araçları fonksiyonel özellikleri karşılaştırması	43

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2-1: Semantik Web Katmanlı Kek [3]	3
Şekil 2-2: Örnek XML dokümanı [6]	5
Şekil 2-3: Örnek XML Şeması [6]	6
Şekil 2-4: Örnek RDF/XML [8]	7
Şekil 2-5: SPARQL olarak “Afrika ülkelerinin başkentleri nelerdir?” sorusu [13]	10
Şekil 3-1: OntoWiki [17]	12
Şekil 3-2: WebProtégé [20]	13
Şekil 3-3: MoKi [23]	14
Şekil 3-4: TopBraid Ensemble	16
Şekil 4-1: SewIde Kullanım Şekli Diyagramı	18
Şekil 4-2: SewIde Yazılım Katmanları	20
Şekil 4-3: NuSphere PhpED	25
Şekil 4-4: Adobe Flash Builder 4.5	27
Şekil 4-5: Temel Erfurt Paketleri	28
Şekil 4-6: SewIde Model-Görünüm-Denetleyici yapısı	29
Şekil 4-7: SewIde'nin Model Güdümlü Mimari ile benzerliği	32
Şekil 4-8: Kullanıcı giriş ekranı	36
Şekil 4-9: Çalışma ortamı işlemleri menüsü	37
Şekil 4-10: Pencere işlemleri menüsü	38
Şekil 4-11: Veri kaynağı seçim ekranı	39
Şekil 4-12: Sınıf özellikleri seçim ekranı	40
Şekil 4-13: Örnek veri görünüm ekranı	41
Şekil 4-14: Sınıf tasarımcısı ekranı	42
Şekil 4-15: 2007 yılı itibariyle PHP dilinin yaygınlığı	44
Şekil 4-16: Flash Player yaygınlığı	45

## KISALTMALAR

<b>Kısaltmalar</b>	<b>Açıklama</b>
<b>APOSDLE</b>	Advanced Process- Oriented Self- Directed Learning Environment
<b>AMF</b>	Action Message Format
<b>BPMN</b>	Business Process Model and Notation
<b>CIM</b>	Computation Independent Model
<b>CSS</b>	Cascading Style Sheets
<b>DI</b>	Dependency Injection
<b>DL</b>	Description Logic
<b>DTD</b>	Document Type Definition
<b>FTP</b>	File Transfer Protocol
<b>GWT</b>	Google Web Toolkit
<b>IDE</b>	Integrated Development Environment
<b>IoC</b>	Inversion of Control
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hyper Text Transfer Protocol
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>JSON</b>	Javascript Object Notation
<b>MDA</b>	Model Driven Architecture
<b>MDD</b>	Model Driven Development
<b>MVC</b>	Model-View-Controller
<b>OMG</b>	Object Management Group
<b>OWL</b>	Web Ontology Language
<b>PHP</b>	PHP: Hypertext Preprocessor
<b>RAD</b>	Rapid Application Development
<b>RDF</b>	Resource Description Framework
<b>RDFS</b>	Resource Description Framework Schema
<b>RIF</b>	Rule Interchange Format
<b>SIOC</b>	Semantically-Interlinked Online Communities
<b>SPARQL</b>	SPARQL Query Language for RDF
<b>SW</b>	Semantic Web
<b>SWRL</b>	Semantic Web Rule Language
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>W3C</b>	World Wide Web Consortium
<b>WWW</b>	World Wide Web
<b>WXS</b>	W3C XML Schema
<b>WYSIWYG</b>	What You See Is What You Get
<b>XML</b>	eXtensible Markup Language
<b>XSD</b>	XML Schema Document
<b>YUI</b>	Yahoo User Interface

## İÇİNDEKİLER

<b>1</b>	<b>GİRİŞ</b>	<b>1</b>
<b>2</b>	<b>SEMANTİK WEB</b>	<b>3</b>
2.1	URI	4
2.2	XML ve XML Schema	4
2.3	RDF	6
2.4	Ontoloji	7
2.5	RDF Schema	7
2.6	OWL	8
2.7	SPARQL	9
<b>3</b>	<b>ÇEVİRİMİÇİ ONTOLOJİ GELİŞTİRME ARAÇLARI</b>	<b>11</b>
3.1	OntoWiki	12
3.2	WebProtégé	13
3.3	MoKi	14
3.4	TopBraid Ensemble	15
3.5	Değerlendirme	16
<b>4</b>	<b>GELİŞTİRME ORTAMI</b>	<b>17</b>
4.1	Gereksinimler	17
4.2	Kullanım Senaryoları	17
4.3	Tasarım	19
4.4	Gerçekleştirim	21
4.5	SewIde Kullanımı	36
4.6	Ontoloji Geliştirme Araçlarının Karşılaştırılması	43
<b>5</b>	<b>DEĞERLENDİRME VE GELECEK İYİLEŞTİRMELER</b>	<b>48</b>
<b>6</b>	<b>KAYNAKLAR</b>	<b>50</b>
	<b>ÖZGEÇMİŞ</b>	<b>53</b>

## 1 GİRİŞ

Web için resmi olarak yapılmış, Web 1.0, Web 2.0, Web 3.0 gibi bir numaralandırma yoktur. Ancak bu ifade tarzı web teknolojilerini sınıflandırabilmek ve web'in gelişim sürecini ifade etmesi açısından genel kabul görmüş bir yaklaşımdır.

Web 1.0, bilginin otomatik olarak işlenebildiği bir ortamdan çok insanlar için doküman paylaşımı sağlayan bir medya olacak şekilde düşünülmüştür. Yayımlanan içeriğinse dinamik bir yapısı yoktur. Sunulan içeriğin biçimi, sunan kişilere bağlıdır ve genelde tek bir biçimde sunulur.

Web 2.0, internet kullanıcılarının da web içeriğine ortak ve demokratik olarak katkıda bulunup, paylaşımlarıyla oluşturdukları içerik şeklinin standart hale geldiği web dönemine verilen isimdir. Bu dönemde insanlar içeriği sadece görüntülemekle kalmayıp, içerik hakkında yorum yapıp, değerlendirmeye hatta içeriğe katkıda bulunmaya başladılar. Wikipedia, Flickr, Youtube, Facebook gibi web siteleri bu kuşağın en çok bilinen uygulamalarıdır. Bu dönem, web içeriğine farklı cihazlarla erişimin de yaygınlık kazandığı dönemdir. Artık web'e erişmek için masaüstü ve dizüstü bilgisayarlar değil; tablet bilgisayarlar, avuç içi bilgisayarlar, cep telefonları da kullanılmaya başlanmıştır.

Bu gelişimin doğal sonucu olarak, Web 3.0 olarak anılan ve dağıtık veri modellerini bir araya getirmek ve ağ üzerinden diğer yazılım sistemleriyle paylaşabilmek için bir model sunan Semantik Web (SW) [1] ortaya çıkmıştır.

SW doğası gereği, yoğun olarak ontolojilerden faydalanmaktadır. İlk olarak ortaya çıkan ontoloji geliştirme araçları klasik masaüstü uygulamalarıydı. Zaman içinde SW'in giderek daha çok kullanılmaya başlamasıyla, kullanıcılar ontolojiler üzerinde ortak olarak çalışma ihtiyacı hissetmeye başladılar. Bu noktada, klasik masaüstü uygulamaları olarak tasarlanan ontoloji geliştirme araçları ortak çalışmalarını desteklemede yetersiz kaldığı için, Web 2.0 teknolojileri kullanılarak hazırlanmış ontoloji geliştirme araçları ortaya çıkmaya başladı.

Bu yeni nesil ontoloji geliştirme araçları, merkezi bir sunucu üzerinde saklanan OWL ontolojileri üzerinde, değişik yerlerde bulunan internet kullanıcılarının ortak olarak çalışıp, değişiklik yapmalarını sağlamaktadır. Ancak bu sistemler çoğunlukla, kullanıcıya çalışma alanını (workspace) kişiselleştirme, kaydetme ve geri yükleme imkânı sunmamaktadırlar.

Bu eksikliği gidermek üzere, kullanıcıların; ontoloji görünümelerini kişiselleştirilebilmesi, bu görünümleri asıl-detay ilişkisi (master-detail relationship) içinde birbirine bağlayabilmesi ve kişiselleştirilen bu çalışma alanlarını sunucu üzerinde saklayıp, istendiğinde geri çağrılabilmesini sağlayan bir sistem geliştirdik.

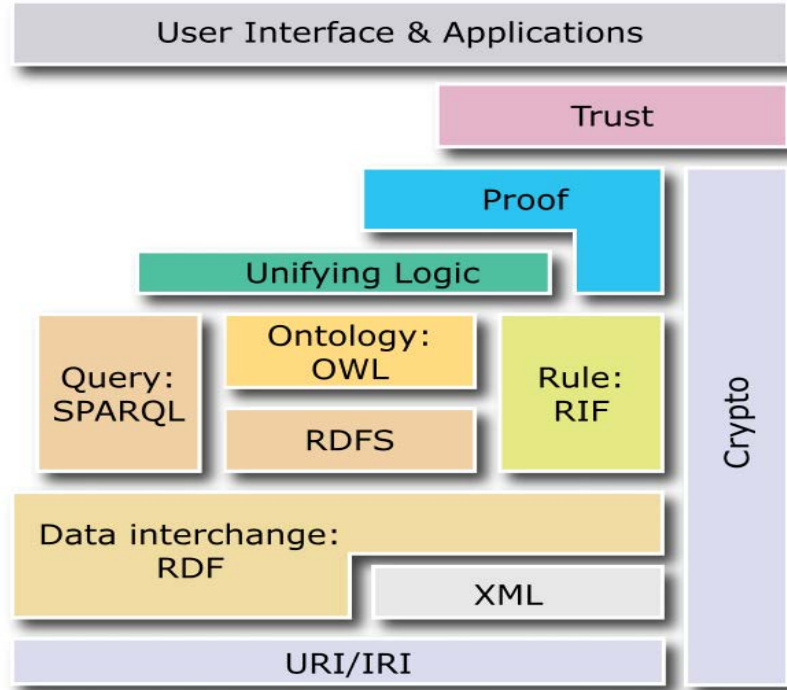
Tezimizin ikinci bölümde Semantik Web ve Sematik Web'in katmanları anlatılmaktadır. Üçüncü bölümde, Web 2.0 teknolojileriyle geliştirilmiş, çevrimiçi ontoloji geliştirme araçları incelenmekte, bu araçların özellikleri geliştirdiğimiz sistemle karşılaştırılmaktadır. Dördüncü bölümde, geliştirmiş olduğumuz Çevrimiçi Kişiselleştirilebilir Semantik Web Ontoloji Geliştirme Ortamı (SewIde)'nin tasarımı ve gerçekleştirim detayları sunulmaktadır. Beşinci bölümde ise SewIde'yi geliştirirken edindiğimiz tecrübeler ve ilgili değerlendirmelerimiz yer almaktadır.

## 2 SEMANTİK WEB

Web 3.0 olarak da tanımlanan, Semantik Web, klasik web'in genişletilmiş bir versiyonudur [1]. W3C tarafından tanımlanmış Semantik Web katmanları Şekil 2-1' de gösterilmiştir.

SW kavramı, 2001 yılında Tim Berners Lee ve arkadaşları tarafından ortaya atılmıştır [2]. Buna göre, İnternet üzerindeki veri web sayfası dokümanları üzerine gömülmüştür ve sadece insanlar tarafından okunup anlaşılabilir.

SW [2], bilginin açık bir şekilde tanımlanmış bir anlama sahip olduğu, web sayfalarının anlam ifade eden içeriğine bir yapı getirmeyi ve insanların olduğu kadar, yazılım sistemlerinin de, web sayfalarını dolaşarak, normalde insanların yaptıkları işlemleri, gerçekleştirebildikleri, yani ortak olarak çalışabildikleri, bir web olarak düşünülebilir. Böylece web hem insanlar tarafından okunan hem de bilgisayar sistemleri tarafından işlenebilen bilgiyi taşıyan bir yapıya kavuşabilecektir.



Şekil 2-1: Semantik Web Katmanlı Kek [3]

Takip eden bölümlerde ilk olarak URI, URL gibi temel web kavramlarına kısaca değineceğiz. Daha sonra Semantik Web'in yapı taşlarından XML, XSD, RDF, RDFS kavramları kısaca açıklanacaktır. Son olarak ise ontoloji, OWL, SPARQL kavramları anlatılacaktır. Bu kavramların ne olduğu ve aralarındaki ilişkileri de ilgili bölümlerde bulmak mümkündür.

## 2.1 URI

URI [4], Evrensel Kaynak Tanımlayıcı (Uniform Resource Identifier), herhangi bir şeyi veya internet üzerindeki kaynakları, tekil (unique) bir tanımlayıcı ifade kullanarak belirtmeye yarayan bir dizi metin karakterdir.

URL (Uniform Resource Locator) ise, URI'den farklı bir şeyi temsil eder. URL internet üzerindeki belirli bir kaynağa ulaşmak için gerekli adresi belirtirken, URI ile ifade edilen şey ise, ulaşılabilir bir web kaynağı olabileceği gibi, herhangi bir nesne veya bir kavram da olabilir.

## 2.2 XML ve XML Schema

XML, Genişletilebilir İşaretleme Dili (eXtensible Markup Language), belgelerin elektronik olarak kodlanmasını sağlayan bir dizi kurallar kümesidir [5]. XML veriyi ve veri biçimini birbirinden ayırarak doküman ve verinin hem insanlar hem de farklı bilgi işlem sistemleri tarafından kolayca okunup paylaşılmasını sağlayan, W3C tarafından tanımlanmış bir standarttır. Bir doküman yaratma dili olan, SGML dilinden basitleştirilerek, oluşturulmuştur. Şekil 2-2'de örnek bir XML dosyası gösterilmiştir.

Çeşitli XML şema dillerinden bir tanesi olan XML Schema, Mayıs 2001'de W3C tavsiyesi olarak yayınlanmıştır. W3C standardı olan XML Schema'nın diğer XML şema dilleriyle karıştırılması nedeniyle WXS yada XSD olarak da anılır. W3C gelecek sürümde tercih edilen isim olarak XSD kabul etmiştir.



```
<?xml version="1.0" encoding="utf-8"?>
<Address xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SimpleAddress.xsd">
  <Recipient>Mr. Walter C. Brown</Recipient>
  <House>49</House>
  <Street>Featherstone Street</Street>
  <Town>LONDON</Town>
  <PostCode>EC1Y 8SY</PostCode>
  <Country>UK</Country>
</Address>
```

Şekil 2-2: Örnek XML dokümanı [6]

XML şema dilleri bir XML dokümanının geçerli olabilmesi için uyması gereken kuralları belirtir. Ancak, diğer birçok şema dillerde farklı olarak, XSD geçerliliği belirlemenin yanı sıra, dokümanın belirli veri yapılarına uygun bir bilgi derlemi üretmesine de imkân sağlar. Bu yönüyle bir veritabanındaki tablo şemasını andırır.

Bir şema, eleman ve öznitelik bildirimleri, basit ve karmaşık tip tanımlamalarından oluşan şema bileşenlerini tarif eden bir soyut çoklu veri derlemidir. “Element type”, “attribute type” ve “complex type” olmak üzere üç çeşit bileşen bulunur. “Element type” bir veritabanı tablosu ise “attribute type” o tablonun sütunları gibidir. “Complex type” ise bu iki bileşenin birlikte kullanılmasını sağlayan bir bileşendir.

XSD dokümanları isim uzayları kullanarak organize olurlar. Tüm adlı şema bileşenleri bir hedef isim alanı (namespace) aittir. Bir şema dokümanına aynı veya farklı isim uzaylarına ait diğer şema dokümanları eklenebilir.

DTD’lerden farklı olarak XSD bir elemanın veya özneliğin içeriğinin bir veri türüne uygunluğunu (örn: tarih, gerçek sayı) denetleyebilir.

XSD 19 temel veri türü tanımlar: boolean, string, decimal, double, float, anyURI, QName, hexBinary, base64Binary, duration, date, time, dateTime, gYear, gYearMonth, gMonth, gMonthDay, gDay, NOTATION. Ancak, sınırlama (restriction), liste (list) ve birleşim (union) kullanarak türetilmiş veri türleri tanımlamak mümkündür. XSD belirtimi dâhilinde 25 tane türetilmiş veri türü tanımlıdır. Şekil 2-3’te XML Şemasına bir örnek verilmiştir.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Address">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Recipient" type="xs:string" />
        <xs:element name="House" type="xs:string" />
        <xs:element name="Street" type="xs:string" />
        <xs:element name="Town" type="xs:string" />
        <xs:element name="County" type="xs:string" minOccurs="0" />
        <xs:element name="PostCode" type="xs:string" />
        <xs:element name="Country">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="FR" />
              <xs:enumeration value="DE" />
              <xs:enumeration value="ES" />
              <xs:enumeration value="UK" />
              <xs:enumeration value="US" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Şekil 2-3: Örnek XML Şeması [6]

### 2.3 RDF

RDF, Kaynak Tanımlama Çerçevesi (Resource Description Framework) [7], W3C tarafından tanımlanmış, bilginin kavramsal olarak modellenmesini sağlayan, basit ve genel amaçlı bir üstveri dilidir. RDF metadata model, bir varlığa ait bir özelliği, özne-yüklem-nesne (subject-predicate-object) olarak şekillendirilen üçlüler (triples) topluluğundan oluşur. Örneğin “Ahmet siyah saça sahiptir.” bilgisinin RDF olarak gösterilişi, “Ahmet”, “sahiptir”, “siyah saç” dır. Burada “Ahmet” özneyi, “sahiptir” yüklemi, “siyah saç” nesneyi belirtir. Kavramların arasındaki ilişkilerin bu şekilde ifade edilmesini sağlayan bu sistem Semantik Web’in temel bileşenlerinden birisidir. RDF’in sözdizimi çok sınırlı olup, bu üçlülerden başka elemanlar kullanılamaz. RDF, XML, Notation 3, Turtle, N-Triples biçimlerinde ifade edilebilir. Şekil 2-4’te RDF/XML dosyasına bir örnek görülmektedir.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
    <foaf:primaryTopic>
      <foaf:Person>
        <foaf:name>Tony Benn</foaf:name>
      </foaf:Person>
    </foaf:primaryTopic>
  </rdf:Description>
</rdf:RDF>
```

Şekil 2-4: Örnek RDF/XML [8]

## 2.4 Ontoloji

“Ontoloji, bir kavramsallaştırmanın tanımsallaştırılmasıdır” [9]. Ontolojiler bir bilgi alanını ifade etmeye yarayan kavram ve ilişkileri tanımlarlar. Ontolojiler, belirli bir uygulamada kullanılan terimleri sınıflandırmada, olası ilişkileri tanımlamada ve bu ilişkileri kullanırken konulacak sınırlandırmada kullanılır [10].

## 2.5 RDF Schema

RDFS, Kaynak Tanımlama Çerçevesi için Şema (Resource Description Framework Schema) gösterimi [11], RDF veri modelini genişleten bir tip sistemidir. RDF kaynaklarına bir yapı kazandırır ve belirli bir alanda kullanılacak olan sözcük kümesini tanımlar.

Kaynakların sınıflar, özellikler ve değerler ile tanımlanmasını sağlayan sınıf kavramı, nesne tabanlı programlama dillerindeki sınıf kavramı ile benzerlik gösterir. Sınıflar arası kalıtım ilişkileri ve özellikler için kısıtlarının tanımlanmasına imkân verir. Bu özellikleri ile RDFS varolan ontoloji dillerine yakın imkanlar sunar, ancak RDFS’te sınıflar arası ayrıklık (disjoint), geçişlilik (transitive), tek olma (unique) gibi ilişkiler ve başka bir özelliğin tersi olma (inverse) gibi özellikler tanımlanamaz.

RDFS'in temel sınıf ve özellikleri aşağıdaki gibidir [11]:

- rdfs:Resource: RDF tarafından tanımlanan her şey kaynaktır.
- rdfs:Class: RDF sınıfı olan kaynakların sınıfıdır.
- rdfs:Literal: Tamsayı, metin gibi basit veri türlerini tanımlar.
- rdfs:Datatype: Veri türlerinin sınıfıdır.
- rdf:XMLLiteral: XML bilgisi içeren sabitleri tanımlamak için kullanılır.
- rdf:Property: RDF özelliklerini tanımlar.
- rdfs:range: Özelliğin değer aralığını ifade eder.
- rdfs:domain: Özelliğin tanım aralığını ifade eder.
- rdf:type: Kaynağın bir sınıfın bireyi olduğunu ifade eder.
- rdfs:subClassOf: Kaynak kalıtımını ifade eder.
- rdfs:subPropertyOf: Özellik kalıtımını ifade eder.
- rdfs:label: Kaynağın adını ifade eder.
- rdfs:comment: Kaynak hakkında açıklama.
- rdfs:Container: Kaynak topluluklarını ifade eder.
- rdfs:ContainerMembershipProperty: Kaynağın topluluğa aidiyetini belirtir.
- rdfs:member: Kaynağın ait olduğunu belirtir.
- rdfs:seeAlso: Kaynak hakkında ek bilgi veren bir diğer kaynağı işaret eder.
- rdfs:isDefinedBy: Özne kaynağı tanımlayan bir diğer kaynağı işaret eder.

## 2.6 OWL

OWL, Web Ontoloji Dili (Web Ontology Language) [12], RDFS'in eksikliklerini tamamlayan ve ilişkileri sınıflandırarak, anlamın ortaya konmasını sağlayan, düzenli bir yapı ortaya çıkarılmasına yardımcı olan bir ontoloji dilidir. OWL, RDFS'e ifade etme yeteneği ekler.

OWL, web kaynaklarında kullanılan kavramların, sınıfların ve özelliklerin anlamlarını, DL (Description Logic) kullanarak, ontoloji olarak ifade etmeyi sağlar. İfade etme detaylarına göre üç farklı versiyonu vardır; OWL-Lite en basit düzeyde

detay sağlar, OWL-DL ortalama düzeyde detay sağlar, OWL-Full ise en yüksek düzeyde detay ifade etmeyi sağlar [12].

Ontolojiler bir bilgi alanını tanımlamak için kullanılan sözcük kümesi olarak görülebilir. Yani bir ontolojiyi diğer insanlarla paylaşmak o alanla ilgili ortak bir anlayış geliştirmeyi sağlar. Örneğin FOAF (Friend Of A Friend)<sup>1</sup> ve SIOC (Semantically-Interlinked Online Communities)<sup>2</sup>; isim, adres, telefon ve kişisel web sayası gibi kişisel bilgileri tanımlayan ve dünyada en geniş kullanım alanı bulan ontolojilerdir. Bu ontolojileri kullanarak, bu ve ilgili diğer ontolojileri kullanan diğer uygulamalarla uyumluluk sağlanabilir.

## 2.7 SPARQL

SPARQL, Sparql Protokolü ve RDF Sorgulama Dili (Sparql Protocol And RDF Query Language) [13], RDF verileri üzerinde sorgulama yapılmasını sağlar. SPARQL sorgu dili olarak rdf/xml, turtle gibi formatları kullanabilir.

SPARQL 4 tip sorgu desteklemektedir;

- SELECT: Sorguda istenen şekilde, üzerinde çalışılan veri kümesinden istenilen değişkenlerin tamamını ya da bir kısmını döndürür.
- ASK: Veri kümesinde sorguyu karşılayan veri olup olmadığını döndürür.
- DESCRIBE: Sorguyla veya URI ile tanımlanan kaynağın, veri kümesi içindeki tanımını döndürür.
- CONSTRUCT: Sorguyu veri kümesi içinde arar ve yine sorguda belirtilen şablona uygun bir şema döndürür.

Şekil 2-5’de görülen SPARQL sorgusunda, ilk başta bulunan ön ek (prefix) bölümünde, sorgu içinde kullanılan isim uzayları için bazı kısaltmalar tanımlanır. Daha sonra sorgu türünü belirten SELECT anahtar kelimesi ve veri kümesi içinden

---

<sup>1</sup> <http://xmlns.com/foaf/spec/>

<sup>2</sup> <http://rdfs.org/sioc/spec/>

hangi deęişkenlerin getirilmesinin istendięini belirten deęişken isimleri verilir. WHERE cümlecigi ise veri kümesi üzerinde aranılan örüntüyü tanımlar.

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
      abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
      abc:isInContinent abc:Africa .
}
```

Şekil 2-5: SPARQL olarak “Afrika ülkelerinin başkentleri nelerdir?” sorusu [13]

### 3 ÇEVİRİMİÇİ ONTOLOJİ GELİŞTİRME ARAÇLARI

Ontolojiler genellikle karmaşık bilgi modelleridir. Bu modellerin geliştirme sürecinin zor olması ve modelin sorgulanabilmesi gerekliliği, geliştiricilere yardımcı olacak yazılım araçlarının kullanımını zorunlu kılmıştır. İşte bu zorunluluk sonucu 90'lı yılların sonuna kadar ortaya çıkan çeşitli ontoloji uygulamaları olmuştur. Ancak esas olarak bu alana ilgiyi artıran anlamsal web [14, 15] olmuştur. Bundan sonra, ontolojilerin kullanıcılar tarafından görsel olarak kolayca tanımlanmasını ve üzerinde sorgu çalıştırmayı sağlayan, ontoloji geliştirme araçları hızla ortaya çıkmaya başlamıştır.

Zamanla ilerleyen iletişim ve yazılım teknolojileri sonucunda ise artık günlük hayatta Web 2.0 platformları geniş ölçekte kabul görmüş ve Semantik Web teknolojileri daha çok benimsenir hale gelmiştir. Bunun sonucu olarak, kullanıcıların daha kolay etkileşmesini ve ortak anlayış oluşturabilmesini sağlayan, bilginin üretimi ve tüketimini akıcı hale getiren ontoloji geliştirme araçlarına gereksinim ve talep hiç olmadığı kadar artmıştır.

Bu dönemde değişen şeylerden biriyse, son kullanıcıların artık bir yazılımı bilgisayarlarına kurmak yerine web tabanlı içerik oluşturma ve çevrimiçi işbirliği sağlayan sistemleri yeğler duruma gelmesidir. Bu yeni nesil sistemler çevrimiçi çalıştığı için son kullanıcı tarafından kurulum veya bakım gerektirmez. Oluşturulan bilgi, internet üzerindeki bulut depolama sistemlerinde güvenli ve yedekli olarak saklanabilir. Birden çok kullanıcı eş zamanlı olarak veriye erişebilir.

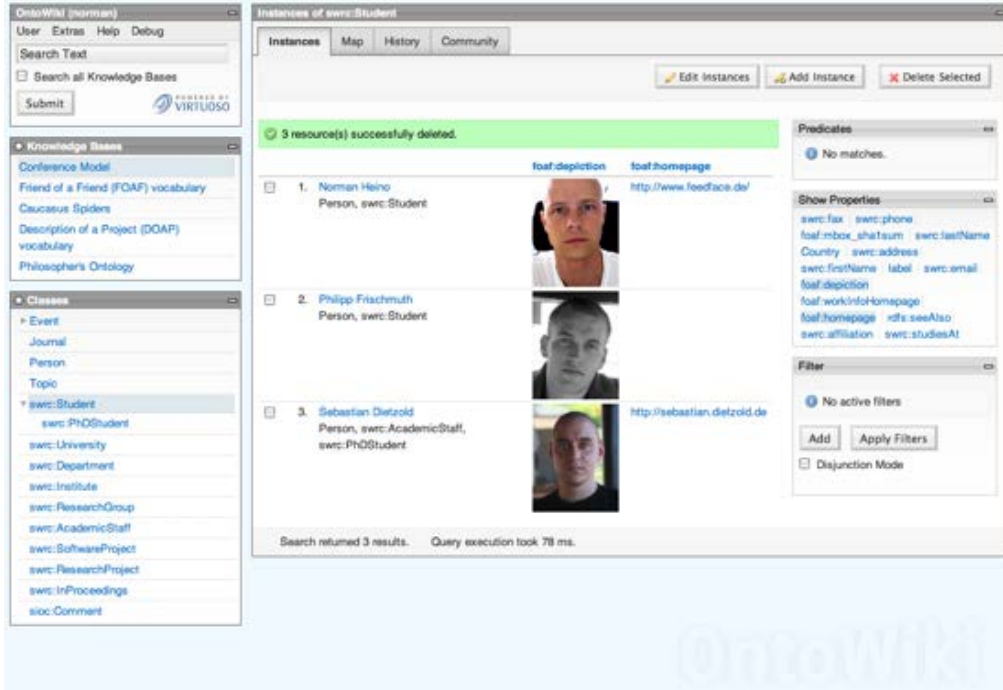
Takip eden alt bölümlerde, bu yeni nesil çevrimiçi ontoloji geliştirme araçlarından olan “OntoWiki”, “WebProtégé”, “MoKi” ve “TopBraid Ensemble” incelenecek ve temel özellikleri açısından karşılaştırılacaktır.

### 3.1 OntoWiki

Leipzig Üniversitesi bünyesinde, S. Auer, S. Dietzold, J. Lehmann ve T. Riechert tarafından 2007 yılında geliştirilmeye başlayan ve halen geliştirilmeye devam edilen OntoWiki [16]; çevik, dağıtık bilgi mühendisliği senaryoları için destek sağlayan bir araçtır.

Sınıfların, birey verisi üzerinde farklı görünümlere sahip bir bilgi haritasının görsel olarak sunumunu sağlar. Metin belgelerinde olan WYSIWYG düzenlemeye benzer olarak, RDF içerik için bir satır içi düzenleme moduyla beraber, Semantik içerik için bir sezgisel geliştirme ortamı sunar. Şekil 3-1’de örnek bir OntoWiki’ ekranı görülmektedir.

PHP dilinde yazılmış olan OntoWiki, Wiki ve Semantik Web paradigmalarını birleştirmeyi ve son kullanıcının, bilgi edinme ve düzeltme işlemlerini basitleştirip hızlandırmayı amaçlar.



Şekil 3-1: OntoWiki [17]

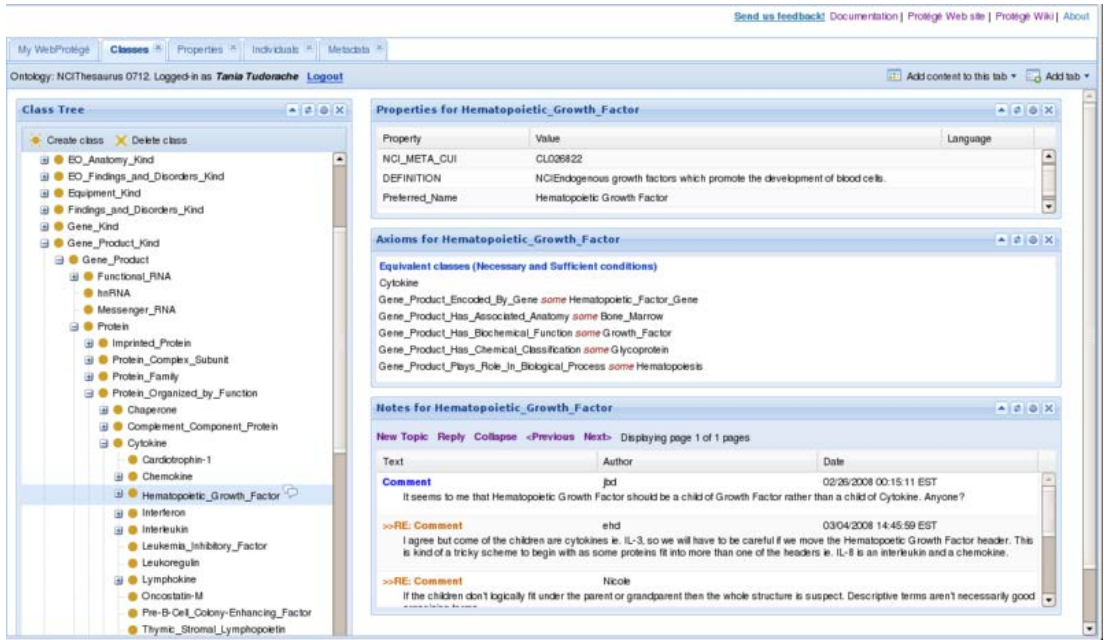


Ontolojilerin birçok kullanıcı tarafından ortaklaşa geliştirilmesini sağlayan OntoWiki, Ontolojilerin görsel bilgi haritası olarak gösterilmesini ve ilgili RDF içeriğin direkt olarak web sayfası üzerinde satır içi düzenlenmesini de imkan verir.

### 3.2 WebProtégé

Stanford Üniversitesi Tıp Fakültesi, Stanford Biyomedikal Bilişim Araştırma Merkezi'nde, Tudorache T., Vendetti J., Noy N. F. tarafından 2008 yılında geliştirilmeye başlanan ve halen geliştirilmeye devam edilen WebProtégé [18]; dağıtık bir ontoloji editörü ve web için bilgi edinme aracıdır.

WebProtégé, kullanıcıların ortak olarak ontoloji geliştirmesini sağlayan ve mevcut Protégé sisteminin bir uzantısı olan Collaborative Protégé [19] için bir web istemcisidir. Kullanıcılara gelişmiş bir web arayüzü vasıtasıyla, ontolojileri ortaklaşa olarak yaratmak ve tartışmak için ortam sağlar. Şekil 3-2'de örnek bir WebProtégé ekranı görülmektedir.



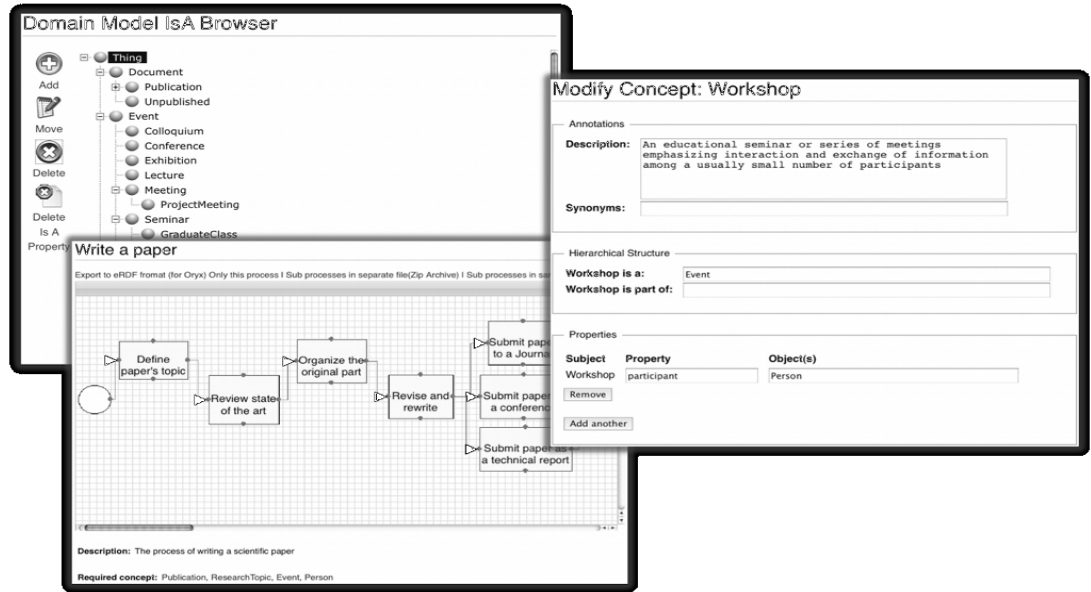
Şekil 3-2: WebProtégé [20]

WebProtégé Java dilinde yazılmıştır. Ortak geliştirme ortamı için Collaborative Protege'den modüller, Ontoloji servisleri için Protege kullanılmıştır. Kullanıcı arabirimi Google Web Toolkit (GWT) kullanılarak geliştirilmiştir ve yeniden kullanılabilir bileşenler olan, “portlet”lerden oluşmuştur. Genişletilebilir olduğu için uygulama geliştiriciler, kendi sekmeleri ve portletlerini kolayca yaratabilirler.

### 3.3 MoKi

Moki [21] (Enterprise Modelling WiKi), yapısal wiki sayfaları aracılığıyla ilişkisel kurumsal modellerin oluşturulmasına imkan veren bir Semantik MedyaWiki uzantısıdır. Moki, farklı bilgi mühendisliği becerileri olan, heterojen uzman ekiplerin, bilgiyi ekleyerek, dönüştürerek ve revize ederek, aktif işbirliği yapmalarını sağlar.

Aracın ilk versiyonu, AB projesi APOSDLE [22] içinde geliştirilmiştir. Geliştirilmesine DKM Araştırma Ekibi ve Know-Center tarafından ortak olarak devam edilmektedir. Şekil 3-3'te örnek bir MoKi ekranı görülmektedir.



Şekil 3-3: MoKi [23]

PHP dilinde yazılmış olan MoKi, Wiki ve Semantik Web paradigmalarını beraber kullanmaktadır. Ardındaki ana fikir, bir wiki sayfasını ontoloji ve süreç modelindeki varlıklarla ilişkilendirmektir. Tümüleşik bir etki alanı ve süreç modelleri yapımı için destek sağlayan MoKi, formlar vasıtası ile bir wiki sayfasının kolayca düzenlenmesini sağlar. OWL ve BPMN dillerinde otomatik içe ve dışa aktarmaya imkan verir. Öntanımlı anlamsal yapılara göre düzenlenmiş eleman listelerini kolayca içe aktarabilir. Etki alanı ve süreç modellerinin grafiksel gösterimi ve düzenlenmesini olanak tanırken, bütünüleşik model değerlendirme işlevleri sağlar.

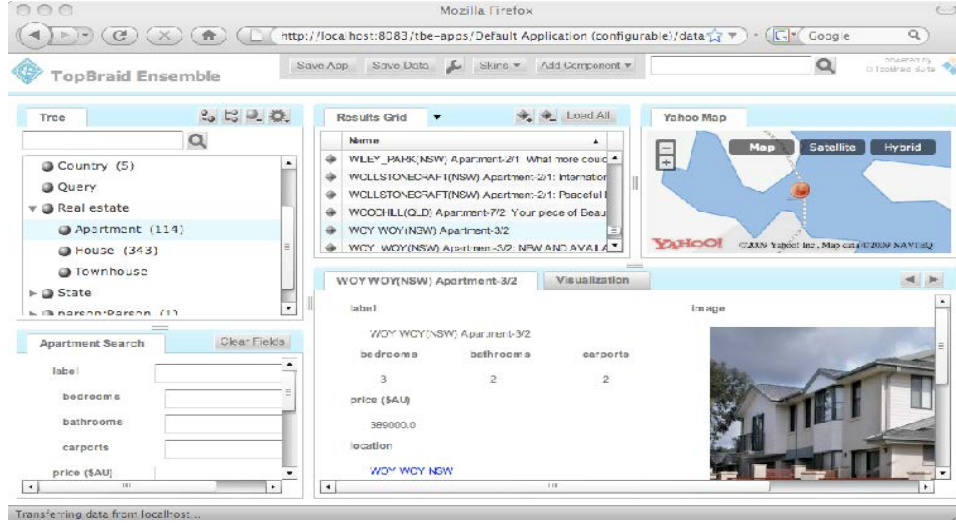
Kullanıcı arabirimi Yahoo User Interface (YUI) ve Semantic Forms kullanılarak geliştirilmiştir. Ontoloji, wiki ve içe/dışa aktarma işlemlerini gerçekleştirmek için, Mediawiki, Semantic Mediawiki ve ParserFunctions kullanılmaktadır.

### **3.4 TopBraid Ensemble**

TopBraid Ensemble [24], TopQuadrant Firması tarafından sunulan, semantik web uygulamaları için, çevik geliştirme ve uygulama birleştirme sağlayan, Adobe Flex tabanlı bir web uygulamasıdır.

SewIde’de olduğu gibi, bu aracında, kullanıcı arayüzü Adobe Flash yüklü web tarayıcılar üzerinde çalıştığı için tüm tarayıcılarda tutarlı bir kullanıcı arayüzü ve görünüm sunar. Uygulama geliştirildikten sonra yine aynı firmanın sunucu katmanı ürünü olan “TopBraid Live Enterprise Server” üzerinde çalıştırılabilmektedir.

Semantik modeller ve ilgili verileri üzerinde gezinmek ve düzenlemek için; düğme, ağaç (tree), ızgara (grid), form, harita, grafik görselleştirme, görsel sorgu oluşturucu arayüz bileşenleri içerir. Arayüz bileşenlerini yapılandırma ve ilişkilendirme işlemlerini kod yazmadan yapmak mümkündür. Sadece bir web tarayıcısı kullanarak kolayca bileşen, iletişim kutusu ve sayfa ekleyerek çok sayfalı uygulamalar oluşturulabilir. “SPARQLMotion” adı verilen betikleri kullanarak uygulamaları belirli çerçeveler içerisinde genişletmekte mümkündür. Şekil 3-4’te örnek bir TopBraid Ensemble ekranı görülmektedir.



Şekil 3-4: TopBraid Ensemble

Sistemin ölçeklenebilir çalışması ve hızlı yanıt verebilmesi için istemci tarafında RDF önbellek kullanılır. Ürünle beraber sunulan “Flex RDF API” kullanılarak, dış sistemler ve özel uygulamalarla entegrasyon sağlanabilir.

### 3.5 Değerlendirme

Yukarıda açıklamalarını verdiğimiz ontoloji geliştirme araçları oldukça yetkin olmakla birlikte bir kısım eksiklikleri de vardır. Bu araçların kullanıcıya; kapsamlı bir çalışma alanını ve ontoloji görünümünü kişiselleştirme, kaydetme, geri yükleme imkânı sunmadığını, ontoloji görünümünü asıl-detay ilişkisi içinde birbirine bağlama imkânı vermediğini, sadece bazı araçların kısıtlı ölçüde düzen ve yerleşim olarak değişiklik yapmaya izin verdiğini gördük.

İncelediğimiz araçlar arasında TopBraid Ensemble istisnai bir yer tutmaktadır. Bu ürün bizim hedeflerimizi büyük ölçüde gerçekleştirmekle birlikte ticari ve kapalı kaynak kodlu bir sistem olma dezavantajını taşımaktadır.

Bu eksiklikleri gidermek üzere, tez çalışmamızda prototip bir ontoloji geliştirme aracı geliştirdik. Gelecek bölümde bu aracı sunacağız ve mevcut araçlarla karşılaştırmasını yapacağız.

## **4 GELİŞTİRME ORTAMI**

Bir önceki bölümde çeşitli ontoloji geliştirme araçlarını inceledik ve bunların eksikliklerini ortaya koyduk.

Aşağıdaki bölümlerde ilk olarak, incelediğimiz sistemlerin özelliklerini ve eksikliklerini göz önüne alarak, SewIde'nin gereksinimlerini tanımlıyoruz. Bu gereksinimlerden yola çıkarak olası kullanım senaryolarını oluşturuyoruz.

Daha sonra, teknik gereksinimlerimiz ve kullanım senaryolarımız çerçevesinde SewIde'yi meydana getirecek alt sistemleri ve ilişkilerini tasarlıyoruz.

Son olarak ise SewIde'nin gerçekleştirilmesi aşamasında kullandığımız yazılım araçlarını, yazılım kütüphanelerini, tasarım örüntülerini, yazılım desenlerini kısaca anlatıyoruz ve gerçekleştirim aşamasının çeşitli aşamalarında karşılaştığımız sorunlara bulunan çözümleri açıklıyoruz.

### **4.1 Gereksinimler**

SewIde'nin ilk gereksinimi, uzakta bulunan bir RDF Veri Ambarının üzerinde işlem yapabilmesidir. Yani veri ambarında ontolojiler oluşturup üzerinde işlem yapılmasına izin vermelidir.

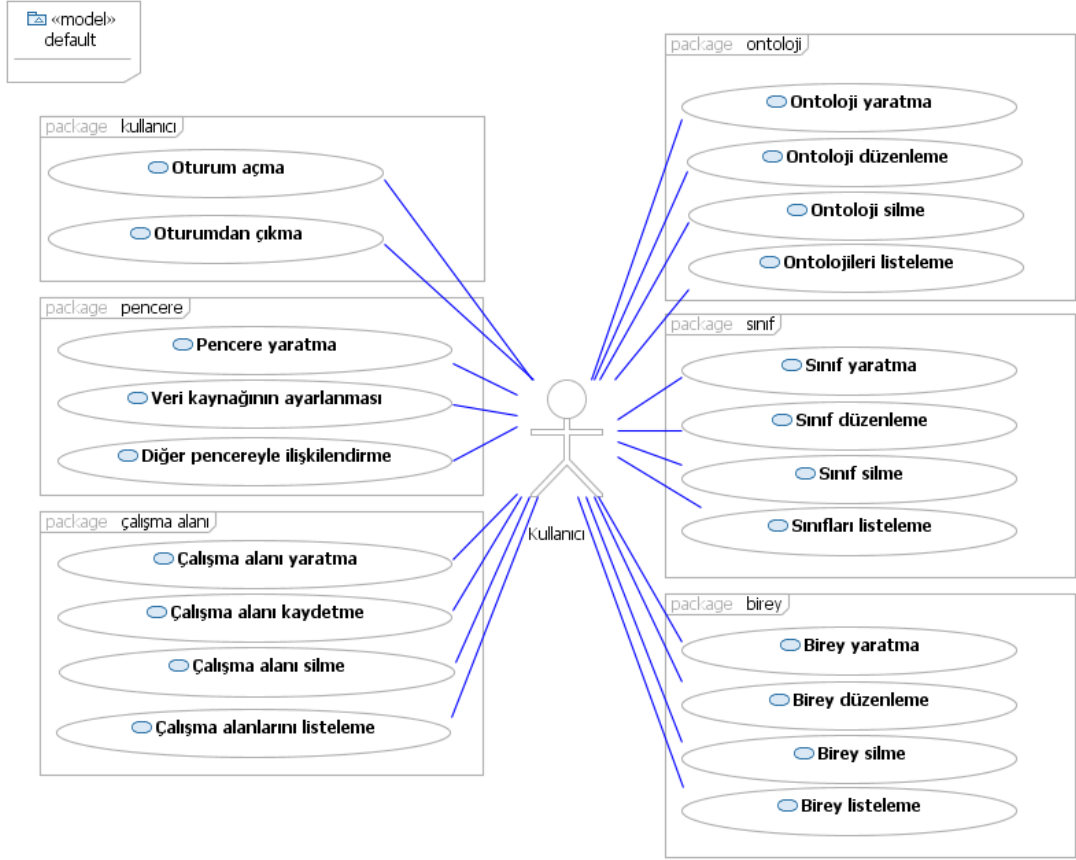
İkinci gereksinimimiz, kullanıcıların ontolojiler üzerinde eş zamanlı olarak çalışabilmesi için kullanıcı girişi ve oturum açma/kapama işlemlerini desteklemelidir.

Üçüncü gereksinimimiz, kullanıcıların çalışma alanlarını kişiselleştirip kaydedebilmesini ve daha sonra geri çağırıp kullanabilmesini sağlamaktır.

### **4.2 Kullanım Senaryoları**

Yukarıda anlattığımız gereksinimlere göre belirlediğimiz, gerçekleştirme işleminde göz önüne aldığımız işlemler ve kullanım senaryoları aşağıda listelenmiş ve kullanım

şekli (use case) diyagramı Şekil 4-1’de gösterilmiştir.



Şekil 4-1: SewIde Kullanım Şekli Diyagramı

### Senaryolar

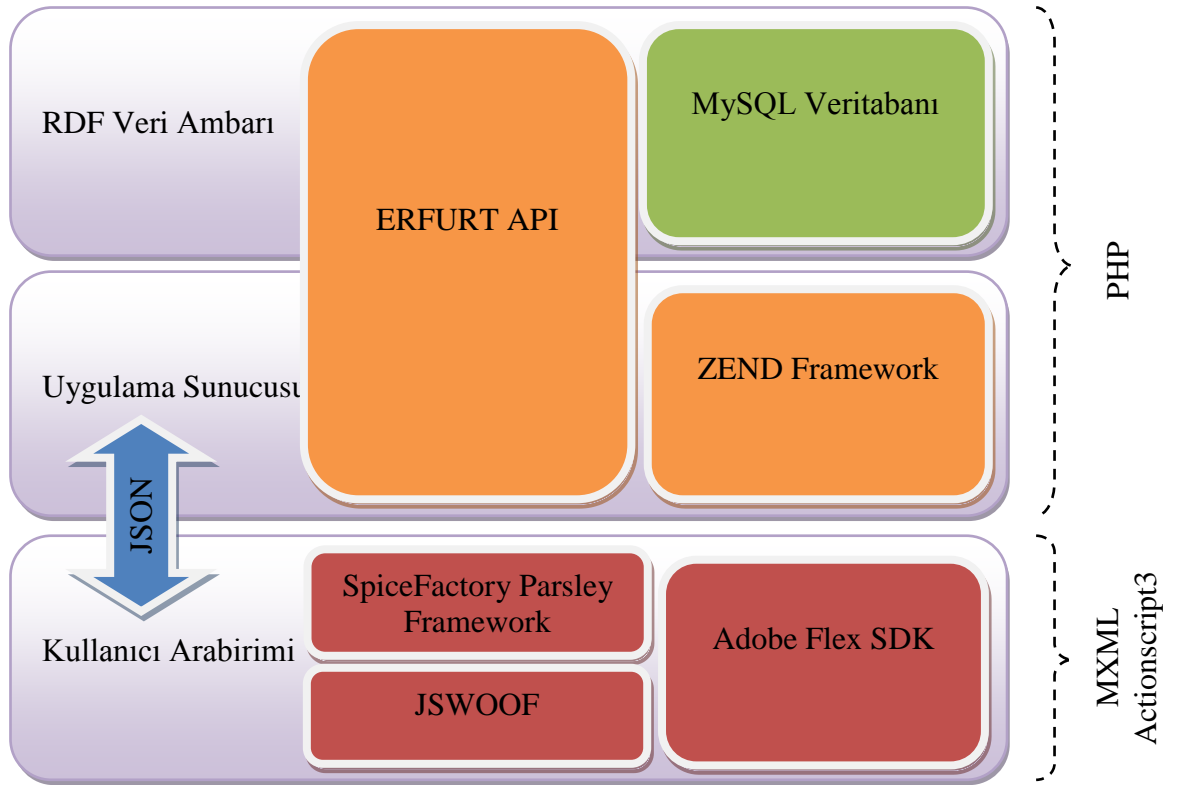
- Kullanıcı İşlemleri:
  - Kullanıcının sisteme giriş yaparak oturum açması
  - Kullanıcının geçerli oturumu kapatması
- Ontoloji İşlemleri:
  - Ontoloji yaratılması
  - Varolan ontolojinin değiştirilmesi
  - Varolan ontolojinin silinmesi
  - Sistemde kayıtlı ontolojilerinin listelenmesi
- Bir Ontolojiye ait Sınıf İşlemleri:
  - Sınıf yaratılması

- Varolan sınıfın deęiřtirilmesi
- Varolan sınıfın silinmesi
- Sistemde kayıtlı sınıfların listelenmesi
- Bir sınıfa ait Birey İşlemleri:
  - Biray yaratılması
  - Varolan bireyin deęiřtirilmesi
  - Varolan bireyin silinmesi
  - Sistemde kayıtlı bireylerin listelenmesi
- Kullanıcı arabirimdeki Pencere (Window) İşlemleri:
  - Yeni pencere yaratılması
  - Pencerenin “Veri Kaynaęının” ayarlanması
  - Pencerenin dięer pencereyle iliřkilendirilmesi
- Çalışma Alanı (Workspace) İşlemleri:
  - Çalışma alanı yaratılması
  - Geęerli çalışma alanının kaydedilmesi
  - Sistemde kayıtlı çalışma alanının silinmesi
  - Sistemde kayıtlı çalışma alanlarının listelenmesi

### 4.3 Tasarım

SewIde, “RDF Veri Ambarı”, “Uygulama Sunucusu” ve “Kullanıcı Arabirimi” olmak üzere 3 katmandan oluşmaktadır.

Adından da anlaşılacağı üzere “Kullanıcı Arabirimi” kullanıcının bilgisayarında, “RDF Veri Ambarı” ve “Uygulama Sunucusu” ise merkezi bir sunucu üzerinde bulunmaktadır. “Kullanıcı Arabirimi” ve “Uygulama Sunucusu” arasındaki veri alış veriři, JSON olarak kodlanmış veri paketleri yoluyla HTTP web servisleri řeklinde yapılmaktadır. řekil 4-2’de bu katmanlar ve iliřkileri grafiksel olarak görülebilir.



Şekil 4-2: SewIde Yazılım Katmanları

### RDF Veri Ambarı (RDF Triplestore)

Bu katman RDF üçlülerinin saklanması sağlayan katmandır. İlişkisel veritabanlarına bezer olarak, veri bu katmanda RDF üçlülerini olarak kaydedilir ve SPARQL gibi sorgu dilleri vasıtasıyla geri okunur. İlişkisel veritabanlarından farklı olarak, RDF Veri Ambarı, çok sayıda RDF üçlüsünün hızlı bir şekilde okunması için optimize edilmiştir. Virtuoso<sup>1</sup>, Joseki<sup>2</sup>, Redland<sup>3</sup>, Sesame<sup>4</sup>, RAP<sup>5</sup>, ARC<sup>6</sup> yaygın olarak kullanılan RDF Veri Ambarları arasında yer alır. Genelde RDF Veri Ambarı,

<sup>1</sup> <http://www.virtuoso.com>

<sup>2</sup> <http://www.joseki.org>

<sup>3</sup> <http://librdf.org>

<sup>4</sup> <http://www.openrdf.org>

<sup>5</sup> <http://www4.wiwi.fu-berlin.de/bizer/rdafapi>

<sup>6</sup> <http://arc.semsol.org>



RDF üçlülerini, fiziksel olarak, Oracle<sup>1</sup>, MySql<sup>2</sup>, DB2<sup>3</sup> gibi, bir ilişkisel veritabanı üzerinde tutar.

SewIde’de RDF Veri Ambarı desteğini sağlamak için, daha önce de bahsettiğimiz OntoWiki projesinin bir bileşeni olan “Erfurt API”nin RDF Veri Ambarı fonksiyonları kullanılmaktadır. Basit kurulumu ve kolay bakımı nedeniyle fiziksel üçlü deposu olarak MySql veritabanınının kullanılması tercih edilmiştir.

### **Uygulama Sunucusu**

Bu katman Kullanıcı Arabirimi’nden yollanan komutların işlendiği katmandır. Kullanıcı oturumlarını yönetir, RDF Veri Ambarına erişimi sağlar, kullanıcı isteklerine göre veri ambarını günceller veya sorgular.

### **Kullanıcı Arabirimi**

Bu katman bir web sayfası içerisine gömülmüş Adobe Flash Player sanal makinesi üzerinde çalışmaktadır. Temel işlevi kullanıcının yapmak istediği işlemleri, web servisleri yoluyla “Uygulama Sunucusu”na iletmek ve sunucudan gelen cevapları da görsel geribesleme olarak kullanıcıya iletmektir.

## **4.4 Gerçekleştirim**

Bu bölümde, geliştirme aşamasında kullandığımız yazılım araçlarını, kütüphaneleri ve temel yazılım kavramlarını açıklamaya çalışacağız.

80’li yılların sonundan itibaren çeşitli araştırmacılar konu üzerinde çalışmıştır, ancak Gang of Four (GoF) olarak bilinen, Eric Gamma, Richard Helm, Ralph Johnson ve John Vlissides tarafından yazılan “Design Patterns: Elements of Reusable Object-Oriented Software” [25] kitabı yayınlandığında tüm yazılım geliştiricilerin dikkatini bu kavrama çekmiştir.

---

<sup>1</sup> <http://www.oracle.com>

<sup>2</sup> <http://mysql.com>

<sup>3</sup> <http://www.ibm.com/software/data/db2/>

Tasarım örüntüleri, yazılım tasarımı sırasında karşılaşılan ve tekrar eden benzer sorunların üstesinden gelmek için yıllardır kullanılan çözümlerdir [26]. Desenler yıllar içerisinde yazılımcı ve araştırmacılar tarafından keşfedilmişlerdir ve belirli bir kişinin tasarımı değildir.

SewIde’de, Uygulama Sunucusunun çalışabilmesi için altyapı olarak, Apache<sup>1</sup> Web Sunucusu, PHP<sup>2</sup> Dinamic Dili ve MySQL<sup>3</sup> ilişkisel veritabanı altsistemlerinden oluşan bir paket kullanılmaktadır. Bütün bu altsistemleri taşınabilir (portable) olarak Windows işletim sistemiyle çalışan bilgisayarlarda, kurulum yapmadan çalıştırmayı sağlayan “Uniform Server”<sup>4</sup> adlı paketin 5.6b-Nano sürümü kullanılmıştır.

Aşağıdaki bölümlerde, gerçekleştirim safhasında kullandığımız web sunucusu, veritabanı, programlama dilleri, kod geliştirme ortamları, yazılım kütüphaneleri, tasarım örüntüleri, yazılım desenleri kısaca anlatılacak. Son bölümlerde ise gerçekleştirim aşamasının çeşitli aşamalarında karşılaşılan sorunlar ve bu sorunları çözmek için nasıl bir yol izlendiği anlatılacaktır.

## **Apache HTTP Sunucusu**

Apache birçok işletim sistemi üzerinde çalışabilen, açık kaynak kodlu bir web sunucu yazılımıdır. 1996'dan beri en yaygın kullanılan web sunucusudur. Netcraft'ın Ocak 2012 raporuna göre 582,716,657 web sitesinin yüzde %65.22'si Apache kullanmaktadır [27].

Kolay kurulum, dokümantasyon ve yaygınlığı göz önüne alınarak, SewIde’de Apache 2.2.15 sürüm HTTP sunucusu öntanımlı ayarlarıyla kullanılmıştır.

---

<sup>1</sup> <http://apache.org>

<sup>2</sup> <http://php.net/index.php>

<sup>3</sup> <http://mysql.org>

<sup>4</sup> <http://www.uniformserver.com>

## **Zend PHP / Zend Framework**

PHP aslen dinamik web sayfaları üretmek için tasarlanmış genel amaçlı bir sunucu betik dilidir. PHP, Rasmus Lerdorf tarafından, kişisel web sayfasını yönetmek üzere yazılan bir dizi PERL betiği olarak doğmuştur. Orijinal adı “Personal Home Page” iken artık “PHP: Hypertext Preprocessor”, yinelemeli kısaltması olarak anılmaktadır.

“Günümüzde PHP, bloglardan forumlara, portal sistemlerinden veri tabanlarına, yapay sinir ağlarından düşünen sistemlere, sınıflardan fonksiyonlara her türlü işlevde kullanılır” [28].

Zend Framework, ise Zend Firması tarafından basitlik, nesne yönelimli geliştirme desteği, en iyi uygulamalar göz önüne alınıp PHP kullanılarak geliştirilmiş ve titizlikle test edilmiş kod tabanından meydana gelmiş bir yazılım çerçevesidir.

Zend Framework’deki en çok kullanılan bileşenler arasında; Model-View-Controller (MVC) bileşenleri, Hızlı Uygulama Geliştirme (RAD), Veritabanı, Uluslararasılaştırma ve Yerelleştirme, Kimlik doğrulama, yetkilendirme ve oturum yönetimi, Web Servisleri, E-Posta, PDF, Arama sayılabilir.

SewIde’de PHP 5.3.3 sürümünü kullanıyoruz. PHP ayarlarına, NuSphere PhpED’e uyum sağlayabilmesi için gerekli olan 1 satırlık bir ayar eklemesi yapılmıştır.

## **MySQL Veritabanı**

MySQL, ile ilgili olarak Wikipedia web sayfasında sistem şöyle tanıtılmaktadır [29]:

“MySQL, altı milyondan fazla sistemde yüklü bulunan çoklu iş parçacıklı (multi-threaded), çok kullanıcı (multi-user), hızlı ve sağlam (robust) bir veritabanı yönetim sistemidir.

UNIX, OS/2 ve Windows platformları için ücretsiz dağıtılmakla birlikte ticari lisans kullanmak isteyenler için de ücretli bir lisans seçeneği de mevcuttur. Linux altında daha hızlı bir performans sergilemektedir. Kaynak kodu açık olan MySQL’in pek çok

platform için çalıştırılabilir ikilik kod halindeki indirilebilir sürümleri de mevcuttur. Ayrıca ODBC sürücüleri de bulunduğu için birçok geliştirme platformunda rahatlıkla kullanılabilir.

Web sunucularında en çok kullanılan veritabanıdır. ASP, PHP gibi birçok web programlama dili ile kullanılabilir.

MySQL'in 4.0 sürümü ile birlikte "transaction" desteği, 4.1 sürümüyle birlikte de alt sorgu desteği eklenmiştir."

SewIde'de, MySQL sürüm 5.1.44 veritabanı sunucusu öntanımlı ayarlarıyla kullanılmıştır.

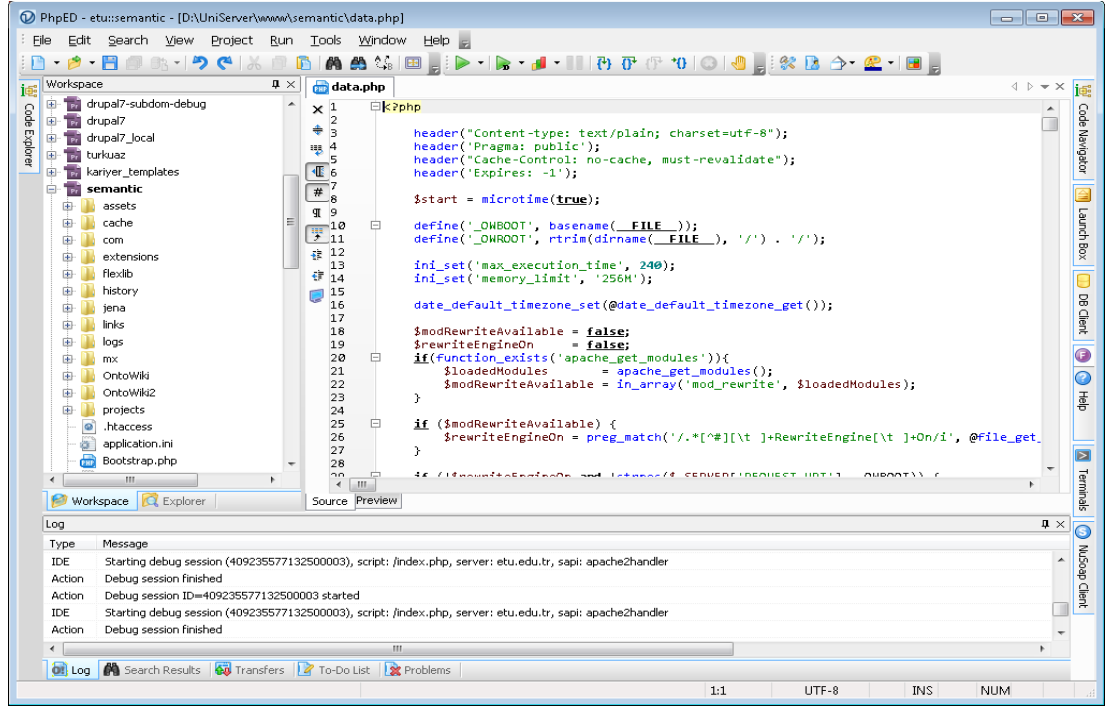
### **NuSphere PhpED**

Şekil 4-3'te örnek bir ekranı görülen PhpED<sup>1</sup>, hem bireysel hem de çok-geliştiricili büyük projeler için rahatça kullanılabilen, PHP için en iyi bütünleşik geliştirme ortamlarından birisi. Web siteleri ve web uygulamaları geliştirmek için tam takım araçlar sunuyor:

- HTML 3.2, 4.0, xhtml ve 5.0 standartları, PHP 4.2 - 5.3 sürümleri desteği,
- Dinamik sözdizimi vurgulama, otomatik değişken vurgulama, kod katlama,
- Otomatik kod tamamlama, anlık hata düzeltme,
- Uzak ve yerel kod üzerinde hata ayıklama ve kod zamanlaması,
- SQLite, MySQL, MSSQL, Oracle, PostgreSQL, InterBase destekli veritabanı gezgini,
- Telnet, SSH, FTP, SFTP, WebDAV destekli terminal ve dosya gezginleri,
- PHP, HTML, CSS için bağlam duyarlı yardım.

---

<sup>1</sup> <http://www.nosphere.com>



Şekil 4-3: NuSphere PhpED

## Adobe Flex SDK / Adobe Flash Builder

Adobe Flex SDK'nın tarihsel gelişimi ile ilgili olarak Wikipedia web sayfasında sistem şöyle tanımlanmaktadır [30]:

“Çoklu platform tabanlı zengin İnternet uygulamalarının Adobe Flash platformu üzerinde geliştirilmesi ve dağıtılması amacıyla Adobe Systems tarafından yayımlanan bir yazılım geliştirme kitidir.

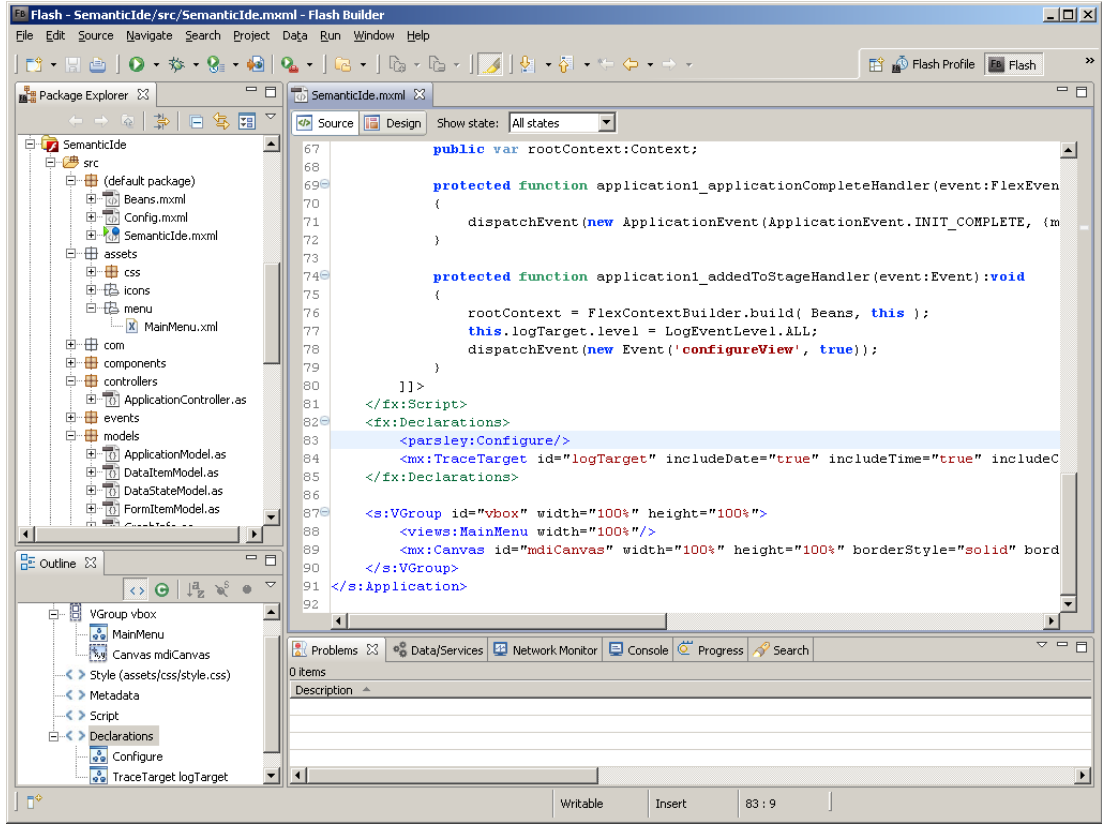
Flex uygulamaları, Adobe Flash Builder kullanılarak ya da Adobe'den ücretsiz olarak kullanılabilen Flex derleyicisi kullanılarak yazılabilir. İlk sürümü Mart 2004'te bir yazılım geliştirme kiti, bir IDE ve Flex Data Services olarak bilinen bir Java EE entegrasyon uygulaması olarak Macromedia tarafından yayınlandı. 2005 yılında Adobe Systems, Macromedia'yı satın almıştır. Şubat 2008'de Adobe, açık kaynak Mozilla Kamu Lisansı altında Flex 3 SDK'yı yayınladı ve böylece Flex uygulamaları, herhangi bir standart IDE kullanılarak geliştirilebilir hale geldi. Ayrıca 'Adobe Flash Builder' [31] denilen ticarî ve tescilli bir IDE vardır.”

Flex SDK, düğme, metin, liste kutusu, ağaç, çizelge, grafik gibi kullanıma hazır çeşitli kullanıcı arayüzü bileşenlerinin yanında, sürükle-bırak, form doğrulama, durum geçişleri, animasyonlu efektler, diyaloglar gibi diğer özellikleri de desteklemektedir. Yerel ve uzak konumlardaki veriye erişim içinse http, rtmp, web hizmetleri gibi iletişim yöntemlerini desteklemektedir.

MXML, XML tabanlı bir işaretleme dilidir. Kullanıcı arabirimlerinin kolayca oluşturulması ve arayüzdeki durum geçişlerini belirtmek için kullanılır. MXML kodu, ECMAScript standardına dayanan ActionScript3 diline dönüştürülür. Actionscript3 ise Flash Player'da bulunan AVM2'de (ActionScript Virtual Machine 2) çalıştırılabilecek ABC (ActionScript Bytecode) koduna derlenir.

Çok katmanlı bir web uygulamasında, Flex arayüzlerini görüntü katmanı olarak kullanarak, sunucu tarafından tüm sayfayı değil sadece arayüzde gösterilecek veri istenir. Bu sayede hem veri iletişim hacmi büyük ölçüde azaltılır hem de sunum ve veri işleme rolleri net bir şekilde ayrılmış olur.

SewIde'de kullanıcı arabiriminin oluşturulmasında, Şekil 4-4'te örnek bir ekranı görülen Adobe Flash Builder 4.5 ve Adobe Flex SDK 4 [32] kullanılmıştır. MXML ve Actionscript 3.0 dilinde yazılan arayüz, daha sonra Flash uygulaması olarak derlenip bir HTML sayfasına gömülerek, Uygulama Sunucusu'nun üzerine yüklenmiştir.



Şekil 4-4: Adobe Flash Builder 4.5

## Erfurt API / OntoWiki

SewIde'nin, sunucu tarafında çalışan RDF Veri Ambarı ve Uygulama Sunucu katmanlarında, RDF Veri Ambarı işlemlerini gerçekleştirmek, SPARQL sorgularını çalıştırmak ve kullanıcı oturumlarını yönetmek için, PHP'de yazılmış olan Erfurt API [33] kullanılmıştır. Erfurt API'nin sağlamadığı bazı yüksek seviyeli işlevler için, OntoWiki uygulamasının bazı kısımları aynen ya da küçük değişiklikler yapılarak kullanılmış ve işleve özel SPARQL sorgusu yazılmıştır. Şekil 4-5'de temel Erfurt paketleri görülmektedir.



Şekil 4-5: Temel Erfurt Paketleri

SewIde'nin kalbi olan Uygulama Sunucusu, başlangıç aşamasında, kullanılan yazılım kütüphanelerinin yerlerini belirler, kullanıcı oturumlarını ayarlar, önbellekleme için gerekli ayarları yaptıktan sonra, kullanıcı tarafında çalışan arayüzden JSON formatında gelen komutları ve parametrelerini işleyerek gerekli işlemleri yerine getirir.

### **Model-Görünüm-Denetleyici Örüntüsü (Model-View-Controller Pattern)**

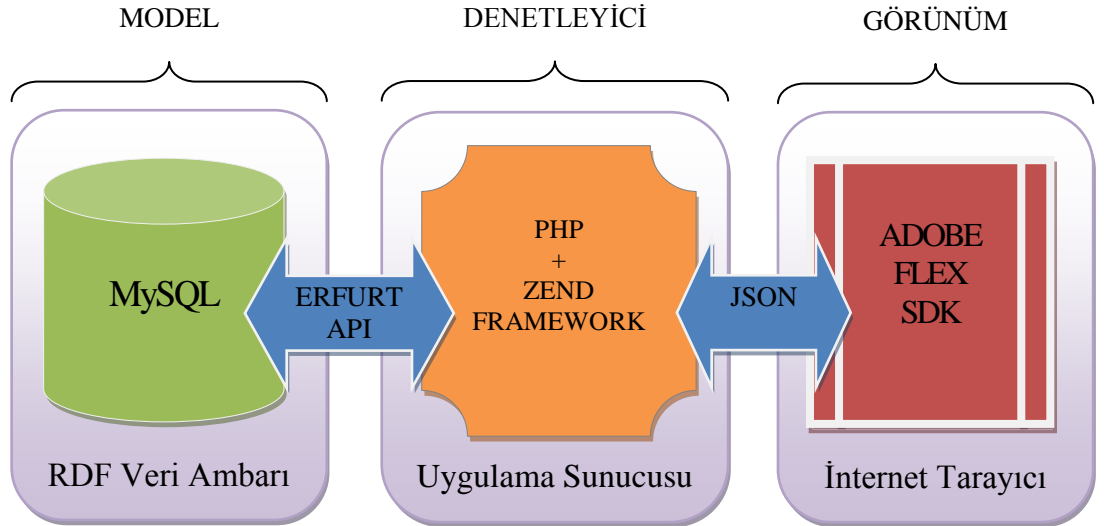
MVC (Model-View-Controller) örüntüsü, veri ve gösterimin birbirinden soyutlanması, verinin (model) ve kullanıcı arayüzünün (view) birbirini etkilemeden düzenlenebilmesi, esasına dayanır. Veri erişimi, iş mantığı ve asıl gerçekleştirim süreçleri ise denetleyici (controller) bileşeninde gerçekleştirilmek suretiyle diğer bileşenlerden soyutlanmaktadır.



Bu desen ilk defa 1979'da Trygve Reenskaug [34] tarafından tanımlanmıştır. Özgün gerçekleştirimi detaylı olarak anlatan Steve Burbeck tarafından kaleme alınmış bir de makale vardır [35].

SewIde Model-Görünüm-Denetleyici yapısı Şekil 4-6'te görülmektedir. SewIde görünüm (view) gerçekleştirimi Adobe Flex kullanılarak hazırlanmıştır. Kullanıcılar yapmak istedikleri işlemleri, arayüz öğeleri vasıtasıyla uygulama sunucusunda konumlandırılmış PHP'de yazılmış olan denetleyiciye (controller) iletmektedir. Denetleyici ise Erfurt-Api aracılığıyla RDF veri ambarından SPARQL sorguları vasıtasıyla elde ettiği verileri (model) arayüzde gösterilmek üzere geri yollamaktadır.

Nesne ve dizi gibi yüksek seviyeli veri öğelerinin aktarımına izin verdiği için, kullanıcı arayüzü ile uygulama sunucusu ile iletişimde JSON formatı tercih edilmiştir. Flash Player ancak 11. sürümden itibaren JSON desteği sunmaya başlamıştır, ancak şu anda en yaygın olarak kullanılan Flash Player sürüm 10'da JSON desteği sağlamak için "JSwoof - JSON Parser For Flex" [36] kütüphanesi kullanılmıştır.



Şekil 4-6: SewIde Model-Görünüm-Denetleyici yapısı

## **Bağımlılık İletimi Deseni (Dependency Injection Pattern)**

DI (Dependency Injection) örüntüsü, IoC (Inversion of Control) prensibinin bir uygulamasıdır ve bir yazılım bileşenine ihtiyacı olduğu gereksinimini sağlama esasına dayanır. Bu terim ilk defa, sistemin çalışma prensibini daha açık bir şekilde anlatmak için Martin Fowler tarafından kullanılmıştır [37].

Geleneksel olarak, bir yazılım nesnesi, yapacağı işi gerçekleştirmek için bir hizmete ihtiyaç duyuyorsa, o hizmet nesnesini yaratmak ve işi bitince yok etmek de onun sorumluluğudur. Bu tür bir yazılım geliştirme metodu izlemek geliştirilen yazılımın karışık ve takibi zor bir hal almasına neden olmaktadır. Halbuki, bu nesne sadece ihtiyaç duyduğu bu hizmet nesnesine erişip, istediği metotlarını çağırması, servis nesnesinin hayat döngüsüne karışmamalıdır. DI örüntüsü işte tam bu noktada imdadımıza yetişmekte ve servis nesnesinin hayat döngüsünün yürütülmesini üstlenmektedir.

Kullanıcı Arabirimi, MVC örüntüsü kullanılarak oluşturulmuştur. Veri modelleri, denetleyici tarafından, Uygulama Sunucusu'ndan gerçek zamanlı olarak çekilip, görüntü katmanına iletilmektedir. Kullanıcıların görüntü katmanıyla etkileşimini, görüntü katmanındaki bileşenlerin birbiri ile olan bağlantılarını ve görüntü katmanının denetleyici katmanında bulunan uzak hizmetlere erişimini kolaylaştırmak için kullandığımız DI örüntüsünü “Spicefactory Parsley Framework” [38] sağlamıştır.

Görüntü katmanında DI örüntüsünün kullanılması birbirine bağımlılığı olmayan, tekrar kullanılabilir ve modüler bileşenlerin tasarımına izin vermiştir.

## **Model Güdümlü Mimari (Model Driven Architecture)**

Bilişim teknolojilerinde görülen hızlı gelişme, yazılım sistemlerinin de giderek karmaşıklaşmasına neden olmaktadır. Bu karmaşıklıkla başa çıkmak için yazılım mühendisliği alanında öne sürülen yazılım geliştirme yaklaşımlarından birisi MDD

[39] (Model Driven Development) dir. MDD, farklı soyutlama seviyelerindeki modelleri kullanarak yazılım geliřtirmedeki karmařıklığı azaltmayı hedeflemektedir.

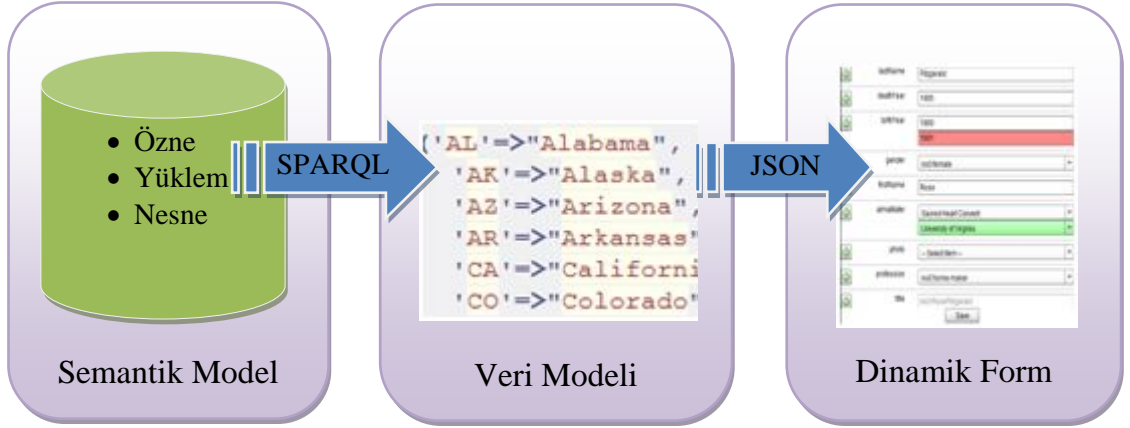
MDD, çoklumodellerin (metamodel) tanımlanmasını ve bu çoklumodeller arasında dönüşümler sağlanarak, sistem modellerinin soyuttan somuta doğru dönüřtürülmesini sağlar. Bu süreç, soyut modellerin otomatik olarak eşlenmesi için kaynak modellerin üzerinde gerekli bir dizi dönüşüm kuralı uygulandıktan hedef modellerin elde edilmesi şeklinde gerçekleşir. Literatürde, MDD'nin derleyicilerden bu yana yazılım teknolojisindeki ilk gerçek paradigma deęiřiklięi olduęu öne sürölmektedir [39].

MDA (Model Driven Architecture) [40] ise, OMG'nin (Object Management Group), MDD yaklaşımının bir gerçekleřtirmesi olarak, 2001 yılında bařlattığı, yazılım sistemlerinin tasarımı yaklaşımıdır.

MDA, soyut yazılım modellerini, gerçek çalıştırılabilir uygulamalara dönüřtürerek yazılım sistemlerinin geliřtirilmesini amaçlar. MDA içerisinde, MOF (Meta-Object Facility) [41] çerçevesini esas alan birçok model dönüşümünü tanımlanmıştır.

MDA'de modeller yazılım geliřtirme sürecinin hemen her aşamasında kullanılmaktadır. Farklı ortamlara entegrasyonu sağlamak amacıyla MDA, modelleri MOF temelli bir dil ile ifade edilir, MOF uyumlu depolarda saklanır, MOF uyumlu araçlar tarafından incelenebilir ve gerektiğinde de XMI'ya (XML Metadata Interchange) çevrilerek bir aę üzerinde iletilir [40].

SewIde'de, Şekil 4-7'de göröldüğü gibi, MDA'de kullanılan [Model] → [Kod] dönüşüne benzer olarak [Semantik Model] → [Veri Modeli] → [Dinamik Form] dönüşümü kullanılmaktadır. Yani, RDF veri ambarında saklanan özne-yüklem-nesne (subject-predicate-object) üçlüleri uygulama sunucusu tarafından sorgulanıp, görüntü katmanına iletilebilecek bir veri biçime dönüřtürölmekte, orada ise gerçek zamanlı bir dönüşümden geçirilerek kullanıcının işlem yapmasına imkân verecek arayüz öğeleri olarak ifade edilmektedirler.



Şekil 4-7: SewIde'nin Model GÜdümlü Mimari ile benzerliği

MDA'de kullanılan modellere paralel olarak semantik modeller özne-yüklem-nesne üçlüleri olarak ifade edilir, RDF uyumlu depolarda saklanır, SPARQL destekli araçlar tarafından sorgulanabilir ve RDF/XML formatında ağ üzerinde iletilebilir.

### Performans ve Önbellek

SewIde'de performansı artırmak için 3 seviyede önbellek kullanılmaktadır.

Birinci seviyede, Erfurt API tarafından çalıştırılan SPARQL sorgularının sonuçlarının veritabanı üzerinde kaydedildiği önbellek bulunmaktadır. Bu önbellek sayesinde zaman ve işlem gücü maliyeti yüksek olan sorgular hızlı bir şekilde cevaplanabilmektedir.

İkinci seviyede, uygulama sunucusunun RDF veri ambarından aldığı sonuçları kaydettiği önbellek bulunmaktadır. Bu önbellek verilerini geçici dosyalarda tutar ve oldukça basit tasarlanmıştır. Amacı tek kullanıcı senaryolarında daha fazla hız sağlamaktır. Önbellek üzerinde herhangi bir kilitleme vb. bir önlem konulmadığı için çok kullanıcı senaryolarında hatalı sonuçlara neden olacaktır ve bu haliyle kullanılmamalıdır.

Üçüncü seviyede, kullanıcı arayüzü bileşenlerinin de bulunduğu görüntü katmanında bulunan önbellek bulunmaktadır. Bu önbellek, arayüzde hızlı ve akıcı bir kullanıcı deneyimi sunmak, değişmeyen bilgileri elde tutarak ağ trafiğini azaltmak ve dinamik

olarak oluşturulan formlarda kullanılan birleşik giriş kutusu (combobox) vb gibi liste barındıran arayüz öğelerini hızlı bir şekilde doldurmak amaçlarını gütmektedir. Çok kullanıcı senaryolarında farklı kullanıcılarda farklı bilgilerin görülebilmesi bu önbelleğin neden olabileceği en önemli yan etkidir.

### **Çok Kullanıcı Erişim**

Birden çok kullanıcının ontolojiler üzerinde ortak çalışabilmesi temel gereksinimlerimizden birisidir. Ancak bu özellik hem görüntü katmanında hem de veri katmanında birçok karmaşıklık beraberinde getirmektedir. Bu konuda ortaya çıkan problemleri daha detaylı irdelemeyi uygun buluyoruz.

Temel sorunlardan ilki, iki veya daha fazla kullanıcının ontolojiler üzerinde aynı anda yapısal değişiklik yaptığında ortaya çıkan problemlerdir. Bu sorunun üstesinden gelmek için bir çeşit kilitleme mekanizması kullanmak gerekmektedir. Bu soruna bizim getirdiğimiz çözüm, her bir veri modeli işlemini bir bütün olarak tablo kilitlemeye dayalı bir veritabanı hareketi (database transaction) çerçevesinde atomik olarak gerçekleştirmektir. Bu sayede kullanıcılar, yapılan değişikliklerden ötürü meydana gelen çakışmalardan etkilenmezler. Bu durumda bir veri tutarsızlığı problemiyle karşılaşılmaz ve yapılan değişiklikler sırayla işlenir. Ancak sadece en son değişikliği yapan kullanıcı verinin tam halini görür, diğer kullanıcı ise sadece kendi yaptığı değişiklikleri görecektir. Bu çakışma senaryosu bir önceki önbellek konusunda anlatılan ikinci seviye önbellek kullanılması durumunda daha da karışık bir hal alacaktır.

Diğer bir sorun ise, farklı kullanıcıların yaptıkları değişiklikleri birbirini etkilemeden kaydedebilmesi gerekliliğidir. Bu soruna getirilen çözüm ise, kullanıcıların arayüzde yaptıkları kişiselleştirmeleri bir çalışma alanı (workspace) çerçevesinde kaydedip geri yüklemelerine olanak vermek şeklinde olmuştur. Ancak SewIde, bu haliyle, kaydedilen çalışma alanları üzerinde bir sahiplik veya yetki denetimi yapmamaktadır.

## PHP ve Java Semantik Web Kütüphaneleri

Bu bölümde, SewIde'yi geliştirmeye başlamadan önce deneyip, özelliklerini araştırdığımız çeşitli SW kütüphaneleriyle olan deneyimlerimizi kısaca aktaracağız.

Geliştirme aşamamızın bu erken döneminde, ilerleyen aşamalardaki teknik ihtiyaçlarımızı karşılayabilecek kadar kapsamlı, dokümantasyonu ve desteği iyi olan bir SW kütüphanesini seçmeye çalıştık.

SewIde'de PHP kullanmayı düşündüğümüz için ilk olarak PHP'de yazılmış olan RAP<sup>1</sup> , ARC<sup>2</sup> , ve OntoWiki/Erfurt<sup>3</sup> SW kütüphanelerini daha sonra ise Java'da yazılmış olan JENA<sup>4</sup> kütüphanesini inceledik.

RAP, RDF modelleri için sorgulama, düzenleme, dizi olarak sunma ve ağ üzerinden sunma işlemlerini yapabilen bir yazılım kütüphanesi. Diğer birçok SW kütüphanesi için temel oluşturmuş. API olarak JENA'ya çok benziyor bu yüzden kolayca anlaşılabilir. Sunduğu özellikler açısından kapsamlı ancak, son sürümü Şubat 2008'de yayınlanmış ve aktif olarak geliştirilmiyor.

ARC, eğer sadece temel seviyede özellikler kullanılacaksa ve daha yüksek ve soyut seviyede API kullanılmayacaksa az hafıza harcayan ve hafif sıklet bir kütüphane olarak tercih edilebilir. Aksi durumda diğer kütüphaneler tercih edilmelidir.

JENA ise akademik çevrelerce çokça kullanılan, daha çok RDF(S) odaklı, gayet sağlam, ancak düşük seviye işlemler sağlayan bir kütüphane.

Yukarıdakilere ek olarak, "Sistem gerçekleştirimini PHP yerine Java'da yaparak ne gibi bir avantaj sağlanır?" sorusunu da irdeledik.

---

<sup>1</sup> <http://www4.wiwiw.fu-berlin.de/bizer/rdfapi>

<sup>2</sup> <http://arc.semsol.org>

<sup>3</sup> <http://ontowiki.net/Projects>

<sup>4</sup> <http://jena.sourceforge.net>

Sonuç olarak, ařađıdaki ıkarımlara ulařtıđ:

- PHP'nin Java'ya oranla daha kolay ve yaygın olduđu,
- SW ile alıřmayı mmkn kılan temel araların ezici bir ođunluđunun Java tabanlı olduđu ancak PHP tabanlı ktphanelerinde yeterli olabildiđi,
- JENA'ya kıyasla PHP tabanlı ktphanelerin đrenme eđrisinin ok daha alak olduđu ve daha yksek ve soyut seviyede API sađladıđı,
- Java kodunun PHP kodundan daha hızlı alıřtıđını, ancak PHP kodunun daha az hafıza ihtiyaı olduđu,
- Temel gerekleřtirmeyi PHP'de yapıp SW iřlemleri JENA iin ktphanesini kullanmak gibi bir alternatif olduđunu ancak hem PHP hem de Java kod tabanını ynetmenin zor olduđu,
- İhtiya duyduđumuz bazı yksek seviyeli iřlemleri SPARQL sorgularıyla almanın daha kolay olduđu (rn: bir sınıfa ait tm bireylerin tm zellikleri).

Bu ıkarımlara ve tecrbelere dayanarak SewIde'de SW ktphanesi olarak kullanmak zere Erfurt/OntoWiki paketini seđtik.

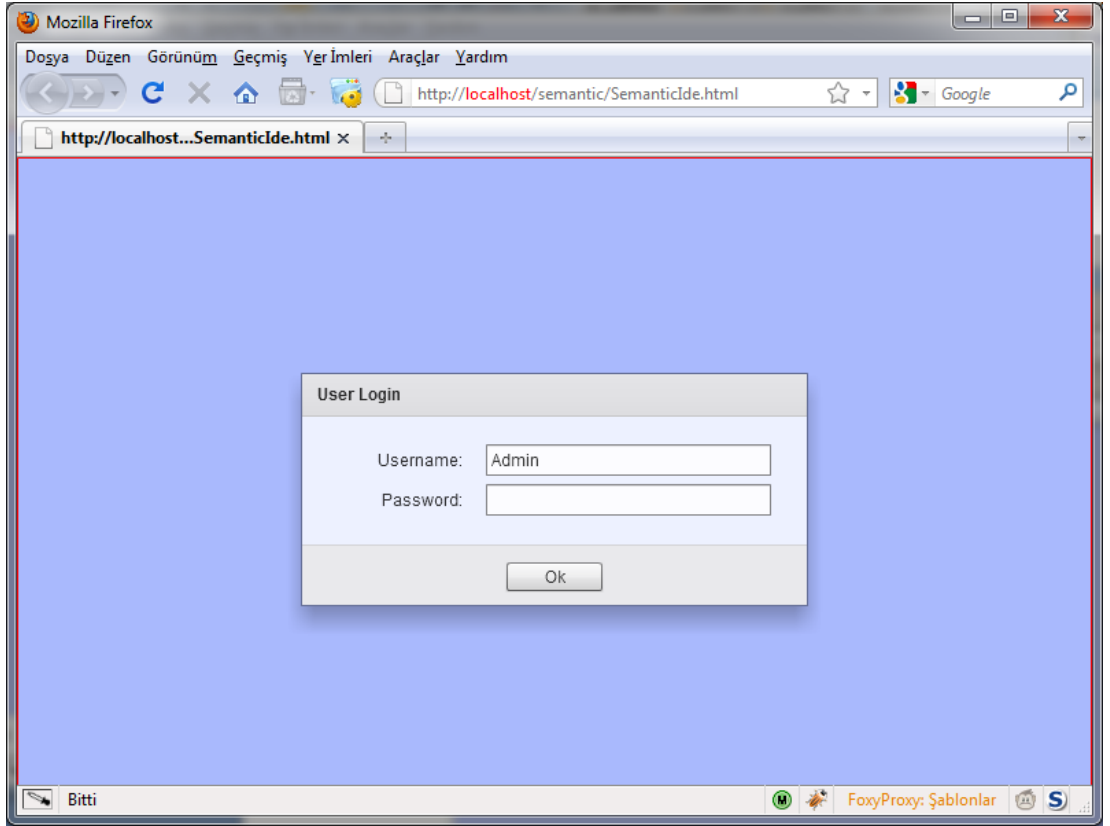
## 4.5 SewIde Kullanımı

Bu bölümde, belirlediğimiz kullanım senaryolarımızın uygulama arayüzleri eşliğinde anlatacağız.

### Kullanıcı Giriş

Bu ekranda;

- Kullanıcı oturumu açılır.



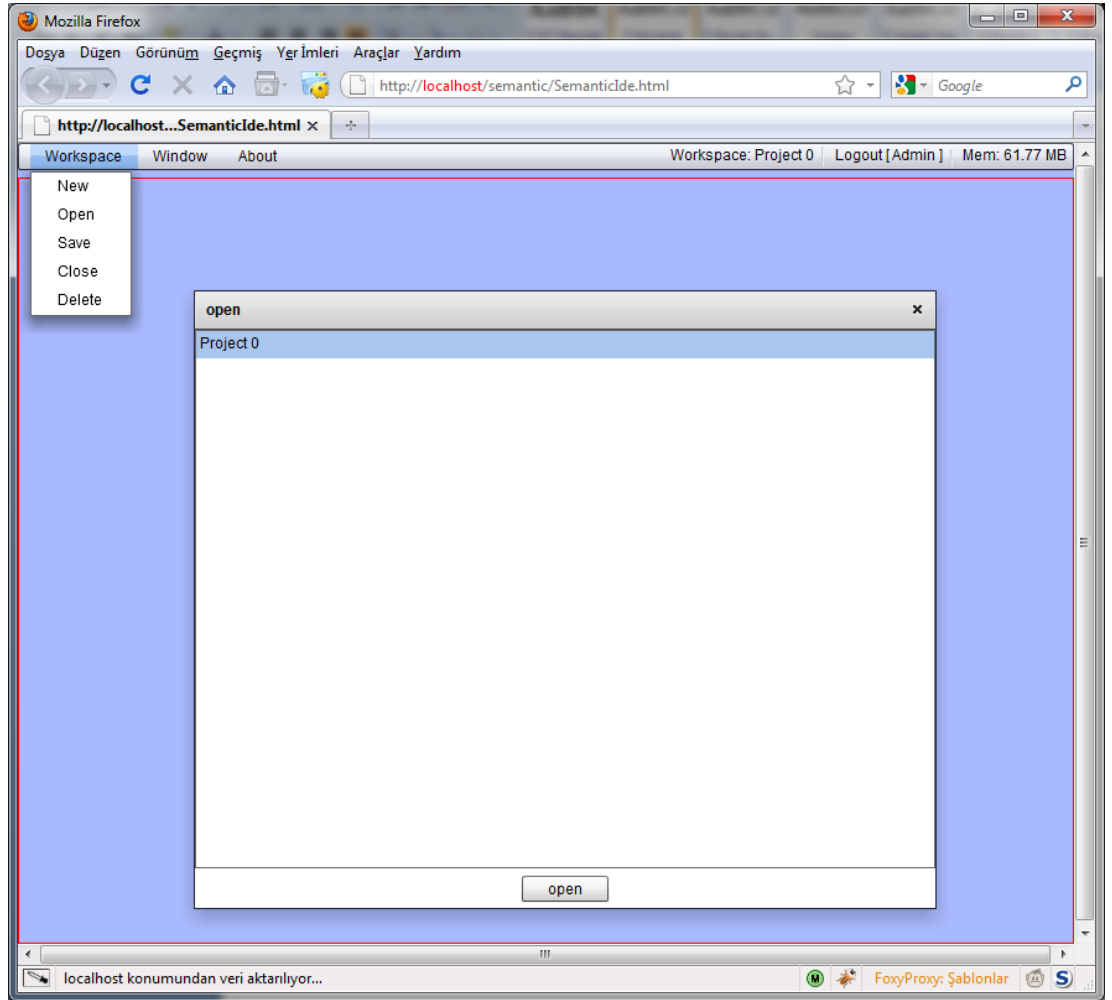
Şekil 4-8: Kullanıcı giriş ekranı



## Çalışma Ortamı İşlemleri

Bu menüdeki komutlarla;

- Yeni çalışma ortamı yaratılabilir
- Daha önce kaydedilmiş olan çalışma ortamı açılabilir
- O anda kullanılan çalışma ortamı kaydedilebilir
- O anda kullanılan çalışma ortamı kapatılabilir
- Daha önce kaydedilmiş çalışma ortamı silinebilir.

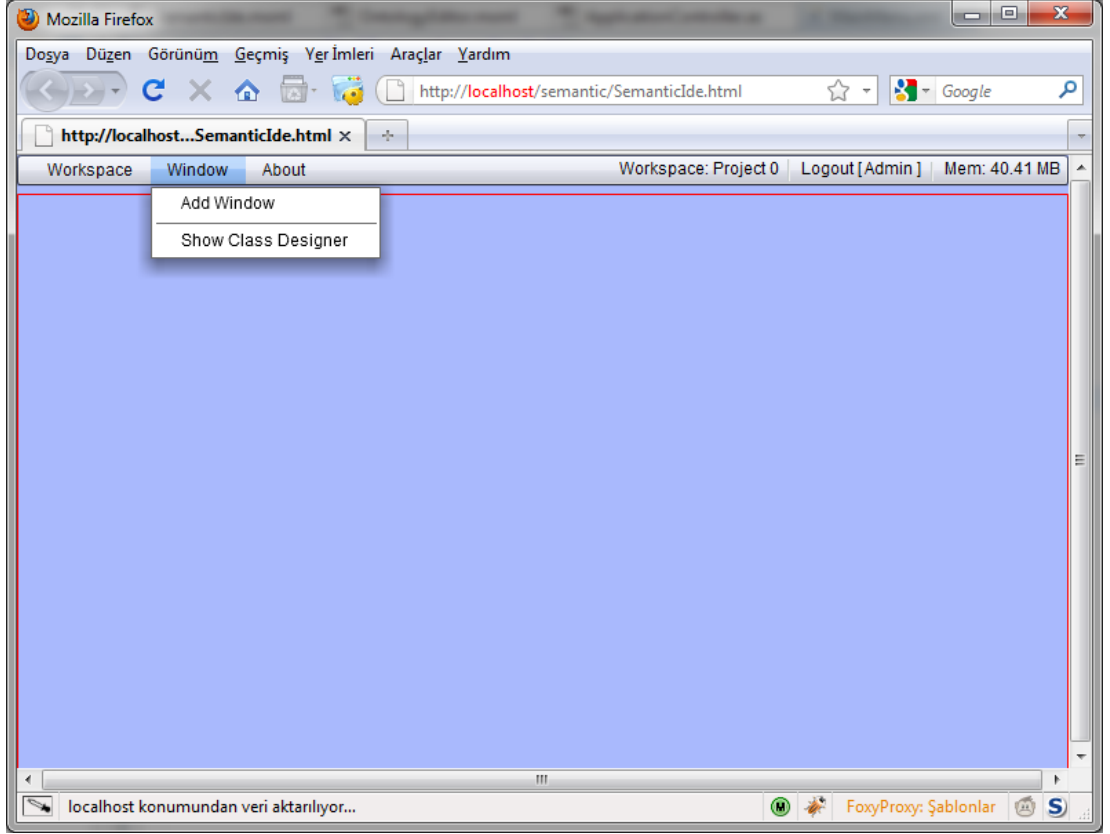


Şekil 4-9: Çalışma ortamı işlemleri menüsü

## Pencere İşlemleri

Bu menüdeki komutlarla;

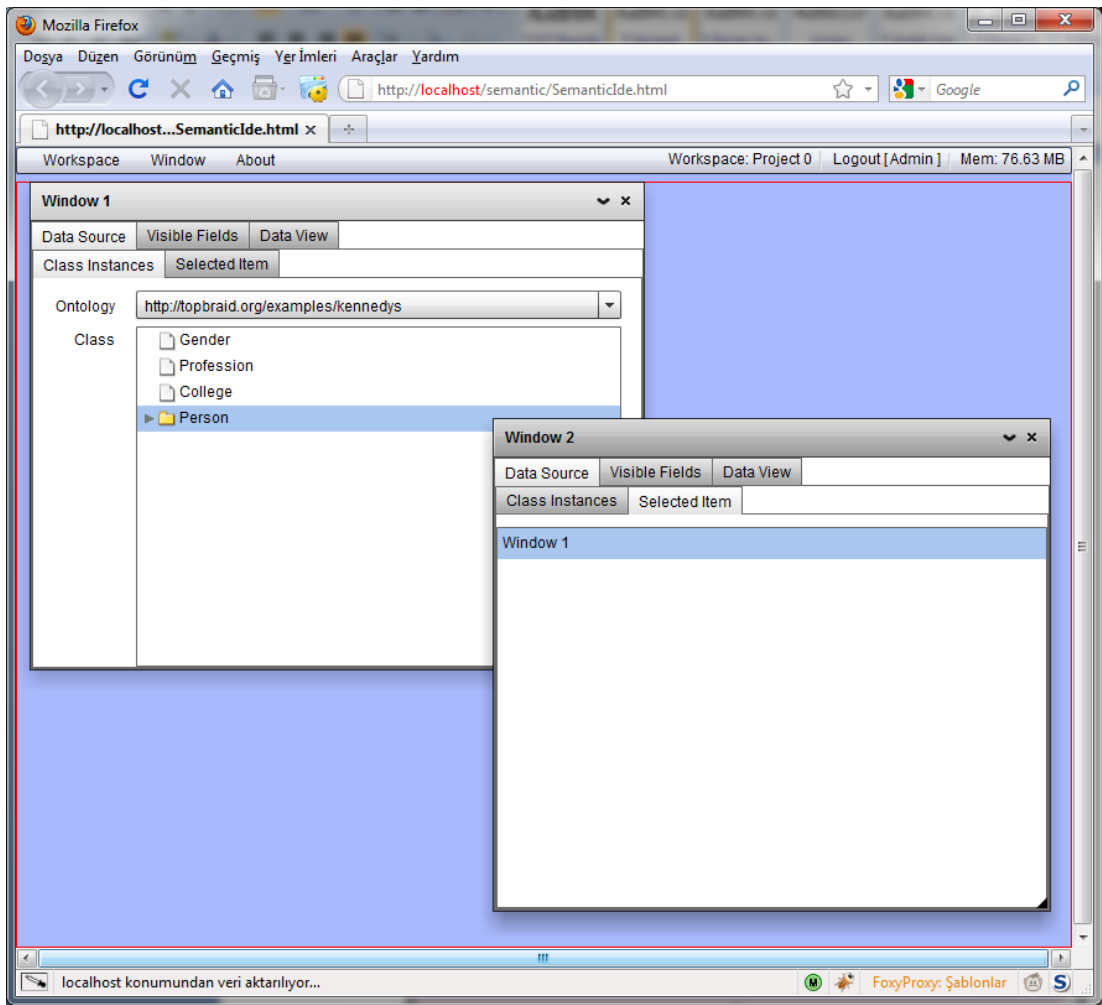
- Yeni bir pencere eklenebilir.
- Sınıf Tasarımcısı açılabilir.



Şekil 4-10: Pencere işlemleri menüsü

## Yeni Pencere Oluşturmak ve Pencere İçin Veri Kaynağını Seçmek

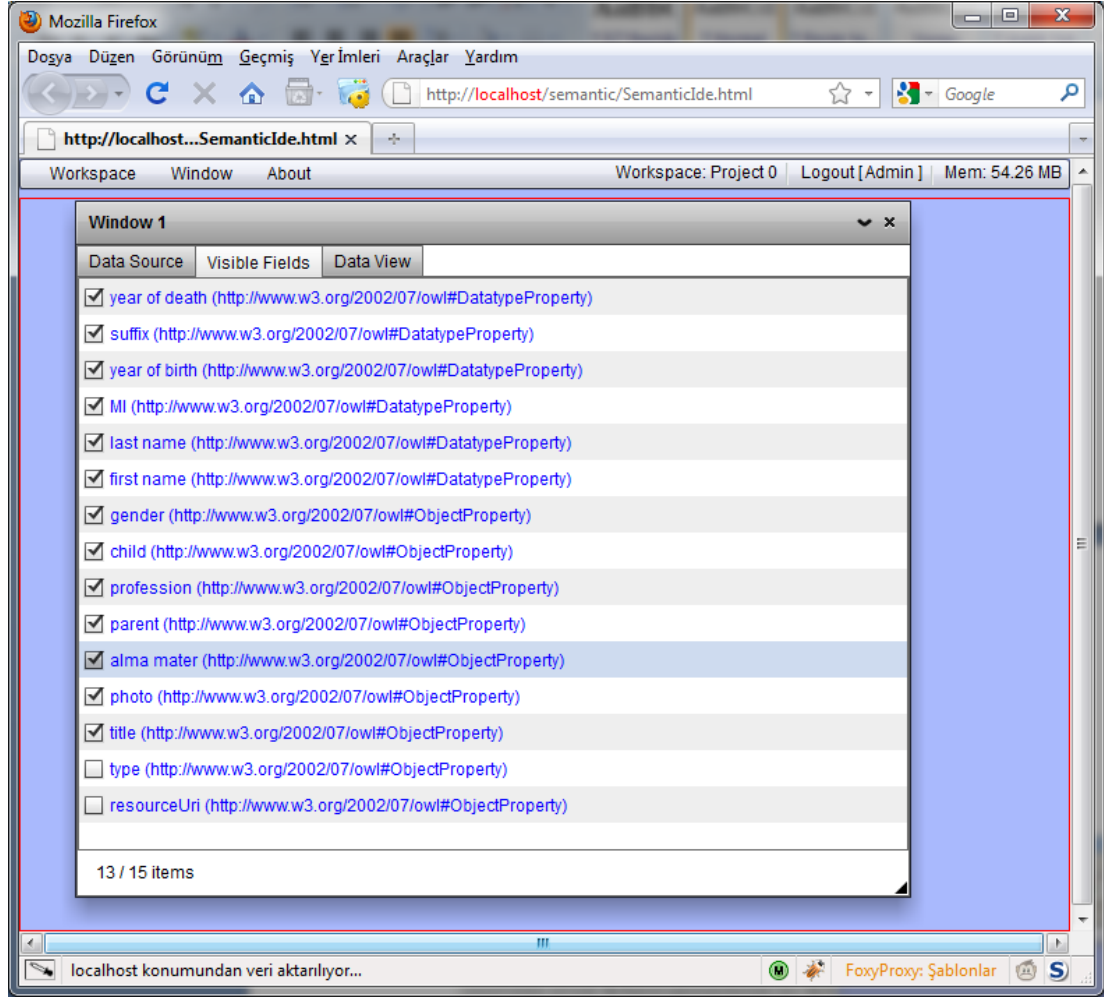
- “Window->Add Window” menüsü seçilir.
- “Data Source” sekmesinden veri kaynağı olarak
  - “Class Instances” sekmesinden sistemdeki bir ontoloji seçilebilir
  - “Selected Item” sekmesinden daha önce oluşturduğumuz bir pencerede seçili olan bir sınıf bireyi seçilebilir.



Şekil 4-11: Veri kaynağı seçim ekranı

## Seçili Sınıfın Görüntülenecek Özelliklerini Seçmek

Seçili pencerenin “Visible Fields” sekmesinden seçili olan sınıfın görünmesi istenen özellikleri seçilir.



Şekil 4-12: Sınıf özellikleri seçim ekranı

## Örnek Veri Görünümü

Örnek veri görünümünde birbirine bağlanmış ve görüntülenmesi istenen sınıf özellikleri seçilerek kişiselleştirilmiş iki pencere görülmekte.

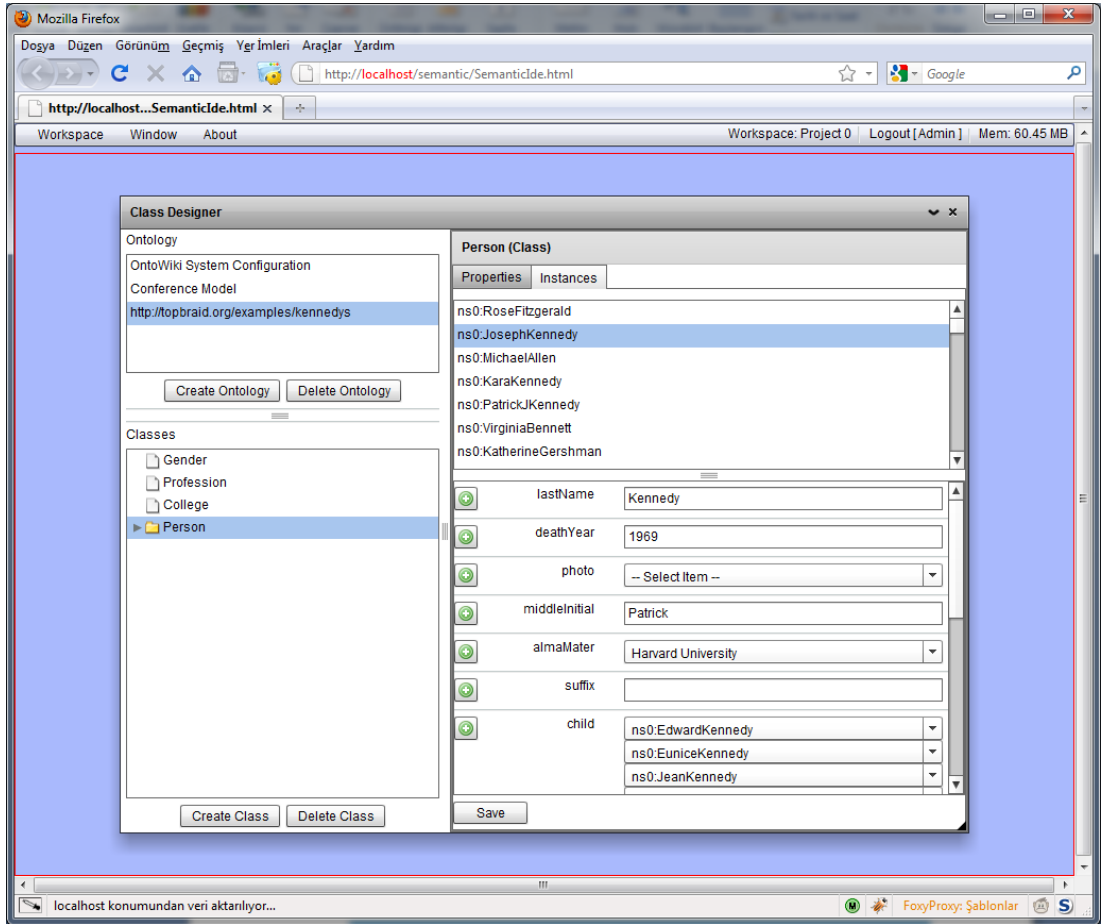
title	firstName	lastName	deathYear	birthYear	gender	photo	profession
ns0:RoseFitzgerald	Rose	Fitzgerald	1995	1890 1891	ns0:female	http://www.tons0:home-m	
ns0:JosephKennedy	Joseph	Kennedy	1969	1888	ns0:male	http://www.tons0:ambass	ns0:banker ns0:producer
ns0:MichaelAllen	Michael	Allen	null	1958	ns0:male	null	null
ns0:KaraKennedy	Kara	Kennedy	null	1960	ns0:female	null	null
ns0:PatrickJKennedy	Patrick	Kennedy	null	1967	ns0:male	http://www.tons0:congress	
ns0:VirginiaBennett	Virginia	Bennett	null	1936	ns0:female	null	ns0:home-m
ns0:KatherineGershman	Katherine	Gershman	null	1959	ns0:female	null	
ns0:EdwardKennedy	Edward	Kennedy	null	1961	ns0:male	null	
ns0:EdwardKennedy	Edward	Kennedy	null	1932	ns0:male	null	
ns0:WilliamCavendish	William	Cavendish	1944	1917	ns0:male	null	
ns0:KathleenAKennedy	Kathleen	Kennedy	1948	1920	ns0:female	null	
ns0:JeanOlsson	Jean	Olsson	null	1955	ns0:female	null	
ns0:ChristopherLawford	Christopher	Lawford	null	1955	ns0:male	null	
ns0:JamesMcKelvy	James	McKelvy	null	1955	ns0:male	null	
ns0:SydneyLawford	Sydney	Lawford	null	1956	ns0:female	null	
ns0:RobinLawford	Robin	Lawford	null	1961	ns0:female	null	
ns0:PeterLawford	Peter	Lawford	1984	1923	ns0:male	null	
ns0:RobertPenderJr	Robert	Pender	null	1953	ns0:male	null	
ns0:VictoriaLawford	Victoria	Lawford	null	1958	ns0:female	null	
ns0:PatriciaKennedy	Patricia	Kennedy	null	1924	ns0:female	null	

Şekil 4-13: Örnek veri görünüm ekranı

## Sınıf Tasarımcısı

Sınıf Tasarımcısıyla;

- Ontoloji yaratılır, silinir
- Sınıf yaratılır, silinir, değiştirilir
- Birey yaratılır, silinir, değiştirilir



Şekil 4-14: Sınıf tasarımcısı ekranı

#### 4.6 Ontoloji Geliştirme Araçlarının Karşılaştırılması

Bu bölümde, ilk olarak, dördüncü bölümde incelediğimiz ontoloji geliştirme araçlarıyla SewIde'nin benzerlikleri ve farkları ortaya koyarak irdelemeye çalışacağız. Bu bağlamda, incelenen araçların özelliklerini teknik ve fonksiyonel olarak iki kategoride incelemekte fayda bulunduğunu düşünüyoruz.

Çizelge 4-1: Ontoloji geliştirme araçları teknik özellikleri karşılaştırması

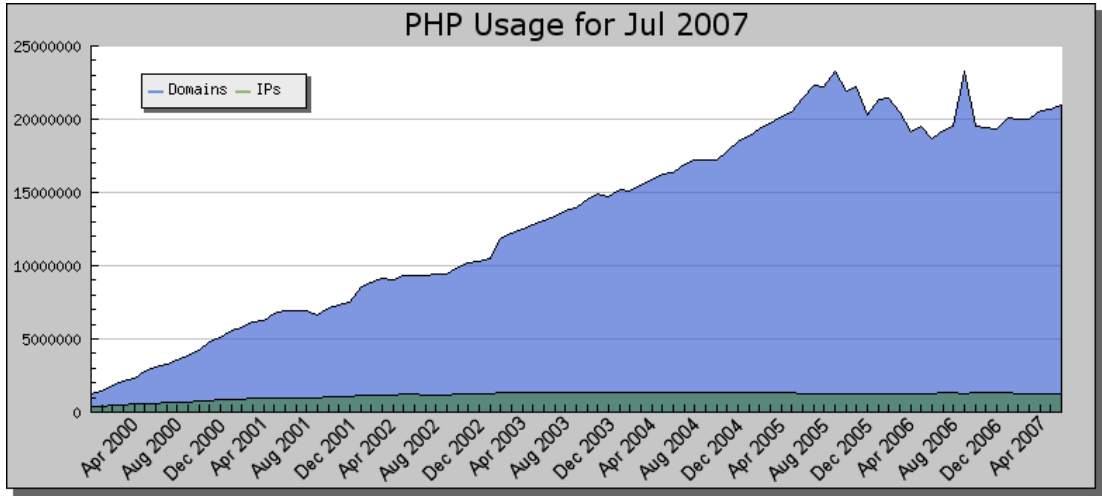
	OntoWiki	WebProtégé	MoKi	TopBraid Ensemble	SewIde
Programlama Dili	PHP	JAVA	PHP	JAVA	PHP
Arayüz Bileşenleri	JQUERY	GWT-Ext	YUI	Adobe Flex	Adobe Flex
Sunucu Kütüphanesi	Erfurt API	Protege	Semantic MediaWiki	Özel	Erfurt API
Açık Kaynak / Ücretsiz	✓✓	✓✓	✓✓	✗✗	✓✓

Çizelge 4-2: Ontoloji geliştirme araçları fonksiyonel özellikleri karşılaştırması

	OntoWiki	WebProtégé	MoKi	TopBraid Ensemble	SewIde
Arama	✓	✓	✓	✓	✗
Wiki	✓	✗	✓	✗	✗
SPARQL Sorgusu	✓	✗	✗	✓	✗
Arayüzü Kişiselleştirme	✗	✗	✓	✓	✓
Asıl-Detay İlişkisi Kurma	✗	✗	✗	✓	✓
Ortak Ontoloji Geliştirme	✓	✓	✓	✓	✓
Ontolojilerin Otomatik Güncellenmesi	✓	✗	✗	✓	✗

## Programlama Dili

SewIde'nin temel hedeflerinden birisi, kolay kurulum, kolay bakım, kolay değiştirilebilme, kolay anlaşılabilme ve en geniş kullanıcı kitlesine hitap etme özelliklerine sahip olmaktır. Bu çerçevede incelediğimiz sistemler PHP<sup>1</sup> ve JAVA<sup>2</sup> dillerinde yazılmıştır. Bu noktada tercihimiz dünyada en yaygın olarak kullanılan PHP dili oldu.



Şekil 4-15: 2007 yılı itibariyle PHP dilinin yaygınlığı <sup>3</sup>

## Arayüz Bileşenleri

Bu kategoride, incelediğimiz sistemlerde JQuery, YUI, GWT ve Adobe Flex Framework kullanıldığını görüyoruz. JQuery<sup>4</sup> ve YUI<sup>5</sup> Javascript dilinde yazılmış arayüz kütüphaneleri sunarlarken, GWT-Ext<sup>6</sup> Java dilinde yazılmış kodun Javascript'e derlenmesi ve ext.js<sup>7</sup> arayüz kütüphanesiyle birleştirilmesi prensibiyle çalışmaktadır. Flex Framework ise XML tabanlı bir bildirimsel bir dil olan MXML

<sup>1</sup> <http://php.net>

<sup>2</sup> <http://java.com>

<sup>3</sup> <http://php.net/usage.php>

<sup>4</sup> <http://jquery.com/>

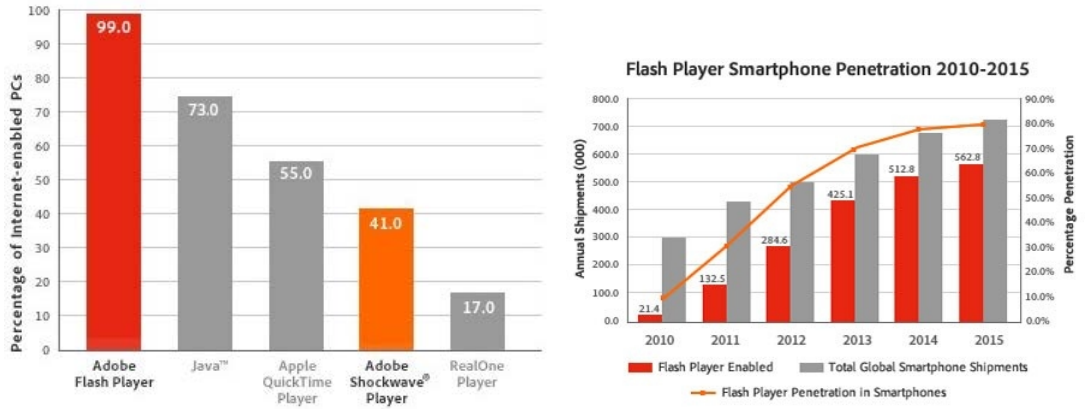
<sup>5</sup> <http://developer.yahoo.com/yui/>

<sup>6</sup> <http://code.google.com/p/gwt-ext/>

<sup>7</sup> <http://www.sencha.com>



ve Actionscript3 dillerinde yazılmıştır. Her ne kadar web tarayıcısına ek olarak bir eklenti kurulmasını gerektirse de, Flash Player'ın yaygınlığı ve aynı kodun mobil cihazlarda da kullanılabilmesine imkan veriyor olması göz önüne alınarak, değişik tarayıcılarda ve cihazlarda mümkün olan en iyi kullanıcı deneyimine olanak vermek için Adobe Flex Framework'de karar kıldık.



Şekil 4-16: Flash Player yaygınlığı <sup>1</sup>

### Sunucu Kütüphanesi (Backend Library)

Bu kategoride, incelenen sistemlerde kullanıcı işlemleri, veri ambarı, ontoloji erişimi, çıkarsama motoru, uygulama sunucusu, önbellekleme ve diğer sunucu tarafı işlemleri için kullandığı destek kütüphanelerini göstermektedir.

SewIde, OntoWiki uygulamasında da kullanılan, Erfurt Api'yi kullanarak belirtilen sunucu işlemlerini gerçekleştirmektedir.

### Açık Kaynak / Ücretsiz

Temel hedeflerimizden birisi olan geniş kitlelere ulaşmak ve açık kaynak kodlu sistemlerin gelişmesine katkıda bulunmak için SewIde'nin açık kaynak kodlu ve

<sup>1</sup> <http://www.adobe.com/products/flashplatformruntimes/statistics.html>

ücretsiz olmasını tercih ettik. İncelediğimiz dört sistemin üçü açık kaynak kodlu ve ücretsizdir.

### **Arama**

Bu özellik, ontoloji geliştirme araçlarının arama özelliklerinin bir özeti niteliğindedir. SewIde, şu haliyle herhangi bir arama özelliği içermemektedir. Her aracın sunduğu arama yetenekleri değişkenlik göstermektedir. Örneğin, OntoWiki sadece sınıf bireyleri üzerinde arama yapmaya izin verirken, TopBraid Ensemble kullanıcı tanımlı arama özelleştirmelerine olanak sağlamaktadır.

### **Wiki**

İncelediğimiz araçlardan OntoWiki ve MoKi, wiki sayfalarının, ontoloji sınıf ve bireyleriyle eşleştirilebilmesini sağlayan hibrit bir yapı sunmaktadırlar. SewIde henüz böyle bir imkan vermemektedir.

### **SPARQL Sorgusu**

Bu özellik, araçların elle girilen bir SPARQL sorgusu yapmaya izin verip vermediklerini belirtmektedir. Karşılaştırdığımız araçlardan OntoWiki ve TopBraid Ensemble bu özelliği taşımaktadır. SewIde, Erfurt Api ve OntoWiki altyapısını temel aldığı için bu özelliği eklemek nispeten kolay olacaktır, ancak zaman kısıtları nedeniyle bu özellik gelecek geliştirmelere bırakılmıştır.

### **Arayüzü Kişiselleştirme**

Bu çalışmamızda incelediğimiz ontoloji geliştirme araçları, birçok yönden oldukça gelişmiş olmalarına rağmen, son kullanıcının kendini rahat hissedeceği ve aşına olduğu bir kullanıcı deneyimi sunmak yerine, her kullanıcı için sabit, çoğunlukla kişiselleştirilemeyen bir arayüz ve deneyim sunmaktadırlar. İncelediğimiz ontoloji geliştirme araçlarının, “TopBraid Ensemble” hariç, hemen hepsinde kullanıcı çalışma ortamının kişiselleştirilmesine ya izin verilmediği ya da çok sınırlı olarak imkân tanındığı görülmüştür.

SewIde diđer sistemlerin bu eksiđini gidererek, alıřma ortamının kiřiselleřtirilip kaydedilmesine olanak sađlamaktadır.

### **Asıl-Detay İliřkisi Kurma**

“TopBraid Ensemble” hari, incelediđimiz ontoloji geliřtirme aralarından hibiri kiřiselleřtirilmiř arayüzler arasında asıl-detay iliřkisi kurulmasına olanak vermemektedir. Bu özellik, kullanıcıların genelde kullanmaya alışageldiđi bir yöntemdir ve sıka kullanılmaktadır. Bu bağlamda, SewIde alıřma ortamındaki arayüzlerin birbiriyle iliřkilendirilmesine izin vermektedir.

### **Ortak Ontoloji Geliřtirme**

İncelediđimiz araların hepsi birden ok kullanıcının ortak olarak ontoloji geliřtirmesine imkân vermektedir.

### **Ontolojilerin Otomatik Güncellenmesi**

Dıř sistemlerde bulunan ontolojilerden veri aktarmak ve arama iřlemlerini gerekleřtirmek hem gecikmelere neden olur hem de yerel sistemin dıř sistemlere bađımlılıđını artırır. Güncellik, güvenilirlik ve performans noktasında bir orta yol bulabilmek adına, dıř sistemlerde barındırılan ontolojileri belirli periyotlarda yerel sistemin veri ambarına aktarmak akıllıca olmaktadır. İncelediđimiz sistemlerden iki tanesinde bu özellik dođrudan desteklenmektedir. Bu özelliđi desteklemeyen SewIde ve diđer sistemlerde de kolayca gerekleřtirilebileceđi dıřünülmektedir.

## 5 DEĞERLENDİRME VE GELECEK İYİLEŞTİRMELER

Kullanıcılar Web 2.0 ile beraber, klasik web'den farklı ortamlardan ulaşılabilen, hızlı ve kolay erişimi sağlayan web'e yönelmeye başladılar. Bir sonraki adımda ise, Web 3.0 olarak anılan ve dağıtık veri modellerini bir araya getirerek ağ üzerinden diğer yazılım sistemleriyle paylaşmayı sağlayan Semantik Web (SW) ortaya çıkmıştır. SW yoğun olarak ontolojilerden faydalanmaktadır. Bu ontolojiler üzerinde çalışmayı kolaylaştırmak için ortaya çıkan ilk nesil ontoloji geliştirme araçları klasik masaüstü uygulamalarıydı. Bu uygulamalar, zaman içinde kullanıcıların ontolojiler üzerinde ortak olarak çalışmasını sağlayan, Web 2.0 teknolojileri kullanılarak hazırlanmış ontoloji geliştirme araçlarına doğru evrilmiştir.

Tezimizi, bu yeni nesil ontoloji geliştirme araçlarında hissedilen bazı eksiklikleri gidermek üzerine kurguladık. Çeşitli özelliklere göre yaptığımız karşılaştırmalarda, bu sistemlerin çoğunlukla, kullanıcıya çalışma alanını (workspace) kişiselleştirme, kaydetme ve geri yükleme imkânı sunmadıklarını tespit ettik. Araştırmamız sırasında, "TopBraid Ensemble" adlı sistemin bizim kriterlerimizin hemen hepsini karşıladığını, hatta fazlasını sunduğu gördük. Ancak bu sistem ticari ve kapalı kaynak kodlu bir sistem olduğu için, yaygınlık ve genişletilebilirlik noktalarında zayıf kalmaktadır.

Tespit edilen bu eksiklikleri gidermek üzere, hazırladığımız tezde, kullanıcıların; ontoloji görünümünü kişiselleştirilebilmesi, bu görünümü asıl-detay ilişkisiyle birbirine bağlayabilmesi ve kişiselleştirilen bu çalışma alanlarını sunucu üzerinde saklayıp, istedikleri zaman geri çağrılabilirdikleri SewIde'yi geliştirdik.

Geliştirilme sırasında çeşitli zorluklarla karşılaştık. İlk etapta, Flex arayüz ve PHP uygulama sunucusu arasındaki iletişimi sağlayacak protokolün Adobe firmasının oluşturduğu, hızlı, ikili sisteme dayalı AMF (Action Message Format) olmasına karar verilmişti. Ancak uygulama geliştirme aşamalarında ortaya çıkan ve çözülemeyen bazı sorunlar neticesinde, daha basit ve takip edilebilir bir veri biçimi olan JSON kullanmaya karar verdik.

Bu çerçevede gereksinimlerimizi belirleyip, bu gereksinimlerimizin gerçekleştirimini sağlayacak sunucu fonksiyonlarını, Erfurt Api ve bazı OntoWiki kütüphanelerini kullanarak gerçekleştirdik.

Sonuç olarak, aynı anda birden çok kullanıcının ontolojiler üzerinde ortak olarak çalışmasına izin veren, çalışma ortamlarını ve arayüzlerini kişiselleştirmelerini, kaydetmelerini ve tekrar geri yükleyerek kullanabilmelerini sağlayan SewIDE'yi ortaya çıkardık.

Çalışmalarımız sırasında, zaman kısıtları yüzünden gerçekleştiremediğimiz bazı özelliklere de kısaca değinelim.

SewIde, bu haliyle herhangi bir arama özelliği içermemektedir. Az bir çabayla, en azından, sınıf bireyleri üzerinde arama yapılabilmesini düşünüyoruz.

Yine, SPARQL sorgulama özelliğinin de, Erfurt Api ve OntoWiki kütüphaneleri aracılığıyla hızlıca gerçekleştirilebileceği kanaatindeyiz.

Ontolojiler üzerinde aynı anda yapılan değişikliklerde meydana gelen çakışmalardan kullanıcıların haberdar edilmesi ve çalışma alanları üzerinde bir sahiplik veya yetki denetimi yapılması da uygulanabilecek diğer geliştirmelerdendir.

TopBraid Ensemble gibi ticari sistemlerin ek olarak sunduğu, ağaç, harita, grafik görselleştirme gibi arayüz öğeleri de yapılabilir geliştirmeler arasında yer bulacaktır.

## 6 KAYNAKLAR

- [1] Herman, I., “Semantic Web Frequently Asked Questions”, W3C, 2009, <http://www.w3.org/RDF/FAQ#relateweb>, erişim tarihi: Ağustos 2009
- [2] Berners-Lee, T., Hendler, J. and Lassila, O., The Semantic Web, Scientific American, 284(5): 34-43, 2001
- [3] Herman, I., “Semantic Web Activity”, W3C, 2009, <http://www.w3.org/2001/sw/>, erişim tarihi: Ağustos 2009
- [4] “Uniform Resource Identifier (URI)”, Wikipedia, 2009, [http://en.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://en.wikipedia.org/wiki/Uniform_Resource_Identifier), erişim tarihi: Ağustos 2009
- [5] “Extensible Markup Language (XML)”, Wikipedia, <http://en.wikipedia.org/wiki/XML>, erişim tarihi: Ağustos 2009
- [6] “XML Schema (W3C)”, Wikipedia, [http://en.wikipedia.org/wiki/XML\\_Schema\\_\(W3C\)](http://en.wikipedia.org/wiki/XML_Schema_(W3C)), erişim tarihi: Ağustos 2009
- [7] “Resource Description Framework (RDF)”, W3C, 2004, <http://www.w3.org/RDF/>, erişim tarihi: Ağustos, 2009
- [8] “Resource Description Framework (RDF)”, Wikipedia, [http://en.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://en.wikipedia.org/wiki/Resource_Description_Framework), erişim tarihi: Ağustos 2009
- [9] Gruber, Thomas, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal Human-Computer Studies Vol. 43, Issues 5-6, November 1995, p.907-928, 1995
- [10] Herman, I., “Semantic Web Frequently Asked Questions”, W3C, 2009, <http://www.w3.org/2001/sw/SW-FAQ>, erişim tarihi: Ağustos 2009
- [11] “RDF Vocabulary Description Language 1.0: RDF Schema (RDFS)”, W3C, 2004, <http://www.w3.org/2001/sw/wiki/RDFS>, erişim tarihi: Ağustos, 2009
- [12] McGuinness, D.L., van Harmelen, F., “OWL Web Ontology Language Overview”, 2004, <http://www.w3.org/TR/owl-features/>, erişim tarihi: Ağustos, 2009
- [13] “SPARQL Protocol For RDF”, W3C, 2008, <http://www.w3.org/TR/rdf-sparql-protocol/>, erişim tarihi: Ağustos, 2009
- [14] Deborah L. McGuinness, Ontological issues for knowledge-enhanced search, Proceedings of Formal Ontology in Information Systems, 1998, Frontiers in Artificial Intelligence and Applications, IOS-Press, 1998
- [15] N. Guarino, C. Masolo, and G. Vetere, Ontoseek: Content-based access to the web, IEEE Intelligent Systems, 1999
- [16] S. Auer, S. Dietzold, J. Lehmann, and T. Riechert, Ontowiki: A tool for social, semantic collaboration, Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007, Banff, Canada, 2007
- [17] “OntoWiki”, <http://ontowiki.net>, erişim tarihi: Ağustos, 2009

- [18] Tudorache, T., Vendetti J., Noy N. F., Web-Protege: A Lightweight OWL Ontology Editor for the Web, OWL: Experiences and Directions, 4th Intl. Workshop, OWLED DC, 2008, Washington, DC, USA, 2008
- [19] “Collaborative Protégé”,  
[http://protegewiki.stanford.edu/wiki/Collaborative\\_Protege](http://protegewiki.stanford.edu/wiki/Collaborative_Protege), erişim tarihi: Aralık 2011
- [20] “WebProtégé”, <http://protegewiki.stanford.edu/index.php/WebProtégé>, erişim tarihi: Ağustos, 2009
- [21] Chiara Ghidini, Barbara Kump, Stefanie N. Lindstaedt, Nahid Mahbub, Viktoria Pammer, Marco Rospocher, and Luciano Serafini, MoKi: The enterprise modeling wiki, In ESWC’09: The Semantic Web: Research and Applications, volume 5554 of LNCS, pages 831–835, Springer, 2009
- [22] “APOSDLE”, <http://www.aposdle.tugraz.at>, erişim tarihi: Aralık 2011
- [23] “MoKi”, <https://moki.fbk.eu/website/index.php>, erişim tarihi: Aralık 2011
- [24] “TopBraid Ensemble”, [http://topquadrant.com/products/TB\\_Ensemble.html](http://topquadrant.com/products/TB_Ensemble.html), erişim tarihi: Aralık 2011
- [25] Gamma, Erich; Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, 1995
- [26] “Tasarım Örüntüleri”, Wikipedia,  
[http://tr.wikipedia.org/wiki/Tasarım\\_örüntüleri](http://tr.wikipedia.org/wiki/Tasarım_örüntüleri), erişim tarihi: Eylül 2009
- [27] “Apache HTTP Sunucusu”, Wikipedia,  
[http://tr.wikipedia.org/wiki/Apache\\_HTTP\\_Sunucusu](http://tr.wikipedia.org/wiki/Apache_HTTP_Sunucusu), erişim tarihi: Aralık, 2011
- [28] “PHP: Hypertext Preprocessor”, Wikipedia,  
<http://tr.wikipedia.org/wiki/PHP>, erişim tarihi: Ağustos, 2009
- [29] “MySQL”, Wikipedia, <http://tr.wikipedia.org/wiki/MySQL>, erişim tarihi: Aralık, 2011
- [30] “Adobe Flex”, Wikipedia, [http://tr.wikipedia.org/wiki/Adobe\\_Flex](http://tr.wikipedia.org/wiki/Adobe_Flex), erişim tarihi: Aralık, 2011
- [31] “Adobe Flash Builder”, <http://www.adobe.com/products/flash-builder-family.html>, erişim tarihi: Aralık, 2011
- [32] “Adobe Flex SDK”,  
<http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+4>, erişim tarihi: Ağustos, 2009
- [33] “Erfurt API”, <http://aksw.org/Projects/Erfurt>, erişim tarihi: Ağustos, 2009
- [34] Reenskaug, Trygve, “MVC XEROX PARC 1978-79”,  
<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>, erişim tarihi: Ekim 2009
- [35] Steve Burbeck, “Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller”, <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>, erişim tarihi: Haziran, 2009
- [36] “JSwoof”, <http://www.waynemike.pwp.blueyonder.co.uk/jswoof/>, erişim tarihi: Ağustos, 2009
- [37] Martin Fowler, “Inversion of Control Containers and the Dependency Injection pattern”,

- <http://martinfowler.com/articles/injection.html#InversionOfControl>, erişim tarihi: Ağustos, 2009
- [38] “Parsley”, <http://www.spicefactory.org/parsley/>, erişim tarihi: Ağustos, 2009
- [39] Selic, B., The Pragmatics of Model-Driven Development, IEEE Software, 20: 19-25, 2003
- [40] OMG, “Model Driven Architecture Guide Version 1.0.1”, OMG Document Number: omg/2003-06-01, 2003, <http://www.omg.org/docs/omg/03-06-01.pdf>, erişim tarihi: Ekim, 2009
- [41] OMG, “Meta Object Facility Specification”, OMG Document Number: AD/97-08-14, 1997, <http://www.omg.org/docs/ad/97-08-14.pdf>, erişim tarihi: Ekim, 2009



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Adı Soyadı: Ahmet ÖMERCİOĞLU  
Uyruğu: T.C  
Doğum tarihi ve yeri: 17.09.1976, Ankara  
Medeni Hali: Evli  
Telefon: (+90) (542) 723 15 06  
Email: omercioglu@etu.edu.tr, omercioglu@hotmail.com

### Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	ODTÜ, Bilgisayar Öğretmenliği	2002

### İş Deneyimi

Yıl	Yer	Görev
2002-2004	Milli Eğitim Bakanlığı	Öğretmen
2004-	TOBB ETÜ	Yazılım Uzmanı

### Yabancı Dil

İngilizce