

# ANLAMSAL YAZILIM MÜHENDİSLİĞİ SÜREÇLERİ

Bariş ULU\* ve Banu DİRİ\*\*

(\*) Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, 34349, İSTANBUL

(\*\*)Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, 34349, İSTANBUL  
baris47@gmail.com, banu@ce.yildiz.edu.tr

## ÖZET

Web sadece insanlar arası ilişkiler için bir veri havuzu değil aynı zamanda makinalar arası iletişimi de sağlayabilecek bir mimaridir. Bu veri havuzu söz dizimsel kurallar ile sınırlı kalan ilişkiyel veri depolarından oluşmaktadır. Anlamsal Web, bugün birer çöplük halini almış ve birbirini tekrar eden verilerin kullanımını sağlayan popüler bir kavram halini almıştır. Bu noktadan yola çıkarak, geleneksel yazılım mühendisliği süreçleri arasındaki söz dizimsel ilişkilerin oluşturduğu süreç modellerindeki eksiklikler yazılım mühendisliği süreçlerinin anlamsal olarak modellenmesi gerekliliğini doğurmuştur. Bu sayede, yazılım mühendisliği tecrübeleri zaman ve kaynakların verimsiz kullanımını engelleyerek yeniden kullanılabilir, anlamsal ilişkilerin çıkarılması ile pek çok yazılım mühendisliği süreci zamanla olgunluğa erişebilecektir. Bu fikirden yola çıkarak, bu bildiride yazılım mühendisliği süreçlerindeki söz dizimsel zaafiyetler ele alınarak anlamsal ilişkilerin gerekliliği tartışılmış tartışmayı destekleyici alternatif bir süreç modeli tanımlanmıştır.

**Anahtar Kelimeler:** Yazılım Mühendisliği Süreçleri, Anlamsal Web, Ontoloji Modelleme, Yazılım Mühendisliği Süreç Ontolojisi.

## SEMANTIC SOFTWARE ENGINEERING PROCESSES

### ABSTRACT

Web is designed as a data pool not only for human communication, but also for machine communication. On the other hand, the main problem of this approach is the usage of data that has been defined syntactically on several databases. Semantic Web is a popular concept to prevent interaction with garbage data everywhere. In order to offer an alternative model for the traditional software engineering process models to overcome the semantic gap among the software processes that causes inefficient time and resource consuming during software development, Semantic Web approach is to be used for the success of the software projects that highly depend on the success of the software management and the quality of the software engineering processes as well. Defining the semantic relations among the software processes and the software management is important for this achievement. In this paper, in order to make use of the previous projects experience, the role and the importance of modeling the semantic relations between software management processes and the other processes is discussed.

**Keywords:** Software Engineering Processes, Semantic Web, Ontology Modeling, Software Engineering Process Ontology.

## 1. GİRİŞ

Anlamsal Web verilerin oluşturduğu bir doku ya da veriler ile çevrilmiş bir topluluktur. Günümüzde kullandığımız ancak çoğu zaman farkında bile olmadığımız veriler vardır. Bu veriler Web 'in bir parçası değildir; uygulamalar tarafından yönetilirler ve bir veri ağının parçası değildir. Çoğu zaman uygulamalar bu verileri kendilerine saklarlar.

Anlamsal Web iki noktayı ön plana çıkarmaktadır: Birincisi verilerin değişiminde ortak bir şekil belirlenmesi ki günümüzde orjinal Web' de sadece ve sadece belgelerin değişimi söz konusu olmaktadır. Anlamsal Web' in öne çıkardığı bir diğer husus ise verinin gerçek yaşamdaki nesnelere nasıl bağlı olduğunu gösteren ve bunu kaydedebilen bir dilin belirlenmesidir. Bu

sayede, bir insan ya da makina bir veri deposundan hareketle sonu olmayan bir veri deposu ağı içerisinde kablolar ile bağlanmamış ancak aynı şeyleri ifade eden nesnelere yolu ile bağlanmış veri depoları arasında anlamsal bütünlüğü sağlayarak global olarak linklenmiş veri depolarından oluşmuş tam, kesin ve doğru arama sonuçlarının elde edilebileceği bir ortam haline almıştır.

Bu bildiri, giriş, sonuç ve kaynakça dışında iki ana ve gelecek çalışmayı anlatan bir ek bölümden oluşmaktadır. İkinci bölümde Anlamsal Web hakkında çokça bilinen mimarisine dayalı olarak kısaca bilgilendirme yapılmaktadır. İkinci bölüm, ilişkisel veri modellerine alternatif olarak ontoloji modelleme ile sona ermektedir. Üçüncü bölümde yazılım mühendisliği süreçleri ve bu süreçlerdeki söz dizimsel ilişkilerin yarattığı sıkıntılar ele alınmıştır. Genellikle yazılım yönetim süreci üzerine yoğunlaşarak süreçler arası anlamsal ilişkiler kabaca tanımlanmış, bildirinin konusu olan fikri destekleyici bir model sunulmuştur.

## 2. ANLAMSAL WEB

Günümüzde Web, kullanıcılarına standartların ötesine geçmeyen, belirli yapılarda oluşturulmuş veri depolarında çöplük haline almış, mantıksal işlemesi yapılmayan veriler sunmaktadır. Buradaki temel problem, farklı noktalarda bulunan verilerin mantıksal ilişkilerinden yoksun olarak kullanıcılara sunuluyor olmasıdır. Verinin sunumu önemlidir ve bu sunum ilişkili verilerin anlamsal ifadelerinin belirlenmesi, yapılandırılması ve sunulması ile mümkün olacaktır. Kısacası, bir verinin sunumunda bazı veriler A, bazı veriler B noktasından, vb. şekilde mümkün olacaktır. Her ne kadar Web bir bilgi kümesi olarak tasarlanmış olsa da insan-insan arasındaki ilişkilerden öteye gidememektedir. Bir insanın yayınladığı veriyi içerik ve söz-dizimi geçerli olmak kaydı ile bir başka insan ulaşabilmektedir. Web’ de her ne kadar iyi tanımlanmış verilerin anlamsal içerikleri de veri depolarında tutulabiliyor ya da sorgulanabiliyor olsa da bu veriler delile dayalı ya da ispatlı yapılar içermemektirler; Anlamsal Web, Web’ de gezinen bir robot şeklinde çıkarsamalara izin verecektir [5].

Günümüzdeki Web’ de varolan sorunları aşağıdaki şekilde özetlemek mümkündür [24]:

- Web üzerinde bulunan bilgilerin yapılandırılmış olmaması
- Dinamik ve dağıtık bir ortam olması
- Çok hızlı bir şekilde büyümesi

Web ilk olarak 1994’ de Tim Berners-Lee<sup>1</sup> tarafından Uluslararası Web Konferansı<sup>2</sup>,’ nda dünya üzerindeki Web bağlantılarının sürekli boş duran, sadece istekleri yönlendiren birer araç oldukları varsayımından ortaya çıkarak ortaya atılmıştır. Temel amaç ya da fikir, çoğunlukla boş duran sanal Web bağlantılarının makineler tarafından işlenen verileri diğer veri noktalarına aktarmasını sağlamaktır. Bu sayede Web’ de yapılan her işlem bir başka veri noktasındaki bir başka işleme kaynak olacaktır.

Anlamsal Web, gelişen bilgi (knowledge) kümelerinin oluşturduğu, herkesin bildiği ya da global olarak sorularına net ve doğru yanıtlar bulabildiği bir veri deposudur [16]. Bu bilgi kümeleri sadece ortam (media) nesnelere (Web sayfaları, resimler, görüntülü içerikler, vb.) değil aynı zamanda insanlar, yerler, organizasyonlar ve olayları da kapsamaktadır [15].

Veri, HTML dosyaları vasıtası ile gizlenmiş, bazı içeriklerde kullanışlı olan bazı içeriklerde ise kullanışlı olmayan yapılardır. Bu sebepten dolaydır ki günümüzde orijinal Web ortamında bulunan veriler, global olarak yayınlanmayan ve herkes tarafından işlenemeyen veriler haline almıştır. Örneğin, yerel spor aktiviteleri, hava durumu bilgileri ya da televizyon yayın akışı bilgileri pek çok Web sayfası tarafından yayınlanmakta ancak tüm bu veriler HTML olarak tutulduklarından global sorgularda yer almamaktadırlar. Kısacası, yerel olduğu bilinen veriler global olarak işlenememekte, dağıtık bir yapıda yayınlanamamaktadır [25] [17] [2].

Anlamsal Web’ de iki adet veri tipi mevcuttur: İlkel veriler ve türetilmiş veriler. İlkel veriler işlemciler tarafından anlaşılabilen verilerdir. Çıkarsama dilleri, ispat dilleri, vb. gibi.

<sup>1</sup> <http://www.w3.org/People/Berners-Lee/>

<sup>2</sup> <http://www.iw3c2.org/>

Türetilmiş veriler ise ilkel verilerden çıkarsama yolu ile elde edilmiş verilerdir [26]. Türetilmiş verilerin anlamlı veriler haline gelebilmesi için ilkel veriler üzerine bu verilere ilişkin verilerin tanımlanması ya da veri hakkında veriler oluşturulması gerekmektedir. Veri-hakkında-veri kısaca üst-veri (meta-data) [32] [30] anlamına gelmektedir. Anlamsal Web’ de temel fikir, üst-veri modelleri arasındaki iletişimi sağlayabilmektir [27]. Üst-veriler ancak ve ancak başkaları ile paylaşıldığı noktada Anlamsal Web’e katkıda bulunabilmektedirler; aksi halde üst-verilerin elde bulunan veriler üzerine yerel ilişkileri tanımlamaktan öteye bir anlamı olmamaktadır [1]. Nesnelerin birbirleri ile bağlanması yolu ile Anlamsal Web, insanlar, yerler ve kavramlar üzerine önerilerde bulunabildiği gibi uygulamaların otomatize olması, entegrasyonu ve verilerin yeniden kullanılmasına da yol açmaktadır; bir başka deyişle bir yap-bozun taşlarını uygun yerlere yerleştirmektir [27] [12].

Anlamsal Web’ de yapı oldukça basittir. Bir veri ile diğer bir verinin arasındaki ilişkiyi ifade eden, tanımlanabilir (descriptive) ifadeler yer almaktadır: “Ayşe’nin Fatma adında bir annesi vardır...” ya da “Fatma, Ayşe’nin annesidir...”, gibi [16]. Bu şekilde tanımlanmış veriler orjinal Web’ de HTML tarafından ifade edilememekte, HTML’in geleceği ya da türevi olarak benzer işaretleme dili ile farklı özelliklerde geliştirilmiş olan RDF<sup>3</sup> (Resource Description Framework) [25] [21] [29] [35] [28] [26] işaretleme dili ile sağlanabilmektedir. Doğal olarak “Ayşe” gibi isimler tüm evrende tekil bir kişiyi ifade edecek nitelikte isimlendirmeler değildir. Bu sebepten, Anlamsal Web, bu tekilliği sağlayabilmek adına daha karmaşık ve tekillik garantisi olan isimlendirme şekilleri kullanmaktadır. Bu isimler kısaca bildiğimiz Web adresleri ya da diğer bir ifade ile URL (Uniform Resource Locator) [6] [28] [26] ya da URI (Uniform Resource Indicator) [7] [28] [26] tanımlamalarıdır.

Anlamsal Web, makinaların insan aklını simule etmesinde rol alan yapay zekadan farklıdır. Sadece bir makinanın iyi-tanımlanmış bir problem karşısında iyi-

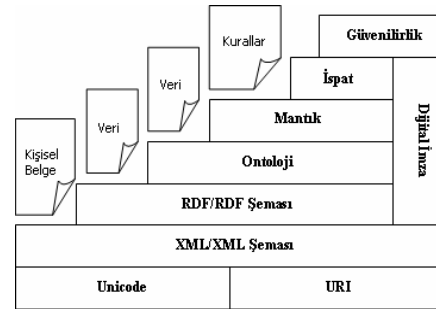
tanımlanmış işlemlerden geçerek iyi-tanımlanmış verileri elde edebilmesi, makinalara insan dilini anlayabilmekten çok insanlara gerekli eforu harcayabilmeleri için gerekli ortamı sağlayabilmekten sorumludur [8]. Bunun yanında hedef sadece doğru veriye ulaşmak değil, aynı zamanda da farklı kaynaklardan gelen verilerin ne zaman entegre edileceği ve hangi verilerin karşılaştırılacağı bilgilerinin de elde edilmesidir [19].

Anlamsal Web’ in prensipleri aşağıdaki şekilde özetlenebilir [19]:

- Her türlü veri bir URI vasıtası ile ifade edilebilir.
- Kaynak ve bağlantıların (link) tipleri vardır.
- Parçalı bilgi tolere edilebilir.
- Gerçek doğru için gereklilik yoktur.
- Evrimsellik desteklenmelidir.
- Basit tasarım yapılmalıdır.

## 2.1. Anlamsal Web Mimarisi

Anlamsal Web’ in gerçekleştirimi için önerilen mimaride URI’ lar en önemli bileşenlerdir. Her URI farklı bir nesneyi ya da kurulacak ilişkiyi ifade etmektedir. URI’ larda yola çıkarak XML belgelerinin ve XML şemalarının oluşturulması üst-verilerin RDF’ ler yolu ile tanımlanmasına ve taşınabilirliğinin sağlanmasına yardımcı olmaktadır. RDF’ lerin oluşturduğu ilişkilendirilebilir veri kümeleri de ontolojiler<sup>4</sup> yolu ile tutulmaktadır.



Şekil 1. Anlamsal Web mimarisi [12] [4]

Şekil 1’ de yer alan mantık katmanı, ontolojilerde tutulmakta olan

<sup>3</sup> Bkz. Bölüm 2.2

<sup>4</sup> Bkz. Bölüm-3

ilişkilendirilebilir verileden taşınabilir bir dil vasıtası ile çıkarsama yapılmasıdır. Başka bir deyişle, iyi tanımlanmış nesnelere söz-dizimsel olarak erişilemeyecek yeni verilerin elde edildiği katmandır [12]. Çıkarsama katmanında güçlü bir mantıksal dilin kullanılması şarttır. Her ne kadar RDF ile miktarsal ya da söz-dizimsel kurallardan uzaklaşmış olsa da mantıksal çıkarsamaların yapılacağı “dayandırma mantığı (predicate logic)” ile matematiksel temellere dayanan mantıksal hesaplamalar yapılması gerekmektedir [25].

Mantık katmanı üzerinde bulunan ispat safhasında, çıkarsama yolu ile elde edilmiş olan verilerin ispatı yapılarak sonuca ulaşılmaktadır. Günümüzde, ispat ve güvenilirlik katmanı üzerine araştırma çalışmaları devam etmektedir [25] [12] [4].

## 2.2. RDF (Resource Description Framework)

Anlamsal Web, veriyi sunabilmek için URI tanımlarını kullanan söz dizimleri üzerine kurulmuştur. Bu söz-dizimlerine “üçleme (triple)” adı verilmektedir. URI’lerin üçlemeleri veritabanları ya da Web’de değiştirilebilir (interchangable) söz-dizimleri şeklindedir. Bu söz-dizimlerine kısaca “RDF” söz-dizimleri adı verilmektedir. Bir üçleme kısaca üç adet URI’den meydana gelmektedir. Bilginin RDF’e aktarılmasının ardından bilginin ayıklanması (parse) daha kolay hale gelmektedir. Örneğin, “Bu sunumun başlığı Anlamsal Web’dir ve Barış Ulu adında bir kişi tarafından hazırlanmıştır...” şeklinde bir ifadenin RDF olarak ifadesi aşağıdaki şekilde oluşturulabilir [25]:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:foaf="http://xmlns.com/0.1/foaf/" >
  <rdf:Description rdf:about="">
    <dc:creator rdf:parseType="Resource">
      <foaf:name>Baris Ulu</foaf:name>
    </dc:creator>
    <dc:title>RDF</dc:title>
  </rdf:Description>
</rdf:RDF>
```

Yukarıdaki RDF’ in ürettiği üçleme ise [25];

```
<> <http://purl.org/dc/elements/1.1/creator> _:x0 .
this <http://purl.org/dc/elements/1.1/title> "RDF" .
_:x0 <http://xmlns.com/0.1/foaf/name> "Baris Ulu" .
```

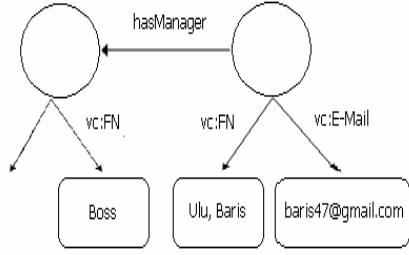
şeklinde olmaktadır. RDF tanımlarının içinde bulunan URI’ lar vasıtası ile Web sayfalarından varolan bilgiyi almanın yanında Web sayfaları ile konuşabilir hale gelmektedir. RDF’ ler bunun yanında ifadelerin makina-işlenebilir (machine-processable) hale gelmesini sağlamaktadır. Makinalar, insanın ne istediğini anlayabilir hale gelmektedir. Aşağıdaki RDF satırı ele alındığında ilk bakışta ne ifade etmek istediği çok açık değildir [28]:

```
<http://barisulu.com/>
<http://love.example.org/terms/reallyLikes>
<http://www.w3.org/People/BarisUlu/RDF/> .
```

Birinci URI konuyu ifade etmektedir. Bu satırda konu Baris Ulu’ yu ifade etmektedir. İkinci URI ise “dayandırma (predicate)” ifadesidir. Konu ile nesneyi birbirine bağlamaktadır. Dayandırma bağlacı bu ifadede “reallyLikes” olarak ifade edilmiştir. Üçüncü URI ise nesneyi yani RDF’ i ifade etmektedir. Bu URI’ ların bir araya geldiği noktada ortaya “Baris Ulu really likes RDF” gibi basit bir ifade ortaya çıkmaktadır [28].

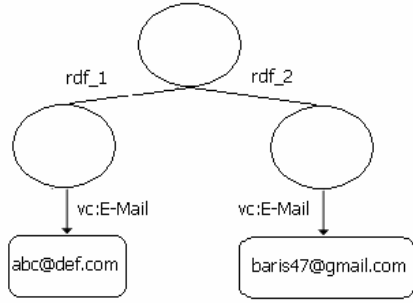
Dünya üzerinde çok değişik makina-işlenebilir veriler içeren veri depoları bulunmaktadır. Hükümetlerin kayıtları, organizasyonların bütçe bilgileri, envanter kayıtları, adres defterleri, vb. gibi. Bir veri bir veri deposunda saklandığı andan itibaren makinalara veri hakkında sorular sormak çok kolay hale gelmektedir. “Son 6 ayda tutuklananların listesi...”, “Envanter listesi kayıtları...” ya da soyadı Ulu olan kişilerin telefon numaraları...”, gibi. Bu noktada RDF bu veri depolarının Web’ de yayınlanmasına aracılık etmektedir. Veri depolarındaki her kayda bir URI verilerek diğer tüm insanların bu kayıtlar ile iletişime geçmesi sağlanacaktır. Sonuç olarak makinalar “Geçen hafta tutuklanan Ulu soyisimli kişinin A organizasyonundaki envanter kaydından telefon numarası bilgisinin elde edilmesi...” şeklinde veri bağlaçları ya da çıkarsamaları ile daha geniş kapsamlı ve daha net veriler elde edeceklerdir. Bu sayede, “Geçen altı ayda tutuklanmış olan A organizasyonundaki X envanter kaydının sahibi kişilerin telefon

numarası...” sorgusu daha anlamlı hale gelecek ve daha net sonuçlar verecektir [28].



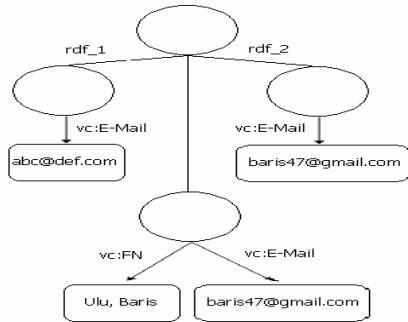
Şekil 2. RDF veri modeli (1) [20]

Şekil 2’ de bir organizasyona ait LDAP kayıtları ve bu kayıtlara ait özellikler bulunmaktadır.



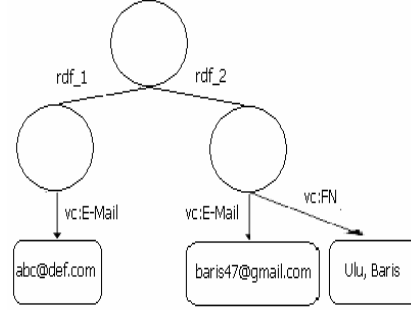
Şekil 3. RDF veri modeli (2) [20]

Şekil 3’ de bir organizasyondaki elektronik posta listesi modellenmiştir. Kişisel elektronik posta adresleri birer RDF belgesi olarak sunulmuştur. Şekil 2 ve Şekil 3’ deki iki modelin bir araya getirilmesi ile Şekil 4’ deki RDF veri modeli elde edilmektedir.



Şekil 4. RDF veri modeli (3) [20]

Şekil 4’ deki model ile birlikte iki farklı veri modelinin bir araya getirilmesi yolu ile Şekil 5’ deki RDF veri modeli elde edilmektedir.



Şekil 5. RDF veri modeli (4) [20]

RDF kısaca insanların makina tarafından işlenebilir veriler yaratmasına yardımcı olan bir modeldir. Anlamsal Web ise bu modelin üstüne kurulacak olan soyut mimaridir. RDF ile tanımlanmış olan veriler her zaman “Anlamsal Web’ lik” değildirler. Bu verilerin temel özelliği Anlamsal Web işlemcilerine uygun halde olmalarıdır [26].

### 2.3. Anlamsal Web ve İlişkisel Veritabanları

Anlamsal Web’ in veri modeli ilişkisel veritabanlarındaki ilişkisel model ile birebir bağlantılıdır. Bir ilişkisel veritabanı tablolar, tabloların satırları ya da kayıtlardan oluşmaktadır. Her kayıt bir ya da daha fazla sahaya sahiptir. İlişkisel veritabanlarında sahaların içerikleri olmadan kayıt bir anlam taşımamaktadır. Benzer şekilde de bir RDF noktasındaki bağlantılar önem taşımaktadır [8]:

- RDF noktası bir kaydı temsil etmektedir.
- RDF özellik tipi (propertyType) bir saha ya da kolonu temsil etmektedir.
- RDF değeri bir kaydın sahasını temsil etmektedir.

Anlamsal Web’ in en temel ve önemli özelliği her zaman için ifadelerdir. Bu ifadeler çöplük halini almış veritabanları üzerinde ilişkisel verilerin elde edilmesine yardımcı olmaktadır [8].

İlişkisel veritabanları atomik (unstructured) veri tiplerine sahiptir. Birleştirme (combination) kuralları zayıftır, veri tipleri uymak koşulu ile her türlü tablo anlamsal karşılığı gözetmeksizin başka bir tablo ile birleştirilebilir. Örneğin; bir ilişkisel

veritabanında, bir çalışanın ayakkabı numarası kadar oda sayısı olan evlerin listesi alınabilir; ancak bu listenin anlamsal karşılığı da her zaman mantıklı olmayabilir [8].

Anlamsal Web, yeni bir veri modeli olarak tasarlanmamıştır. Farklı veri modellerinin birleşimine olanak vermektedir. Merkezi veri depoları yoktur, veriler dağıtık biçimde modellenmiş üst-veriler yardımı ile elde edilmektedir [9] [23]. En çarpıcı özelliği de farklı veritabanlarına bilgi eklenebilmesine olanak sağlaması ve bunlar arasında sofistike işlemler yapılmasına olanak sağlamasıdır [1] [8].

## 2.4. Ontolojiler

İnsanlar, organizasyonlar ve uygulama sistemleri birbirleriyle sürekli bir iletişim halindedirler. Değişik ihtiyaçlara ve içeriklere bağlı olarak ortak iletişim dilinin sağlanması farklı kabullenmelere bağlı olarak değişiklik gösterecektir. Bu da karşılıklı iletişimde farklı jargonların varlığı sebebiyle farklı kavramlar, yapılar ve metodların varlığını da beraberinde getirecektir. Bu da;

- Zayıf iletişime
- Bir alandaki gereksinimlerin belirlenmesinde zorluklara, ve
- Tekerleğin yeniden keşfedilmesi için fazladan efor harcanmasına

neden olacaktır.

Bir alandaki anlamsal ilişkiler, insanlar ve sistemler arasında paylaşılmış ve ortak olan bir anlama havuzu oluşturacaktır. Bu sayede ontolojiler, insanlar arası ortak bir dil yaratacağı gibi, belirli bir uygulama alanının da modeli olarak kullanılabilirlerdir.

Ontolojilerin varlık sebebi aşağıdaki şekilde özetlenebilir [22]:

- İnsanlar ve yazılım ajanları ya da uygulamaları arasındaki ortak iletişimi sağlamak
- Belirli bir alandaki bilginin yeniden kullanılabilirliğini sağlamak
- Belirli bir alanda sağlıklı kabullenmeler ve çıkarsamalar yapabilmek

- Belirli bir alandaki bilgi ile operasyonel bilginin birbirinden ayrılmasını sağlamak
- Belirli bir alandaki bilginin analizini sağlamak (örn, nesneye yönelik modelleme)
- Veri hakkındaki verilere sahip olmak (üst-veri yönetimi)

Anlamsal ilişkilerin toparlanması sürecinden önce bu ilişkilerin modellenmesi, gerçekleştirilebilir ya da geliştirilebilir bir model olarak hazırlanması gerekmektedir. Bu ilişkilerin sağlanması ve farklı noktalardaki (üst-veri noktaları) verilerin ortak bir iletişim dili ile modele dahil edilmesi ile sunumunun yapılması gerekmektedir. Bu ortak dil, ortak bir sözlük ile verileri modeleyebilmeli, belirli bir alanda varolan verilerin iletişimini sağlayabilmelidir. Bu iletişimi sağlayan ortak dile ya da modele *ontoloji* adı verilmektedir.

Ontoloji, sözlük anlamı itibari ile kavramsallaşmış bir nesnenin tanımlanması olarak ifade edilmektedir. Yapay zeka çalışmaları sonucunda ortaya atılmış, filozofi tabanlı “varoluş bilimi” olarak da bilinen bir ifade tarzıdır. Daha formal bir tanımlama yapmak gerekirse bir ajan için ya da ajanlar grubu için varolan kavramlar ya da ilişkilerdir ya da kısaca toparlanmış bilgi kümeleridir [28] [14] [3].

Yapay zeka ve Web araştırmacıları ontoloji kavramını kendi jargonlarına dahil ederek ontolojinin terimler arasındaki ilişkileri ifade eden bir belge olarak nitelendirmişlerdir. En tipik ontoloji örneğinin bir taksonomisi (taxonomy) ve çıkarsama kuralları vardır. Taksonomi, nesne sınıfları ve bu nesnelere arasındaki ilişkileri tanımlar. Örneğin; bir *adres*, *yer tipinde* tanımlanabilir ve *şehir kodları sadece yer tipindeki verilere* verilecek şekilde tanımlanır. Bu sayede pek çok ilişki tanımlanabilir ya da türetilebilir: “Eğer şehirler yer tipindeyse, şehir kodları şehir bilgilerinde yer almaktadır; şehirler genellikle Web sitelerine sahiptirler. O halde, herhangi bir veritabanında şehir koduna ilişkin bir Web sayfası ilişkisi yer almıyorsa bile şehir kodları ile Web sitelerine erişilebilir...” şeklinde bir çıkarsama yapmak mümkün olmaktadır. Diğer bir yandan, “Bir şehir kodu eyelet kodu ile ilişkilendirilmişse ve bir adres bu şehir kodunu içermekteyse, bilinen adres

eldeki eyelet koduna sahip olmalıdır...” şeklinde bir başka çıkarsama da yapılabilir [3].

Verinin yönetiminde ya da ontolojilerde bilgi, çıkarsama amacı ile tutulmaktadır. Çıkarsama nedeni ile çoğu zaman pek çok örneğin kayıtlanmasına gerek yoktur. Çoğu zaman, örneklerin dolaşılması amacı ile sayısal ifadelerin belirtilmesi yeterli olmaktadır. Bir ontolojide açıkça belirtilmiş bilgiyi kullanarak yeni bilgilere ulaşmaya *çıkarsama* denmektedir. Çıkarsama yaparken açıkça belirtilmiş bilgi ve bu bilgilerden nasıl çıkarsama yapılacağını gösteren kurallar kullanılır [25] [34].

### 3. ANLAMSAL YAZILIM SÜREÇLERİ

Yazılım geliştirme süreçleri IEEE'nin 12207.0-1996 standardında belirlenen ortak çerçevede kapsamında [31]:

- Birincil Süreçler,
- Destekçi Süreçler, ve
- Organizasyonel Süreçler

olarak üç ana başlık altında toplanmıştır. Süreçler farklı gruplar içerisinde tanımlanmış olmasına karşın birbiri ile ilişkileri mevcuttur. Yazılım süreçlerinin belirlenmesinde kullanılan pek çok model sürecin anlaşılmasının geliştirilmesi, iletişim ve yönetimin daha iyileştirilmesi ya da süreçlerin otomasyonu gibi amaçları hedeflemektedir [10]. Bu süreç modelleri, bir sürecin hangi adımları içerdiği ya da bu adımların nasıl gerçekleştirileceği hakkında bilgi vermektedirler. Ancak, bir sürecin iyileştirilmesi ya da yeniden tasarlanması için daha derin bilgiye yani 'nasıl' sorusunun cevabına gereksinim vardır [13]. 'Nasıl' sorusunun cevabı süreç modellemede anlamsal ilişkilerin kurulmasında yardımcı olacaktır. Bu sorunun cevabı ile [11]:

- Yazılım süreçleri arasındaki iletişimin kuvvetlendirilmesi, deneyimlerin gelecekteki projelerde yeniden kullanılabilirliği,
- Yazılım geliştirme süreçlerinin olgunlaşması,

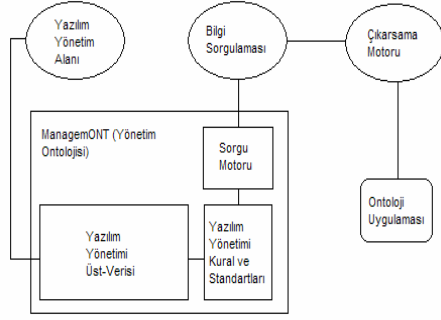
- Yazılım geliştirme standartlarının belirlenmesi ve bunların geliştirilmesi,

sağlanacaktır.

Geleneksel yazılım mühendisliği süreçleri risk, gereksinim, kalite, maliyet, doküman, sınama, doğrulama ve zaman yönetim aktivitelerine bağlı olarak yazılım yönetim sürecine kuvvetle bağlı yaşamaktadır. Bu yönetim aktivitelerinin sonuçları geleneksel olarak belge yığınları şeklinde saklanmakta ve her yeni ürün ya da proje başlangıcında bu belgeler ayrıştırılarak tecrübelerin başarılı bir şekilde yeniden kullanımı sağlanmaktadır. Sonuç olarak, ürün ya da projeler ayrıştırmanın zor ve zaman alıcı maliyeti, bunun yanında elde edilmiş olan tecrübelerin kaynaklar tarafından taşınmış olmasından dolayı sıfırdan tekrar tecrübe edilmektedirler. Bu da ürün ya da proje planlamasında zaman ve kaynakların verimsiz kullanılmasına neden olmaktadır. Yazılım mühendisliği süreçlerinin anlamsal modellenmesi yolu ile;

- Risk, maliyet ve gereksinim tecrübelerinin yeniden kullanılması,
- Başarılı kapatılan projelerdeki proje planlarının yeniden kullanılması,
- Kaynak kullanımına ilişkin istatistiksel verilere ulaşılması,
- Alt görevler için doğru ya da doğruya yakın zaman çizelgelerinin oluşturulması,
- Belgeleme, sınama ve doğrulama süreçlerinin beklenen olgunlukta devamı

sağlanabilecektir.

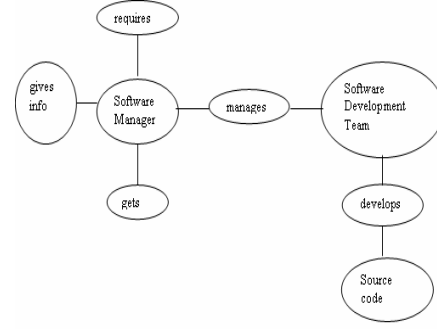


Şekil 6. Anlamsal yazılım yönetim süreci modeli

Şekil 6’ da belirtilen yazılım mühendisliği alanı diğer yazılım mühendisliği süreçlerini de içermektedir. Bu süreçler, üst-verileri yolu ile, yazılım yönetim süreci ile ilişki halindedirler. [18] [33] [31]. Üst-veriler, IEEE 12207.0-1996 standardında belirtilmiş olan standartlar ve diğer süreçlere olan dışsal ilişkiler ile meydana gelmektedir. Sorgu motoru, kural ve standart tabanlı üst-veri üzerinde çalışarak bilgiyi sorgulayan çıkarsama motoruna gerekli veri kümesini vermektedir. Çıkarsama motoru ontoloji uygulamasının gereksinimlerine bağlı olarak akıl yürütme faaliyeti sonucu gerekli yazılım yönetim bilgilerine ulaşmayı sağlamaktadır. Sorgu motoru, yazılım yönetim süreci kural ve standartları ile yazılım yönetim süreci üst-verisi birlikte yazılım yönetim ontolojisini oluşturmaktadır. Yazılım yönetim ontolojisi kısaca *ManagemONT* [33] olarak adlandırılmıştır.

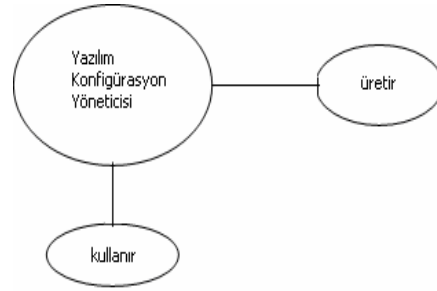
### 3.1. Yazılım Yönetim Süreci ve Anlamsal İlişkiler

Şekil 7’ de yazılım yönetim sürecinin yazılım geliştirme süreci ile kabaca ilişkisi belirtilmiştir. Yazılım yöneticisi, yazılım geliştirme ekibini yönetmekte ve yazılım geliştirme gibi de kaynak kodu üretmektedir. Diğer taraftan, yazılım yöneticisi ile yazılım konfigürasyon yöneticisi ile kaynak koda ilişkin paketin hazırlanmasına ilişkin “ister” adında bir anlamsal ilişkisi de mevcuttur. Yazılım yöneticisi, dokümantasyon grubuna “bilgi verir” ilişkisi ile ürün yönetimine de son-kullanıcı gereksinimlerini alabilmesi için “alır” ilişkisi ile bağlıdır.



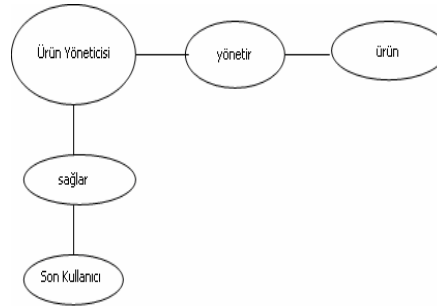
Şekil 7. Yazılım yönetim süreci

Şekil 8’ de yazılım konfigürasyon yönetiminin anlamsal ilişkileri ele alınmıştır. Kaynak kod paketlerinin oluşturulmasına ilişkin “üretir” ve yazılım geliştirme ekibinin ürettiği kaynak kodun paket hazırlanmasında kullanılmasına ilişkin “kullanır” isimli ilişkileri belirtilmiştir.



Şekil 8. Yazılım konfigürasyon yönetim süreci

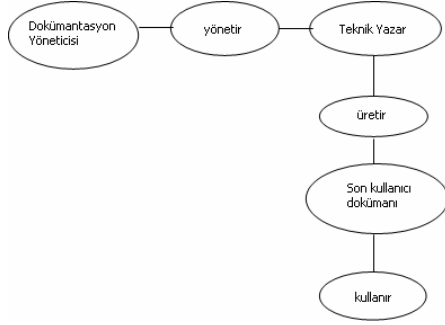
Ürün/Proje yönetim süreci Şekil 9’ da ele alınmıştır. Ürün yöneticisi ele alınırsa, ürünün yönetilmesine ilişkin “yönetir” ilişkisi ve son kullanıcıdan gereksinimlerin elde edilmesine ilişkin “sağlar” ilişkileri yer almaktadır.



Şekil 9. Ürün/Proje yönetim süreci

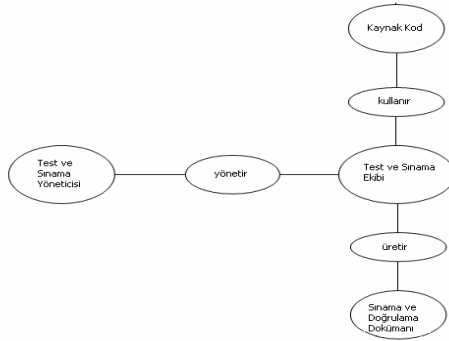
Dokümantasyon süreci ilişkileri Şekil 10’ da ele alınmıştır. Dokümantasyon yöneticisi “yönetir” ilişkisi ile teknik yazarı

yönetmektedir. Teknik yazar, “üretir” ilişkisi ile son kullanıcı dokümanını üretir ve son kullanıcı da “kullanır” ilişkisi ile son kullanıcı dokümanını kullanmaktadır.



Şekil 10. Yazılım dokümantasyon süreci

Şekil 11’ de sınamaya ve doğrulama sürecine ait ilişkiler ele alınmıştır. Süreç, sınamaya ve doğrulama yöneticisinin sınamaya ve doğrulama ekibini yönetmesine ilişkin “yönetir” ilişkisini, sürecin materyali olan kaynak kodun kullanılmasına ilişkin “kullanır” ilişkisi ve sınamaya ve doğrulama sonuçlarının hazırlanmasına ilişkin “üretir” ilişkisini içermektedir.



Şekil 11. Yazılım sınamaya ve doğrulama süreci

Yazılım mühendisliği süreçleri yukarıdakiler ile sınırlı değildir. Her süreç kendi içinde anlamsal ilişkilere sahiptir. Süreçlerin kendi içindeki ve diğer süreçler ile tanımlanmış anlamsal ilişkilerinin bütününe yazılım süreci ontolojisi adı verilmektedir [34] [33].

#### 4. GELECEK ÇALIŞMA

Bu bildiriye ilişkin araştırma IEEE 12207.0-1996 baz alınarak yazılım mühendisliği süreçlerinin belirlenmesi ile başlamıştır. Bu standartlar

yazılım yönetim sürecinin üst-verilerinin tanımlanmasına yardımcı olacaktır. Yukarıda bahsedilen ilişkiler daha detaylı olarak ele alınacaktır. Bu ilişkiler sayesinde yazılım yönetim ontolojisi elde edilmesi hedeflenmektedir. HP JENA<sup>5</sup> gibi sorgu işleyebilen arayüzler yolu ile ontoloji üzerindeki gerekli sorgulamaların yapılması ve RDF temelli bu model üzerinde çıkarsama yapılabilmesi amacı ile çıkarsama motoru gerçekleştirimi amaçlanmıştır. Bahsedilen modelin gerçekleştirimi sonrasında bir ontoloji uygulaması olarak düşünülen Anlamsal Yazılım Yönetim Birimi ile birlikte anlamsal yazılım yönetim süreç modelinin realize edilmesi planlanmaktadır. Hedeflenen yazılım yönetim aracı, günümüzde sadece üzerine yüklenen doğru veri ile ancak doğru sonuçlar üretebilen söz dizimsel yazılım yönetim araçlarına alternatif olarak düşünülmüştür.

#### 5. SONUÇ

Bilgi yönetiminin çok önemli olduğu günümüzde, bilgi dağıtık ve karmaşık yapıdadır. Varolan teknolojilerin bilgiler arasındaki ilişkileri ortaya koyması, doğru ve güncel verilere ulaşması güçtür. Ulaşılan bilgi ile ilişkisi bulunan diğer bilgilerin bilinmediği durumda elde edilen bilgi de yetersiz hale gelmektedir.

Yazılım kuruluşlarında tamamlanan projeler hakkında bilgiler, kuruluşların tanımladığı havuzlarda toplanmaktadır. Ancak gelecekte bu bilgilerin kullanılması güç olmakta ve kullanılmak istense de adaptasyon ve söz dizimsel zorluklardan dolayı vazgeçilmektedir. Bunun sonucunda yazılım yönetim süreci tarafından başarılı projelerde ya da ürün geliştirme süreçlerinde kullanılan plan, zamanlama ya da görevlendirme çizelgeleri yeni projelere uyarlanamamaktadır.

Bu bildiride, yazılım yönetim sürecinin sahip olduğu verilerin tanımlanması ve bu verilerin diğer yazılım mühendisliği süreçleri ile olan anlamsal ilişkilerinin belirlenmesinin önemi tartışılmıştır.

<sup>5</sup> <http://www.hpl.hp.com/semweb/tools.htm>

Bu sayede elde edilen üst-veriler yazılım yönetim ontolojisinin verileri olarak kullanılacaktır. Elde edilen yazılım yönetim modeli ile birlikte yazılım yönetim süreci üzerinden gerekli verilerin sorgulanması ya da doğru bilgilerin çıkarsanması sağlanabilecektir. Bu çıkarsama yolu ile başlanacak yazılım projeleri daha önceden başarılı olmuş projelerin ontolojiler üzerinde kayıtlı olan verileri sorgulanarak oluşturulacaktır. Ontolojiler sayesinde oluşturulmuş olan anlamsal süreç ilişkileri ile birlikte gelecekte yapılacak olan yazılım projeleri geçmişte başarılı olmuş projelerin verileri kullanılarak sağlanabilecektir.

## 6. Kaynakça

- [1].Allen, J.: Making a Semantic Web, 11/2001, Microsoft Corp.
- [2].Berners-Lee, T., Connolly D. ve Swick R. R.: Web Architecture: Describing and Exchanging Data, 06/1999, <http://www.w3.org/1999/04/WebData>
- [3].Berners-Lee, T., Hendler J. ve Lassila O.: The Semantic Web, 17/2001, [http://www.sciam.com/print\\_version.cfm?articleID=00048144-10D2-1C7084A9809EC588EF21](http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C7084A9809EC588EF21)
- [4].Berners-Lee, T.: Semantic Web on XML, 12/2000, <http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html>
- [5].Berners-Lee, T.: Semantic Web Roadmap, 10/1998, <http://www.w3.org/DesignIssues/Semantic.html>
- [6].Berners-Lee, T.: Uniform Resource Locators (URL), 12/1994, <http://www.w3.org/Addressing/rfc1738.txt>
- [7].Berners-Lee, T.: Uniform Resource Identifiers (URI): Generic Syntax, 08/1998, <http://www.ietf.org/rfc/rfc2396.txt>
- [8].Berners-Lee, T.: What The Semantic Web can Represent, 1998/09, <http://www.w3.org/1999/xhtml>
- [9].Ceri, S., ve Pelagatti, G.: Distributed Databases, 1984, McGraw Hill, ISBN: 0-07-010829-3
- [10].Curtis, W., Kellener, W. I. ve Over, J.: Process Modelling, *Comm. ACM*, 35 (9), 1992
- [11].Doheny, J. G., ve Filby, I. M.: A Framework and Tool for Modelling and Assessing Software Development Processes, AIAI-TR-204, Aug 1996, Wilmslow
- [12].Dumbill, E.: Building the Semantic Web, 07/2001, <http://www.xml.com/pub/a/2001/03/07/buildingsw.html>
- [13].Eric S. K. Yu ve Mylopoulos J.: Understanding why in Software Process Modelling, Analysis and Design, 16. International Software Engineering Conference, May 1994, Italy
- [14].Gruber, T.: What is an Ontology, 03/1993, <http://ksl-web.stanford.edu/people/gruber/>
- [15].Guha, R. ve McCool R.: TAP: A Semantic Web Platform, 2003, <http://tap.stanford.edu/tap.pdf>
- [16].Hawke, S.: How the Semantic Web Works, 04/2002, <http://www.w3.org/2002/03/semweb/>
- [17].Hendler, J., Berners-Lee, T., ve Miller, E.: Integrating Applications on the Semantic Web, 10/2002, <http://www.w3.org/2002/07/swint>
- [18].Humphrey, W. S.: Managing the Software Process, SEI Series, US, ISBN: 0-201-18095-2
- [19].Koivunen, M. R. ve Miller E.: W3C Semantic Web Activity, 11/2001, <http://www.w3.org/2001/12/semweb-fin/w3csw>
- [20].McBride, B.: The Semantic Web, 12/2002, Euroweb Conf.
- [21].Miller, E., Swick, R. ve Brickley, D.: Resource Description Framework (RDF) v1.173, 03/2006, <http://www.w3.org/RDF/>
- [22].Noy F. N. ve McGuinness D. L.: Ontology Development 101: A Guide to Creating Your First Ontology, 06/2004, Stanford Uni.
- [23].Özsu, M. T. ve Valduriez, P.: Distributed Database Systems, 1991, Prentice Hall, ISBN: 0-13-607938-5
- [24].Öztürk, Ö., Özacar, T. ve Unalır, M. O.: Ontoloji Tabanlı Türk Şarap Portalı Tasarımı, 06/2004, Ege Uni. Comp. Eng. Dept., Izmir
- [25].Palmer, S. B.: The Semantic Web: An Introduction, 09/2001, <http://www.w3.org/>
- [26].Palmer, S. B.: The Semantic Web, Taking Form, 06/2001, Semantic Web Agreement Group
- [27].Singer, M.: Semantic Web: Out of the Theory Realm, 12/2003, <http://siliconvalley.internet.com/news/article.php/3076961>

- [28].Swartz, A.: The Semantic Web in Breadth, 05/2002, <http://logicerror.com/semanticWeb-long>
- [29].Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification, 02/1999, <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- [30].Tannebaum, A.: Metadata Solutions, 2002, Addison Wesley, US, ISBN: 0-201-71976-2
- [31].Thayer, R. H. ve Sommerville, I.: Software Engineering Volume 2: Supporting Processes, 2002, IEEE Series, ISBN: 0-7695-1557-6
- [32].Tozer, G.: Metadata Management, 1999, Artech House, US, ISBN: 0-89006-280-3
- [33].Ulu, B.: ManagemONT, 2004, , Semantic Web, Ege Uni. Comp. Eng. Dept., Izmir
- [34].Ulu, B. ve Topaloglu, N. Y.: Anlamsal Yazılım Yönetimi Modelleme, 2005, Ege Uni. Comp. Eng. Dept., Izmir
- [35].W3C: Resource Description Framework (RDF) Schema Specification, 03/1999, <http://www.w3.org/TR/1999/PR-rdf-schema-19990303/>