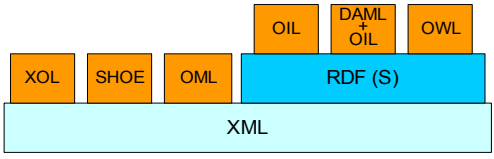
Semantik Web Teknolojileri

İbrahim TÜRKYILMAZ, Cumali YAŞAR
Çanakkale Onsekiz Mart Üniversitesi, Çanakkale
iturkyilmaz@comu.edu.tr

Semantik Web, yapay zekâ olmamakla beraber yapay zekâ teknolojileri kullanabilen bir teknolojidir. Bu teknoloji makinenin anlayabileceği akıllı veriler ortaya çıkarır. Makinenin anlayabileceği akıllı veri kavramı, sadece makinenin mevcut iyi tanımlanmış veriler üzerinde iyi tanımlanmış işlemler yaparak iyi tanımlanmış bir problemi çözebilme yeteneğini ifade eder. Makinelerden insanların dilini anlamalarını istemek yerine, insanlardan bilgiyi daha düzenli tanımlamak için daha fazla çaba göstermelerini istemek akıllı verinin daha kolay oluşturulmasını sağlar (Maedche ve Staab, 2001).

Semantik Web teknolojilerini oluşturan önemli öğeler Şekil 2-8 de şematik olarak ifade edilmektedir. Bu teknolojileri barındıran sisteme Semantik Web Teknolojileri denir. Semantik Web teknolojileri günümüzde uygulamaya konulan WEB 2.0 denilen yapıyı oluşturmaktadır.

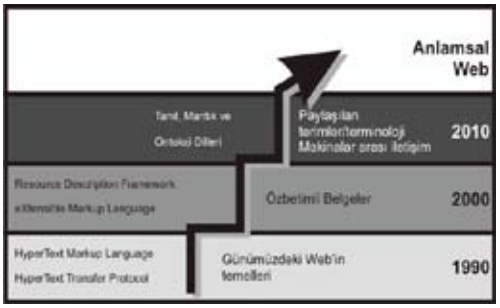


Şekil 2-8 Semantik Web teknolojisi kullanılarak arama yapan sistem

Şekil 2-8 Semantik Web teknolojilerini hiyerarşisini göstermektedir. Bu teknolojinin tüm basamakları birbirleriyle bağlantılıdır. XML hariç hepsinin kendi içinde sınıf modeli mevcuttur. Bu teknolojilerden kısaca bahsetmek gerekirse; RDF yani Kaynak Açıklama Altyapısı (Resource Description Framework) Semantik Web için temel veri modelidir. Bu model, farklı değişim formatları olan RDF/XML, N-Triples, N3 ve Turtle için temel teşkil etmektedir. Semantik Web'in kalbini ise Ontoloji'ler oluşturmaktadır. Bu yüzden de OWL yani Web Ontoloji Dili, tanımlamalar ve veriler arasındaki ilişkileri kurabilmek için kullanılmaktadır (Çankaya, 2007).

Semantik Web'in de tıpkı bilim gibi bir gelişim süreci sonunda oluşacağı gözükmektedir. Yani, en başta mükemmel ve tam olarak işleyen bir sistem yapmak yerine temel işlevleri gerçekleştiren basit ve sağlam bir sistemle başlayarak daha sonra bu sistemi daha karmaşık bir yapıya kavuşturmak hedeflenmiştir. Öncelikle bilginin gösterimi için bir biçim olan HTML üretilmiş ve bu bilginin transferi için http protokolü yazılmıştır. Daha sonra bunların üstüne bu bilgiye ait yapısal bilgisini ve anlamını ekleyebilmek için XML ve RDF dilleri eklenmiştir (Berners-Lee ve diğ, 2001).

Son aşamada ontoloji tanımlamalarını oluşturan, işaretleme dilleri (OIL, DAML+OIL, OWL) verilmektedir. Her aşamada baştaki çekirdek sisteme yeni özellikler ekleyerek sistem genişletilmektedir. Bununla beraber önceden eklenen her sistemde tıpkı son eklenen sistem gibi gelişmesine devam edecektir. Örneğin şu anda OWL Dili ile ilgili çalışmalar sürerken, RDF dili ile de ilgili çalışmalar sürmektedir. Hatta OWL "Candidate Recommendation" aşamasına gelmişken; RDF hala "Working Draft" aşamasındadır. Yani gelişme sürecinde daha geri aşamadadır. Sonuç olarak, Şekil 2-9 görüldüğü gibi Semantik Web'in hayata geçirilebilmesi için yaklaşık 20 yıllık bir sürenin geçmesi gerektiği tahmin edilmektedir.



Şekil 2-9 Semantik web in yıllar içinde bünyesindeki bileşenlerin çizelgesi.

Ontoloji Nedir?

Ontolojil'er Semantik Web'in gerçekleştirilmesinde kullanılacak anahtar teknolojidir. Ontoloji terimi felsefede varlık bilimi olarak tanımlanmaktadır. Ontoloji yirminci yüzyılın sonlarında yapay zekâ alanında yoğun bir şekilde kullanılmıştır. Ontoloji belli bir alandaki bilgilerin paylaşımını ve yeniden kullanımını sağlayan "kavramlaştırmaların biçimsel ve açık tanımlanması" olarak tanımlanmış ve kullanılmıştır. Son zamanlarda ontolojilerin kullanımı zeki sistemler, elektronik ticaret, doğal dil işleme ve bilgi yönetimi konularında yaygınlaşmaktadır.

Ontolojinin bilgisayar bilimindeki en çok kabul gören Gruber'e (1993) ait olan tanımı; "kavramsallaştırmanın açıkça belirtilmesidir" şeklinde ifade edilmesidir. Kavramsallaştırma, belli bir tasarım aşamasında soyut model oluşturma anlamına gelir. Maedche ve Staab (2001)'e göre bu modelin, tasarım aşamasında bilinen tüm bilgilerinin geride hiç bir soru bırakmayacak şekilde açıkça tanımlanması gerekir.

Maedche ve Staab (2001)'de verilen ontolojinin başka bir tanımı da ontolojinin neler içermesi gerektiğini ve neleri içeren modellerin ontoloji sayılması gerektiğini açıklar. Bu tanımın diğerlerinden farkı; burada sözü edilen ontolojilerin web ontolojileri olmasıdır. Her terimin tanımlayıcısı bulunur ve ontolojide bulunan terimler sonlu sayıdadır. Bunlar; terimlerin anlamlarını, terimler arasındaki ilişkileri, terimler arasındaki eşanlamlılıkları ve sıradüzen ilişkilerini içerir. Sonuç olarak, bir kavram kümesinin ontolojisinin olabilmesi için en azından aşağıdaki niteliklere sahip olması gerekir:

- Terim dağarcığının sonlu ve genişletilebilir olması gerekir,
- Sınıflar ve terimler arasındaki ilişkilerin etmenler tarafından tek bir şekilde anlaşılır olması,
- OWL'nin sınıf yapısını kullanması.

Üç temel niteliğin yanında ontolojilerin anlatım gücünü arttırabilmek ve nitelikli sorgulama sonuçları almak için,

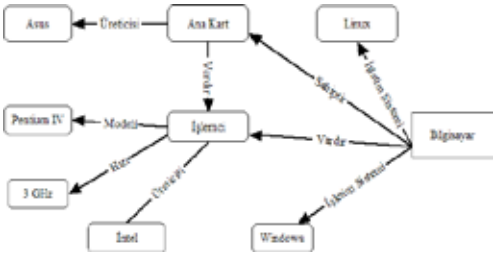
- Örnek içerme,
- Sınıf niteliklere sahip olma,
- Değer karşılaştırma kıstasları,
- Mantıksal karşılaştırma kıstasları,

özellikleri de yukarıda verilen ontoloji olabilme niteliklerine eklenebilir. Bu mantıksal yapıların da dâhil edilmesiyle; ontolojiler, karmaşık bilgileri modelleyebilen ve bu bilgiler üzerinde sonuç üretebilen bir seviyeye gelirler.

Web Ontoloji Dili

Web Ontology Language (OWL), ontolojileri tanımlamak ve çeşitlemek için kullanılan bir dildir (Wikipedia, 2007). OWL, bilginin içeriğini sadece insanlara gösteren değil, bunun yanında bilgisayarlar tarafından işlenebilmek üzere tasarlanmıştır. OWL, XML, RDF ve RDF Schema (RDF-S) tarafından desteklenen web içeriğinin bilgisayarlar tarafından daha iyi yorumlanabilmesini sağlamayı hedef almaktadır. OWL, OIL ve DAML+OIL gibi dilleri temel alır ve şu anda W3C kurumu tarafından tavsiye edilen bir dildir. OWL, DAML+OIL

web ontoloji dilinin araştırma tabanlı bir sürümüdür (Wikipedia, 2007).

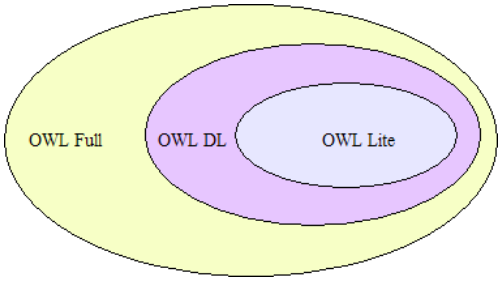


Şekil 2-10 OWL dillerinin şematik yapısı

OWL dilinin RDF olarak şematik kullanımına bir örnek Şekil 2-10 de verilmektedir. Terimler arasındaki bağlantı RDF söz dizimi ile yapılmaktadır. Söz dizim kuralları bilgiyi anlamlaştırmak için tasarlanmıştır. Bunu şu şekilde bir örnek ile açıklayabiliriz. Bir kişi "ben Apple kullanıyorum" cümlesinde Apple kelimesini RDF olarak tanımlaması gerekirse Apple kelimesini; üreticisini, tipini, modelini, işletim sistemini tanımladığında Apple kelimesine anlam yüklemiş olur. Bilgisayar Apple kelimesini okuduğunda bunun bir tanımlanmış bir bilgisayar olduğunu anlayacaktır. Bu tanımlamayı ontoloji dili ile Şekil 2-10 gösterilmektedir.

OWL ile tanımlanan bilgi tek bir amaç için düşünülemez. Bu bilgi başka ontolojiler tarafından da tanımlanabilir veya genişletilebilir. OWL belgelerinin yanlış yorumlanmaması ve bilgisayarlar tarafından işlenebilmesi için düzgün bir söz dizimine ihtiyaç vardır. OWL belgelerinde kullanılacak terminoloji RDF dosyalarının genişletilmiş halidir (Owl, 2007).

OWL üç alt dile sahiptir: Şekil 2-11 de görüldüğü gibi OWL Lite, OWL DL ve OWL Full. Bu üç dil, kullanıcılar tarafından kullanılmak üzere özel olarak tasarlanmışlardır (Owl, 2007). OWL dilinin alt dilleri birbirlerini kapsayacak biçimde tasarlanmışlardır. Şekil 2-11 de bu yapı gösterilmektedir.



Şekil 2-11 OWL Dilini oluşturan dil yapısı

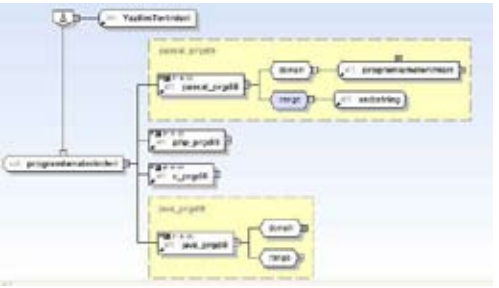
OWL Full: RDF ve RDF(S) bütün bileşenlerini kullanır. RDF'in sağladığı tüm kolaylıkları sağlar. Ancak hesaplanabilirliği desteklemez. Mesela RDF de tanımlanan sınıf bir topluluğu ifade edebildiği gibi bir bireyi tanımlayan sınıf da olabilir. OWL Full daha çok tanımlama bilgisinin hesaplama bilgisinden önemli olduğu durumlarda kullanılabilir. OWL Full de bazı durumlarda otomatik mantık yürütmeler imkânsız olabilir. OWL Full dili RDF ile tam olarak uyumludur. RDF dokümanlarının yapısı aynı zamanda OWL Full dokümanlarıdır. Dezavantajlarından biri OWL Full yapıları tam kararlı değildir. Bundan dolayı kavramlar arasında tam ve etkili bir mantık yürütemez. Örneğin, Ahmet Bey bir bilgisayar malzemesi satan tekno marketten yazıcı almış olsun. Mantıksal olarak Ahmet beyin gelecekte yazıcısı için kartuş, toner veya ribon gibi yazıcı sarf malzemelerinden birini satın alabilecek potansiyel müşteri olabileceği sonucu çıkarılabilir. Satın almış olduğu yazıcı türü belli olmadığından yazıcısı için kullanacağı sarf malzeme çok çeşitli olabilmektedir. Bu şekilde oluşturulan mantıksal ifadede ilgi bağının zayıflığı sonucunda firmanın hangi sarf malzemeye yatırım yapması kararını verebilmesi için karşısına birden çok seçenek çıkmaktadır. Sonuç olarak bu tür mantıksal çıkarımlar tek bir sonuca ulaşmada etkin bir mantık oluşturamamaktadır.

OWL DL: OWL Full'un bir alt dilidir. OWL ve RDF oluşturan yapılarına kısıtlar ekleyerek kullanılır. Bu kısıtlar tanımlama mantığını bağlantısını sağlar. OWL DL, terimlerin anlamlarını tanımlamada kolaylık sağlar. Otomatik hesaplamaları mümkün kılararak akıl yürütmeyi kolaylaştır. Örneğin Ahmet Bey'in teknoloji marketinden bir adet lazer yazıcı aldığını düşünelim. Mantıksal yürütme yaparsak; eğer Ahmet Bey lazer yazıcı almışsa çok kuvvetli bir ihtimalle yazıcısı için gelecekte toner alacaktır. Mantıksal olarak lazer yazıcıya sahip bir kişinin lazer yazıcısı için toner alması gerektiği çıkarımına ulaşılarak etkin bir mantık yürütülmektedir.

OWL Lite: OWL dilinin tek bir kısmını kullanır. OWL DL nin kısıtlarını kullanır. OWL Lite sadece üst sınıf tanımlanabilir. Üst sınıflarında karar yapıları kullanılır. Sınıflar arasında eşitlik ve alt sınıf ilişkileri vardır. OWL Lite mantıksal sonuçlarını değerlendirerek iki değer gönderir. Bu değerler 0 ve 1 dir. Özellikle durum belirteçlerinde; eğer bir çıkarım varsa 1 kabul edilir. Diğer durumlar 0 kabul edilir. Örneğin bir bilgisayarın sahip olduğu işlemci, ana kart, ram belleğine bakıldığında sunucu özelliği varsa 1 değeri gönderir, diğer durumlarda 0 değeri gönderilir.

Ontoloji Nasıl Oluşturulur?

Ontoloji oluşturmak için öncelikle bir alan belirlenmelidir. Bu alan üzerinde sonlu sayıda terim olmalıdır. Bu terimlerin sınıfları ve alt sınıfları belirlenmelidir. Sınıfların bireyleri tanımlanmalıdır. Sınıf tanımlamaları bir referans sınıfta olabilir. Ancak alt sınıflar referans sınıf olamaz. Alt sınıf tanımlamaları genelde daha özel bireyleri tanımlamak için kullanılır. Örneğin bilişim ontolojisi oluştururken sınıf olarak yazılım terimlerini alt sınıf olarak nesne yönelimli programlama dilleri terimlerini (nesne, sınıf, fonksiyon) yazabiliriz.



Şekil 2-12 Sınıf yapısının gösterimi

Ontoloji oluştururken çok önemli bir tanımlama da sınıfların özelliklerini belirtmekten geçer. Her sınıfın kendine özgü belirli bir özelliği vardır. Sınıflara özellik ataması yapılırken veri tipleri ve nesne özellikleri ayıca belirtilebilir. Şekil 2-12 görüldüğü gibi özellik belirtilirken, terimin üst sınıfı ve bağlantı adresi belirlenmektedir. Bu üst sınıfa domain denir. Bağlantı adresine ise range denir. Ayrıca domain kavramı programlama tipleri arasındaki turtalılığı kontrol etmek için kullanılır. Range kavramı ise terimin bağlantı noktalarını belirler. Aynı zamanda domain kavramı sınıflardan bilgi çıkarımı içinde kullanılabilir. Her sınıfın bir domain alanı vardır. Bu domain bir URI adresi olabilir. Ontoloji oluştururken semantik bağlar sınıflara yansıtılır. Bu yansıtma aidiyet eylemini gerçekleştirir. Birey sınıfa aidiyetlerini ancak bir eylem ile belirtir. Bu eylemler sahiplik eki iledir.

Basit Anlamda Örnek Bir Ontoloji Şeması ve Ontoloji Dili Oluşturulması:

Şimdiye kadar bir ontoloji oluşturmayı teorik olarak açıkladık. Uygulamalı olarak bir ontoloji nasıl oluşturulabileceğini göstermeye çalışacağız. Örnek ontoloji oluşturacağımız alan bilişim terimleridir. Alan belirlendikten sonra bilişim terimlerinin sınıfları ve bu sınıflara ait terimleri ilgi alanlarına göre gruplayalım. Biz sınıf olarak:

- Network Terimleri
  (DNS, IP, RJ45, …)
- Yazılım Terimleri
  (Nesne, Fonksiyon, Koşul, …)
- Donanın Terimleri
  (Ağ Kartı, İşlemci, Bellek, …)

olmak üzere üç tane sınıf tanımlayalım. Bu tanımlamaları şematik olarak belirledikten sonra ontoloji oluşturma araçlarını kullanmaya sıra gelir. Ontoloji oluşturma aracı olarak Protégé 3.3.1 sürümü kullanılabilir. Protégé programı, ücretsiz olarak dağıtılmaktadır. Terimlerin sınıfları belirledikten sonra; eğer gerekiyorsa bu sınıfların alt sınıflarını belirlememiz gerekir. Örneğin Donanım Terimlerinin alt sınıflarını

- MİB terimleri
  (Ön Bellek, CPU, RAM Bellek, …)
- Bellek Terimleri (Statik Bellek, Dinamik Bellek, Swap Bellek, …)
- I/O terimleri
  (Yazıcı, Tarayıcı, Monitör, …)

olmak üzere alt sınıflara ayırabiliriz.

Alt sınıflar tanımlandıktan sonra terimlerin özellikleri ve ilişkileri belirlenir. Daha sonra bu ilişkilerin eylemleri tanımlanır. Örneğin; "İşlemci bellek kullanır. Ekran kartının belleği vardır. C++ bellek kullanır. Oyun yazılımları dinamik belleği ve ekran kartının belleğini kullanır" cümlelerinin terim olarak bellek terimi üzerinde bağlantılarını yapabiliriz. Tanımlamalar ve ilişkiler belirlendikten sonra ontoloji dili kullanılarak ontoloji oluşturulmaya başlanır. Şekil 2-13 te görüldüğü gibi sınıflar ve alt sınıflar belirlenmiştir. Alt sınıflara ait terimlerin tipleri ve bağlantıları kod olarak verilmiştir.

Terimlerin bağlantıları kod içerinde içinde URI şeklinde yazılmaktadır. Bağlantı şeklini örnek koda bakarak inceleyebiliriz (Kod:2.1).

Şekil 2-13 Sınıflar ve alt sınıflar oluşturulması

Kod :2-1

```
<?xml version="1.0"?>
<rdf:RDF xml:base="http://ww.owl-
ontologies.com/unnamed.
owl" xmlns="http://www.
owl-ontologies.com/unnamed.
owl#" xmlns:owl="http://
www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.
w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:rdfs="http://www.
w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://
www.w3.org/2001/XMLSchema#">
```

1

```
<owl:Ontology rdf:about=""/>
<owl:Class
rdf:ID="DonanimTerimleri"/>
<owl:Class
rdf:ID="bellekterimleri">
<rdfs:subClassOf
"rdf:resource
="#DonanimTerimleri"/>
</owl:Class>
<owl:Class
rdf:ID="NetworkTerimleri"/>
<owl:Class
rdf:ID="veritabaniterimleri">
      <rdfs:subClassOf>
<owl:Class
rdf:ID="YazilimTerimleri"/>
      </rdfs:subClassOf>
      </owl:Class>
<owl:Class
rdf:ID="algoritmaterimleri">
      <rdfs:subClassOf
rdf:resource="#Yazilim
Terimleri"/>
</owl:Class>
```

2

```
…
<webterimleri rdf:ID=”ipdns”/>
       <webterimleri
rdf:ID=”ipipx-spx”/>
<veritabaniterimleri
rdf:ID=”join”/>                    } 3
<veritabaniterimleri
rdf:ID=”innerjoin”/>
<veritabaniterimleri
rdf:ID=”function”/>
…
<rdf:Description
rdf:ID=”bilisimterimleri_
Instance_6”>
<rdf:type>Cls(:INSTANCE-
ANNOTATION,
FrameID(0:1020 0))</rdf:type>
<programlamaterimleri
rdf:ID=”java”/>
<veritabaniterimleri
rdf:ID=”leftjoin”/>                } 4
<veritabaniterimleri
rdf:ID=”update”/>
<algoritmaterimleri
rdf:ID=”while”/>
<programlamaterimleri
rdf:ID=”c”/>
<programlamaterimleri
rdf:ID=”owl”/>
</rdf:RDF>
```

Kod:2.1’i incelediğimizde 1. parantez XML Name Space tanımlamalarının yapıldığı yer, 2. parentez Sınıfların tanımlandığı yer, 3. parantez alt sınıfların tanımlandığı yer. 4. parantez bağlantıların tanımlandığı yer olarak incelenebilir.

Sonuç

Ontolojiler kavramsal olarak algoritma mantığı ile çalışmaktadır. Algoritmanın yapısından farklı olarak nesneler arasındaki bağlantı bir eylem ile belirlenmektedir. Bu eylemler ontolojide terimlerin birbirlerini nitelemesi şeklindedir. Ayrıca eylemi yapan cümle; özne, nesne ve yüklem şeklinde sıralanmaktadır. Örneğin, kediler balık sever. Yaşlı insanların genelde bir kedisi vardır. Öyleyse yaşlı insanlar, kedilerine balık alır.

Günümüzde ontoloji oluşturma işi genelde bilgisayar yazılımcıları tarafından yapılmaktadır. Ontoloji oluşturulması yapılacak işlerin tanımlama ve sonuç alma süreçlerini kısaltmaktadır. İleride ontoloji kullanımının yaygınlaşmasıyla kurumların yapısal dönüşümlerinde, görev tanımlamalarında ve iş paylaşımlarında kurumların karar verme mekanizmaları daha hızlı çalışacağı beklenmektedir.

Cimiano (2006)’ya göre günümüzde en popüler ontolojiler WordNet Ontolojisi, Wiki Ontolojisi, Eclips Ontolojisi, Medical Ontoloji, Gen Ontolojisi sayılabilir. Bu ontolojilerin bir ortak özelliği; standart olarak kabul görmeleri ve açık kaynak kod prensiplerine uygun olarak dağıtılmalarıdır.

Kaynaklar

Akyokuş S., (2003). Bilişim Kurultayı. (15 Mart 2007), http:// http://www.akyokus.com/Presentations/

Berners-Lee, T., Hendler, J. and Lassila O. 2001. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284 (5): 34-43.

Cardoso, J. and A. Sheth. 2003. Semantic e-Workflow Composition. Journal of Intelligent Information Systems 21(3): 191-225.

Cimiano, P., 2006. Ontology Learning and Population from Text, Algorithms, Evaluation and Applications. Springer Press Inc., Philipp. 347 p.

Çankaya, N. (1 Mart 2007). Semantik Web 3.0. Temmuz 2007, http://www.nuricankaya.com/

Davies, J., Fensel, D., Harmelen, F. 2003. Towards The Semantic Web. John Wiley&Sons, Canada. P: 10-31.

Gruber, Tom R. 1993. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.

Hammond, B., A. Sheth, 2002. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content. Real World Semantic Web Applications. lOS Press: p:29-49.

Kaon, 2001 KAON is an open-source ontology management infrastructure targeted for business applications, March 2007, http://kaon.semanticweb.org/

Gürel O. ve Göksu O. 1964, Kara Kuvvetleri Komutanlığı tarafından Elektronik Hesap Merkezi Yayınları: EHMY-2, Ankara, (185 .)

Kim, W. and J. Seo. 1991. Classifying schematic and data heterogeinity in multidatabase systems. IEEE Computer 24(12): 12-18.

Klein, M. and A. Bernstein (2001). Searching for Services on the Semantic Web Using Process Ontologies. International Semantic Web Working Symposium (SWWS), Stanford University, California, USA.

Maedche, A. and Staab, S. 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16 (2): 72–79.

Maedche, A., Staab, S. 2000. Semi-automatic Engineering of Ontologies from Text. In: Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering.

Owl (2007). OWL Web Ontology Language Reference, Retrived 2007, Recommendation, World Wide Web Consortium, http://www.w3.org/TR/owl-ref/. 2004.

Protégé, 2007, Protégé is a free, open source ontology editor and knowledge-base Framework, Retrived 2007, http://protege.stanford.edu/.

Smith, B. and Welty, C. 2001. Ontology: Towards a New Synthesis, Proc. Formal Ontology in Information Systems (FOIS 2001). *ACM Press*, New York. 3–9.

W3C, (2007). The World Wide Web Consortium. (Mart 2007). http://www.w3.org/.

Wikipedia, (2003). Açık kaynak kodlu ansiklopedik sözlük. (Eylül 2007), http://tr.wikipedia.org.