# XML

KURTEL’ e göre XML’ in 4 konudaki başarısı nedeni ile kullanılmaya devam etmektedir. Bunlar:

* Uygulamalardan bağımsız veri ve belge oluşturma imkânı.
* Üst veri (veri hakkında veri – meta data) gösterimi için standartlar sunması.
* Veri ve belge için ortak yapısal standartlar sunması.
* Sınanmış bir teknoloji olması (KURTEL, 2008).

XML bu özellikleri ile anlamsal web’ in en önemli bileşenlerinden biri olmaktadır.

Örnek bir XML belgesi içeriği aşağıdaki gibidir.

<?xml version="1.0"?>  
<bilisimogrenci>  
<adi>Furkan</adi>  
<soyadi>Dündar</soyadi>  
<adresi>Akyazı/SAKARYA</adresi>  
</ bilisimogrenci>

HTML’ den en önemli farkı, önceden tanımlı olmayan kullanıcılar tarafından oluşturulabilen tag’ ların kullanılabilmesidir. HTML verinin şekli ile ilgilenirken, XML verinin kendisi ile ilgilendiğinden XML belgelerinde verinin tanım bilgilerine geniş verilir. Bu şekilde verilerin değişik ortamlara taşınması amaçlanmıştır.

XML'in avantajları:

* Açık standartları vardır.
* Genişletilebilir XML belgesine kendi etiketlerimizi ekleyebiliriz.
* İşletim sistemine, uygulamanın yazıldığı programlama diline veya veri kaynağına bağlı olmadığından esnek web uygulamalarının geliştirilmesinden kolaylık sağlar.
* Farklı veri kaynakları ve uygulama programları arasında verilerin akışına ve veri bütünleşmesine olanak sağlar
* Veri kaynaklarından sağlanan verilerin değişik görünümlerinin elde edilmesine olanak sağlar.
* Veri kaynaklarındaki veriler üzerinde yerel olarak hesaplama yapılmasını sağlar. Elde edilen veriler tarayıcı tarafından okunur ve ardından sonraki aşama için bir yerel uygulama programına aktarılır.
* Farklı uygulama programları içerisindeki verileri tanımlayabilir. Verinin kendisi kendini tanımlayabildiğinden kaynak uygulama programına bağlı kalmadan veri işlenebilir.

XML, sadece değişen verilerin güncellenmesi için de olanak sağlar. Değişen verinin görüntülenmesi bütün sayfanın yeniden yüklenmesine ihtiyaç duymaz.

# WEB SERVİSLERİ

Web servisleri, web standartları kullanarak, farklı programlama dilleri ile farklı işletim sistemlerinde oluşturulan yazılım bileşenlerinin birbirleri ile etkileşimine olanak sağlayan teknolojilerdir (DUTTA, 2008).

Web servisleri, web ortamında belirli görev veya görevleri yerine getiren işlemleri tanımlamak için ara yüz oluşturmaktadırlar. Web Service Description Language (WSDL) kullanılarak tanımlanan web servisleri, servis yayınlama, bulma, bağlanma ve çağırma işlemleri için Simple Object Access Protocol (SOAP)’ ı kullanmaktadırlar (YÜKSEK, 2009).

**Şekil : Web servis mimarisi**



**Kaynak:** (YÜKSEK, 2009)

# Web servis mimarisi yukarıdaki şekilde de görüldüğü gibi 3 temel bileşenden oluşur. Bunlar servis sağlayıcısı, servis kayıtçısı ve servis kullanıcısıdır. Servis sağlayıcısı, servis kullanıcılarının servis sağlayıcısındaki servislere ulaşmasını sağlar. Servis kayıtçısında ise servis sağlayıcısının yayınladığı web servislerinin nasıl çağrılacağı ile ilgili tanımlar kayıt edilir. Servis kullanıcısı, servis kayıtçısındaki kurallara bakarak servis sağlayıcısındaki servisleri çağıran uygulamalardır.

Web servisleri genellikle veri formatı olarak XML’ i kullanırlar.

# SOAP (Simple Object Access Protocol)

Servis kullanıcısı ile servis sağlayıcısı arasında iletişimim hangi kurallar üzerinden sağlanacağını SOAP belirler. Bu iletişimin standartların W3C tarafından belirlenmektedir. XML veri formatını kullandığı için farklı platformlarda hazırlanan uygulamalar arasında iletişime imkân tanır. 3 çeşit SOAP mesajı oluşabilir. Bunlar metod kullanım isteği, cevap mesajı ve hata mesajıdır.

Bir SAOP mesajında en temel olarak 3 bileşen bulunmaktadır. Bunlar Envelope, Header ve Body’ dir (ALGAN, 2011).

Envelope, SOAP mesajlarının kapsayan yapıdır. İçeriğinde body veya header gibi diğer elemanlarda bulunur. Her envelope’ in bir header taşıma zorunluluğu olmasa da mutlaka bir body taşımaktadır. Fakat eğer envelope içinde header bulunuyorsa, header mutlaka envelope’ nin ilk elemanı olmak zorundadır.

Header, meta-data (veri hakkında veri) taşıyan SOAP bileşenidir. SOAP’ ın çağırdığı metod ile herhangi bir ilişkisi yoktur.

Body, web servisi kullanıcısının web servisi sağlayıcısına, metodunun adı ve parametrelerini gönderdiği SOAP bileşenidir. Web servis sağlayıcısının gönderdiği cevap mesajındaki sonuç değeri de body bileşeninde gönderilir. Hata mesajında ise hatanın adı ve tanımı SOAP’ ın body bileşeninde gönderilir.

# WSDL (Web Service Definition Language)

SOAP’ ın farklı platformlarda ve dillerde oluşturulan uygulamaları konuşturması için gerekli olan standartlar WSDL ile oluşturulur. XML veri formatını kullanan WSDL SOAP’ ın isteklerini gerçekleştirebilmek için, web servisinin adı, çağrılabilecek metodlar, bu metodların kullanabileceği parametreler, parametrelerin veri tipleri, geri dönecek cevap ve cevabın formatı, hata oluştuğunda geri dönecek mesaj, web servis sağlayıcılara ulaşılabilecek adres gibi verileri saklar (W3.ORG, 2011).

# UDDI (Universal Description, Discovery and Integration)

Servis kullanıcısı, kendi ihtiyacına uygun bir web servisini nerede bulabileceğini, bu web servisini nasıl kullanabileceğini, UDDI’ ler aracılığı ile öğrenebilir. UDDI’ ler bir çeşit web servisi rehberidir. Fakat henüz tam olarak hazır bir teknoloji olmadığından UDDI’ leri kullanan API’ ler sürekli olarak güncellenmektedirler. Örneğin web servislerini sınıflandırmaktaki sıkıntılar halen devam etmektedir.

# SEMANTİK WEB

Web 3.0 dan önceki teknolojiler olan Web 1.0 ve Web 2.0’ ın temel amacı, makineler aracılığı ile insanlar arasında veri ve bilgi alışverişini kolaylaştırmaktır. Klasik web olarak da adlandırılan Web 1.0 ve Web 2.0’ ı temel alan ancak bu teknolojilerden çok daha üstün olduğunu iddia eden Anlamsal Web yani Web 3.0’ ın hedefi ise insanlar arasındaki veri ve bilgi paylaşımının yanı sıra makineler arasında da veri ve bilgi paylaşımını sağlamaktır. Bu amaçla anlamsal web verilerin iyi tanımlanabilmesi, birbirleri arasındaki hiyerarşinin ve ilişkilerin net belirlenmesi ve bu bilgi ve işlemlerin makineler tarafından kolaylıkla anlaşılıp kullanılabilmesi için standartlar ve teknolojiler geliştirmeyi hedeflemektedir. Artık web, web sayfalarındaki bilgileri taşıyan makineler ile bu bilgileri kullanan insanların oluşturdukları birliktelikten ibaret olmaktan öte, verilerin saklandığı klasik ortamlar olan veritabanları ve bu verileri işleyen programlar, servisler, ajanlara ek olarak daha ileriki zamanlarda da ev gibi çevre ortamlarda bulunan makinelerdeki verilerin de işlendiği devasa ve karmaşık bir yapı olacaktır (KURTEL, 2008).

Örneğin bir hasta tedavisi için doktora gitmek istediğinde, Web 1.0’ da internetteki hastaneleri inceleyerek kendine uygun hastane ve doktorları seçmeye çalışır. Web 2.0’ da ise hastane ve doktor ile etkileşime girerek hem kendisi için hem de hastane için uygun olan zamanda randevu alır. Sıra numarası almak, oraya ulaşım için biletleri tedarik etmek vb. işlemleri yine hasta kendisi yapar. Web 3.0’ da ise, anlamsal web kullanıcının ontolojisini inceleyerek (sosyal sigorta, boş zaman aralıkları, hastalık geçmişi, daha önceki doktorları ve tedavileri vb.) uygun hastane ve doktorları hastaya önerecek ve gerekli tüm sıra alma ve diğer işlemleri anlamsal web yapacaktır.

“Telefon çaldığında müzik setinde Beatles’ dan “We can work it out” çalıyordu. Pete telefonu açtığında, telefonu, ses kontrolü olan makinelere mesaj yollayarak seslerini kıstı.” (LEE, Hendler, & Lassila, The Semantic Web, 2001) Tim Berners-Lee ve arkadaşları yukarıdaki örnekte anlamsal web’ in sadece bilgisayarlarda değil tüm makinelerde uygulanabileceğini (Web 4.0) anlatmaya çalışıyorlar. Anlamsal web üzerinde bulunan telefon, yine anlamsal web üzerinde bulunan müzik setine, sesin kısılması gerektiğini yine anlamsal web üzerinden komutlar ile anlatıyor. Bu hayalin gerçekleşmesi için, tüm makineler, büyük bir kablosuz ağ içinde birbirine bağlanabilmeli ve birbirlerini tanıyabilmelidirler. Bunun için gerekli teknoloji ve altyapı günümüz itibari ile mevcut değildir.

Tim Berners-Lee ve arkadaşları aynı makalede anlamsal web’ in tanımı ve 3 temel özelliği üzerinde durmuşlardır.

Bu özellikler:

* Anlamsal web’in kaynağı klasik web’ dir. Klasik web tamamen bırakılıp yerine yeni bir teknoloji getirmek yerine, klasik web genişletilerek, yeni bir web sürümü olarak anlamsal web geliştirilmiştir.
* Klasik web’ de veriler sunulmaktadır. Anlamsal web’ de ise verilere ek olarak bu verilerden elde edilen bilgiler de bulunmaktadır. Bazı bilgiler başka bilgilerin verisi olarak da kullanılabilmektedir.
* Web 2.0 teknolojisi insanlar arasında işbirliği yapılmasını desteklemektedir. Anlamsal web ise sadece insanlar arasındaki işbirliğini yeterli görmeyip insanlar ile makineler arasında işbirliği yapılmasını da içermektedir. Bu sayede makinelerin insanlara göre üstün oldukları çok daha hızlı ve doğru işlem yapabilme özelliklerinden yararlanılmaktadır. Zamanla insanların makinelere güveni bu sayede artmasıyla, makinelerin kendi aralarında daha çok iletişime geçip işbirliği yapmalarının da önü açılmaktadır. Böylece tezimizin amaçlarından biri olan, makinelerin **insanların haberi olmadan daha fazla süreci gerçekleştirmesine, kendi kendilerine karar verip daha fazla süreci doğru ve hızlı şekilde tamamlamalarına** uygun yazılımlar çalıştırılabilecektir.

Bilgi, **insanlara has** algılama, anlama, muhakeme etme, yorumlama, doğrulama, değerlendirme ve bunun gibi iç içe geçmiş birçok etkinlikten oluşan bilme etkinliği ve bu bilme etkinliği sonucu elde edilen çıktı olarak tanımlanır (KUÇURADİ & Cohen, 1995). Bilginin kaynaklarından biri verilerdir. Verilerden elde edilen bilgiler başka bilgilere ulaşmak için veri olarak da kullanılabilirler. Bu da veri hakkında veriler yani üst veriler kavramını ortaya çıkarmaktadır. Kuçuradi’ nin sadece insanlara has olarak belirttiği etkinlikleri makinelerin başarılı şekilde taklit edebilmeleri için kaynak olarak kullandıkları üst verilerin doğru oluşturulması ve bu üst verileri doğru şekilde işleyen araçların olması çok önemlidir (MAEDCHEA & Staabb, Applying semantic web technologies for tourism information systems, 2002). Klasik web’ de makineler sadece veriye ulaşılabilirken anlamsal web’ de bilgiye de ulaşılabilir ve bunları paylaşabilir olduklarından, artık arama yapıldığında, aranan ile daha ilgili sonuçlar bulunabilmektedir.

Anlamsal web’ i, dünya çapındaki tüm makinelerin sanal bir ağla birbirine bağlandıkları bir yapı olarak düşündüğümüzde, her yerden erişime açık, zamanla bilgi birikiminin arttığı, insanlığın ortak bilincini temsil eden, hemen her şeyin üzerinde toplandığı, veri ve bilginin sadece depolanmayıp sürekli kullanılıp geliştirildiği, sürekli büyüyen bir beyin olarak da tanımlayabiliriz (FENSEL & Musen, 2001).

Anlamsal web ile ilgili tartışılan konulardan birisi de anlamsal web’ in bir yapay zekâ teknolojisi olup olmadığıdır. Tim Berners-Lee bu yanılgının sebebi olarak, anlamsal web’ in makineler tarafından anlaşılabilir belgeler üretmesi olduğu söylemektedir. Ancak makinelerin insanların dilini anlaması yerine, insanların makinelerin anlayacağı şekilde veriler içeren belgeler üretmesi gerektiğini söyleyerek anlamsal web’ in yapay zekâ teknolojisi olmadığını savunmuştur (LEE, Web Architecture from 50,000 feet, 1998). Tim Berners-Lee aynı makalede, anlamsal web’ in kendisinden önce denenmiş ve başarısız olmuş bir teknolojinin tekrar kullanılmaya çalışılan yeni bir sürümü olmadığını savunmuştur. İki teknoloji arasında temel fark, yapay zekâ teknolojilerinde, her kavramın bir modelde sadece bir yeri olmasıdır. Bu nedenle de birden fazla modelin ilişkilendirilmesinde veya birleştirilmesinde problemler çıkmaktadır. Bu da bağımsız modeller oluşturmaya engel olmaktadır. Anlamsal web geçmiş teknolojilerdeki hataları görüp tam ve kusursuz bir sistem olması yerine, bilinen kısıtları olan ve katmanlı olarak geliştirilen bir sistem yapısına sahip olması tasarlanmıştır (LEE & Fischetti, 1999). Yapay zekâ uzmanları ise bu tür kısıtları olan bir sistemin anlamlı ve tek başına yeterli olmayacağını, anlamsal web araçlarının bu tür sorunların çözümü için uzun yıllardır araştırma yapılan yapay zeka sistemlerinin çözümlerine bakarak geliştirilmesi gerektiğini çünkü anlamsal web’ in bir çok bilgi sistemin birleştiği dev boyutlu bir sistem olacağını savunmaktadırlar (SCHWARTZ, 2003).

# Anlamsal Web’ in Katmanları

Tim Berners-Lee 1999 yılında anlamsal web kavramını ortaya atmıştı. Anlamsal web’ e sürekli artan ilgi bu konuda yapılan araştırma yapanların ortak kararlar alabilmeleri için yine Tim Berners-Lee tarafından World Wide Web Community (W3C) kurulmuştur.

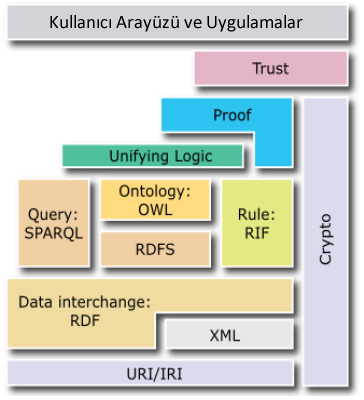
Araştırmacılar farklı mimariler kullanarak anlamsal web uygulamaları gerçekleştirmeye çalışırlarken W3C tarafından önerilen yaklaşımların anlamsal web alanında standartların oluşmasında etkili olacağı tahmin edilmektedir. XML bunu en iyi gösteren standartlardandır.

W3C’ nin anlamsal web için belirlediği temel ilkeler şunlardır:

* Her şey Uniform Resource Identifier’ ler (URI) ile belirtilmelidir.
* Kaynakların ve bağlantıların tipleri olabilir.
* Yarım veya eksik bilgi kabul edilebilir.
* Tam gerçeğe ihtiyaç yoktur.
* Evrim desteklenir.
* Minimalist bir tasarım vardır.
* Anlamsal web katmanlı bir mimariye sahiptir.

Anlamsal web’ in katmanları bir alt katmanın sunduğu servislerden faydalanırken bir üst katmanın kullanması için servisler sunar. Her bir katmanın gelişimi birbirinden bağımsız ilerleyebilirken bu sayede mimari daha esnek olarak geliştirilebilmektedir. Tam ve eksiksiz kabul edilen bir sistem yerine eksikleri bilinen ve zamanla üzerinde güncellemeler yapılabilen yapıla bilen bir sistem yaklaşımı seçildiğinden anlamsal web’ in evrimsel bir yapı geçireceği düşünülmektedir.

**Şekil : Anlamsal web’ in katmanları**



**Kaynak:** (W3ORG, w3.org, 2007)

# URI (Uniform Resource Identifier)

Her şey URI’ ler ile belirtilmelidir. Özne, nesne, yüklem, özellik gibi tüm varlıklar kurallara uygun şekilde isimlendirilmiş, tanımlandırılmış olmalıdırlar. Örnek olarak Kaan, Melek ve Anne kavramlarından, Kaan öznedir, Melek nesnedir, Anne ise Melek ile Kaan arasındaki ilişkiyi gösteren yüklemdir. İngilizce dil yapısı ile Türkçe dil yapısı arasındaki uyumsuzluktan dolayı hemen anlaşılmakta zorluk çeksek de bu örnekte anlatılan Melek’ in Kaan’ ın annesi olduğudur.

# XML (Extensible Markup Language)

KURTEL’ e göre XML’ in 4 konudaki başarısı nedeni ile kullanılmaya devam etmektedir. Bunlar:

* Uygulamalardan bağımsız veri ve belge oluşturma imkânı.
* Üst veri (veri hakkında veri – meta data) gösterimi için standartlar sunması.
* Veri ve belge için ortak yapısal standartlar sunması.
* Sınanmış bir teknoloji olması (KURTEL, 2008).

XML bu özellikleri ile anlamsal web’ in en önemli bileşenlerinden biri olmaktadır.

Örnek bir XML belgesi içeriği aşağıdaki gibidir.

<?xml version="1.0"?>  
<bilisimogrenci>  
<adi>Furkan</adi>  
<soyadi>Dündar</soyadi>  
<adresi>Akyazı/SAKARYA</adresi>  
</ bilisimogrenci>

# RDF

XML dili verilerinin kodlanması ve taşınması için sözdizimi yapısının belirlendiği bir veri modelidir. Bu model ile web ortamındaki nesnelerin (veri kaynaklarının), kaynak özelliklerinin ve özellik değerlerinin tanımlanması sağlanır. RDF sözdizimindeki nesne, özellik, değer üçlüsü RDF’ nin temel yapısıdır.

Örnek bir RDF belgesi içeriği aşağıdaki gibidir.

<*rdf*:RDF

xmlns:*rdf*="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:*s*="http://description.org/schema/">

<*rdf*:Description about="http://www.w3.org/Home/Lassila">

<*s*:Creator>

<*rdf*:Description about="http://www.w3.org/staffId/85740">

<*rdf*:type resource="http://description.org/schema/Person"/>

<*v*:Name>Ora Lassila</*v*:Name>

<*v*:Email>lassila@w3.org</*v*:Email>

</*rdf*:Description>

</*s*:Creator>

</*rdf*:Description>

</*rdf*:RDF> (W3ORG, w3.org, 1999)

# RDFS

RDFS gösterimi RDF veri modelini genişleten bir tip sistemidir. RDF veri modelindeki kısıtları tanımlar, verinin anlamlandırılmasını sağlar. Bu tip sistemi bir alanda kullanılacak olan sözcük kümesini tanımlar.

Örnek bir RDFS belgesi içeriği aşağıdaki gibidir.

Ornk:Dell rdf:type rdfs:Class

Ornk:DellUreticisi rdf:type rdf:Property

Ornk:DellTipi rdf:type rdf:Property

Ornk:DellBellek rdf:range xsd:Integer

Ornk:DellİsletimSistemi rdf:range rdfs:Literal

Ornk: DellBellek rdfs:domain Ornk:Dell

Ornk:DellİsletimSistemi rdfs:domain Ornk:Dell

Ornk:DellSunucu rdfs:subClassOf Ornk:DellTipi

# SPARQL (Protocol And RDF Query Language)

RDF verileri için kullanılan bir sorgulama dilidir.

Örnek bir SPARQL sorgusu sözdizimi aşağıdaki gibidir.

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 . } (WIKI, wikipedia, 2010)

# RIF (Rule Interchange Format)

Kural, bir ya da daha fazla şartın sağlanması durumunda elde edilecek sonucu tanımlar. Kurallar kural katmanında saklanır ve kural yordamı tarafından çalıştırılır. Kural yordamının çalıştırılması ile elde edilen çıkarsamalar ve çıkarsamaların üçlüleri (nesne, özellik, değer) ontolojiye eklenir.

Örnek çıkarsama;

anne(?Kaan,?Melek) ∧ kardes(?Melek,?Merve) *→*teyze(?Kaan,?Merve)

Şartlar Sonuç

# Ontoloji

Ontoloji kelimesinin felsefedeki anlamı varlık bilimi’ dir. İlk olarak Aristo tarafından Ontoloji kavramı ortaya atılmıştır. Aristo, bilginin sistemli bir şekilde saklanmasıyla daha verimli şekilde kullanılabileceğini savunmuştur. Ontoloji, belirli bir alandaki bilgilerin paylaşımı ve yeniden kullanılması için kavramsallaştırma ve açıkça tanımlama üzerinde durmuştur. Ontoloji ile tanımlanan bilgiler üzerinden çıkarsama yapılması için ontoloji ve mantık birlikte kullanılmıştır. Mantığın temellerini de Aristo ortaya atmıştır. Ontolojinin bilgisayar bilimleri alanındaki en çok kabul gören tanımı, “Ontoloji, kavramsallaştırmanın açıkça belirtilmesidir” (GRUBER, 1993) şeklindedir.

Kavramsallaştırma, tasarım aşamasında soyut bir model oluşturmaktır. Oluşturulan bu modelde, geride hiçbir soru bırakmayacak kadar açık tanımlar yapılmalıdır (MAEDCHEA & Staab, Ontology Learning for the Semantic Web, 2001).

Maedche ve Staab kavramları içeren modellerin web ontolojisi sayılabilmesi için hangi özelliklere sahip olmaları gerektiğini tartışmışlardır. Bu özellikler aşağıdaki gibi vermişlerdir;

* Kavram (nesne) sayısı sonlu fakat genişletilebilir olmalıdır.
* Sınıflar ve nesneler arasındaki ilişkiler sadece bir şekilde anlaşılır olmalı farklı anlamlar çıkarılmamalıdır.
* OWL’ nin sınıf yapısını kullanmalıdır.

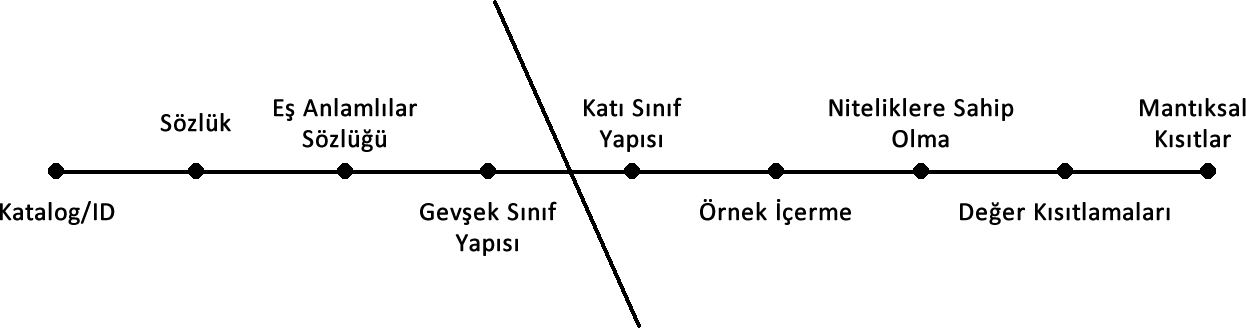
Yukarıdaki temel özelliklere ek olarak;

* Örneklerin olması,
* Sınıf özelliklerine sahip olması,
* Değer karşılaştırma kıstaslarının olması,
* Mantıksal karşılaştırma kıstaslarının olması özellikleri de eklenebilir.

Bu özelliklere sahip ontolojiler, karmaşık bilgileri modelleyip bu bilgilerden sonuç çıkarabilir hale gelebilirler.

McGuinness, web ontolojisinin taşıması gereken özellikleri ve hangi özellikleri içeren modellerin ontoloji sayılabileceğini Ontologies Come of Age isimli makalesinde anlatmıştır (McGUINNESS, 2003). Övünç 2004 yılında sunduğu yüksek lisans tezinde, bu çalışmaları incelemiş aşağıdaki Ontoloji Spektrumunu açıklamıştır.

**Şekil : Ontoloji spektrumu**



**Kaynak:** (ÖZTÜRK, 2004)

# Logic

Logic katmanı kuralların çalışmasıyla sonuca ulaşmak için kullanılan katmandır. Proof katmanının buradan gelecek verilere ihtiyacı vardır.

# Proof ve Trust

Şu aşamada geliştirilmesi henüz çok ilerlememiş katmanlardır. Logic katmanından gelen verileri değerlendirirler.

# Anlamsal Web’ in Teknolojik Bileşenleri

Anlamsal web’ in XML, web servisleri ve Ontoloji olmak üzere 3 teknolojik bileşeni vardır (KURTEL, 2008).

# XML

HTML’ den en önemli farkı, önceden tanımlı olmayan kullanıcılar tarafından oluşturulabilen tag’ ların kullanılabilmesidir. HTML verinin şekli ile ilgilenirken, XML verinin kendisi ile ilgilendiğinden XML belgelerinde verinin tanım bilgilerine geniş verilir. Bu şekilde verilerin değişik ortamlara taşınması amaçlanmıştır.

XML'in avantajları:

* Açık standartları vardır.
* Genişletilebilir XML belgesine kendi etiketlerimizi ekleyebiliriz.
* İşletim sistemine, uygulamanın yazıldığı programlama diline veya veri kaynağına bağlı olmadığından esnek web uygulamalarının geliştirilmesinden kolaylık sağlar.
* Farklı veri kaynakları ve uygulama programları arasında verilerin akışına ve veri bütünleşmesine olanak sağlar
* Veri kaynaklarından sağlanan verilerin değişik görünümlerinin elde edilmesine olanak sağlar.
* Veri kaynaklarındaki veriler üzerinde yerel olarak hesaplama yapılmasını sağlar. Elde edilen veriler tarayıcı tarafından okunur ve ardından sonraki aşama için bir yerel uygulama programına aktarılır.
* Farklı uygulama programları içerisindeki verileri tanımlayabilir. Verinin kendisi kendini tanımlayabildiğinden kaynak uygulama programına bağlı kalmadan veri işlenebilir.
* XML, sadece değişen verilerin güncellenmesi için de olanak sağlar. Değişen verinin görüntülenmesi bütün sayfanın yeniden yüklenmesine ihtiyaç duymaz.

# Web Servisleri

Web servisleri, web standartları kullanarak, farklı programlama dilleri ile farklı işletim sistemlerinde oluşturulan yazılım bileşenlerinin birbirleri ile etkileşimine olanak sağlayan teknolojilerdir (DUTTA, 2008).

Web servisleri, web ortamında belirli görev veya görevleri yerine getiren işlemleri tanımlamak için ara yüz oluşturmaktadırlar. Web Service Description Language (WSDL) kullanılarak tanımlanan web servisleri, servis yayınlama, bulma, bağlanma ve çağırma işlemleri için Simple Object Access Protocol (SOAP)’ ı kullanmaktadırlar (YÜKSEK, 2009).

**Şekil : Web servis mimarisi**



**Kaynak:** (YÜKSEK, 2009)

# Web servis mimarisi yukarıdaki şekilde de görüldüğü gibi 3 temel bileşenden oluşur. Bunlar servis sağlayıcısı, servis kayıtçısı ve servis kullanıcısıdır. Servis sağlayıcısı, servis kullanıcılarının servis sağlayıcısındaki servislere ulaşmasını sağlar. Servis kayıtçısında ise servis sağlayıcısının yayınladığı web servislerinin nasıl çağrılacağı ile ilgili tanımlar kayıt edilir. Servis kullanıcısı, servis kayıtçısındaki kurallara bakarak servis sağlayıcısındaki servisleri çağıran uygulamalardır.

Web servisleri genellikle veri formatı olarak XML’ i kullanırlar.

# SOAP (Simple Object Access Protocol)

Servis kullanıcısı ile servis sağlayıcısı arasında iletişimim hangi kurallar üzerinden sağlanacağını SOAP belirler. Bu iletişimin standartların W3C tarafından belirlenmektedir. XML veri formatını kullandığı için farklı platformlarda hazırlanan uygulamalar arasında iletişime imkân tanır. 3 çeşit SOAP mesajı oluşabilir. Bunlar metod kullanım isteği, cevap mesajı ve hata mesajıdır.

Bir SAOP mesajında en temel olarak 3 bileşen bulunmaktadır. Bunlar Envelope, Header ve Body’ dir (ALGAN, 2011).

Envelope, SOAP mesajlarının kapsayan yapıdır. İçeriğinde body veya header gibi diğer elemanlarda bulunur. Her envelope’ in bir header taşıma zorunluluğu olmasa da mutlaka bir body taşımaktadır. Fakat eğer envelope içinde header bulunuyorsa, header mutlaka envelope’ nin ilk elemanı olmak zorundadır.

Header, meta-data (veri hakkında veri) taşıyan SOAP bileşenidir. SOAP’ ın çağırdığı metod ile herhangi bir ilişkisi yoktur.

Body, web servisi kullanıcısının web servisi sağlayıcısına, metodunun adı ve parametrelerini gönderdiği SOAP bileşenidir. Web servis sağlayıcısının gönderdiği cevap mesajındaki sonuç değeri de body bileşeninde gönderilir. Hata mesajında ise hatanın adı ve tanımı SOAP’ ın body bileşeninde gönderilir.

# WSDL (Web Service Definition Language)

SOAP’ ın farklı platformlarda ve dillerde oluşturulan uygulamaları konuşturması için gerekli olan standartlar WSDL ile oluşturulur. XML veri formatını kullanan WSDL SOAP’ ın isteklerini gerçekleştirebilmek için, web servisinin adı, çağrılabilecek metodlar, bu metodların kullanabileceği parametreler, parametrelerin veri tipleri, geri dönecek cevap ve cevabın formatı, hata oluştuğunda geri dönecek mesaj, web servis sağlayıcılara ulaşılabilecek adres gibi verileri saklar (W3.ORG, 2011).

# UDDI (Universal Description, Discovery and Integration)

Servis kullanıcısı, kendi ihtiyacına uygun bir web servisini nerede bulabileceğini, bu web servisini nasıl kullanabileceğini, UDDI’ ler aracılığı ile öğrenebilir. UDDI’ ler bir çeşit web servisi rehberidir. Fakat henüz tam olarak hazır bir teknoloji olmadığından UDDI’ leri kullanan API’ ler sürekli olarak güncellenmektedirler. Örneğin web servislerini sınıflandırmaktaki sıkıntılar halen devam etmektedir.

# Ontoloji

Ontoloji, bir önalandaki tüm kavramları, bu kavramların tanımlarını ve bu kavramlar arasındaki ilişkileri modeller. Veriyi ve bilgiyi saklamak, ontolojileri geliştirmek ve web ortamındaki nesneleri tanımlayıp kullanmak için ontoloji dilleri geliştirilmektedir (TÜRKYILMAZ, 2008).

2002 yılında W3C tarafından geliştirilen OWL (Web Ontology Language) dili kavramlar arasındaki ilişkiyi göstermek için RDF teknolojisini kullanmaktadır. Kavramlar arasında ilişki kurulduğunda makineler tarafından bilginin anlamlaştırılması kolaylaşmaktadır. Örneğin bir kişi “ben Dell kullanıyorum” cümlesinde Dell kelimesini RDF olarak tanımlaması gerekirse; Dell kelimesinin üreticisi, modeli, belleği, işletim sistemi tanımladığında Dell kelimesine anlam yüklemiş olur. Makine Dell kelimesini okuduğunda bunun bir tanımlanmış kavram olduğunu anlayacaktır.

OWL ile oluşturulan kavram ve ilişkiler sadece tek bir çalışma için kullanılmaz. Bu bilgi başka ontolojiler tarafından da ulaşılabilir, tanımlanabilir, değiştirilebilir ve genişletilebilir. OWL belgelerinin hatalı yorumlanmaması ve bilgisayarlar tarafından doğru işlenebilmesi için standartlara ihtiyaç vardır. OWL belgelerinde kullanılan söz dizim kuralları RDF dosyalarının genişletilmiş halidir.

OWL üç alt dile sahiptir. Bunlar OWL Lite, OWL DL ve OWL Full. OWL dilinin alt dilleri birbirlerini kapsayacak biçimde tasarlanmışlardır (TÜRKYILMAZ, 2008).

* + - * 1. OWL Full

RDF ve RDF(S) in tüm standartlarını tanır. RDF’in tüm avantajlarının kullanımına izin verse de hesaplanabilirliği destekleyemez. Örneğin RDF de tanımlanan sınıf kavramı bir topluluğu ifade edebildiği gibi o bir üst sınıftan çok az farkla ayrılan belki de sadece bir bireyi tanımlayan sınıflar da olabilir. OWL Full tanımlama bilgisinin hesaplama bilgisinden daha önemli olduğu durumlarda tercih edilir. OWL Full de bazen mantık yürütmeler olanaksızlaşır. Dezavantajlarından biri ise OWL Full standartlarının tam olarak kararlı bir yapıya henüz ulaşamamış olmasıdır. Bu nedenle kavramlar arası mantık yürütmede etkili olamayabilir.

Örneğin, Kaan bey bir firmadan oyun konsolu satın aldı. Bu bilgiden yola çıkılarak Kaan beyin gelecekte çeşitli oyunlar satın alacağı tahmini yapılabilir. Fakat Ali Bey’ in satın aldığı oyun konsolunun markası ve modeli belli olmadığından satın alabileceği ürün yelpazesi geniş olabileceğinden etkili bir tahmin yapılamaz. Sonuç olarak yürütülen bu mantık sonucu firma hangi ürüne yatırım yapması gerektiği konusunda fikir sahibi olamamaktadır çünkü firmanın karşısında çok fazla seçenek vardır. Bu nedenle bu tür mantıksal çıkarımlar etkili bir sonuca ulaşmada yeterli veriye sahip değillerdir.

* + - * 1. OWL DL

OWL ve RDF oluşturan yapılara kısıtlar tanımlanmasına imkân vererek terimlerin makineler tarafından daha kolay ve doğru şekilde anlamlandırılmasına olanak sağlar. Hesaplamaya izin vermesiyle mantık yürütmeleri kolaylaştırır. Örneğin Kaan Bey firmadan bir adet Playstation 3 aldı. Mantıksal olarak yakın gelecekte Kaan beyin Playstation 3 oyunları satın alacağı sonucuna ulaşılabilir.

* + - * 1. OWL Lite

OWL DL nin standartlarından yararlanarak kısıtlar tanımlama avantajını kullanır. Fakat OWL Lite ile sadece üst sınıf tanımlanabilir. OWL Lite ile mantıksal sonuçlar değerlendirilerek sadece iki değere ulaşılabilir. Bu değerler 0 ve 1’ dir. Bir durumun göstergesi olarak bir sonuca ulaşılabiliyor ise 1, aksi durumda 0 değerine ulaşılır. Örneğin bir öğretmenin sahip olduğu sicil numarası, verdiği dersler, boş saatleri kontrol edildiğinde ders atanabilir özelliği 1 değeri gönderir, diğer durumlarda 0 değeri üretilir.

Ayrıca ABD tarafından DAML (DARPA Agent Markup Language) ve Avrupa Birliği tarafından OIL (Ontology Interface Layer) ontoloji geliştirme dilleri de geliştirilmişlerdir. ABD ve AB tarafından tanımlanan DAML+OIL dili de 2000 yılında kullanıma sunulmuştur.