Yazılım Yönetim Ontolojisi

Barış Ulu[1] Banu Diri[2]

[1,2]Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul
[1]e-posta: baris47@gmail.com [2]e-posta: banu@ce.yildiz.edu.tr

Özetçe

Günümüz yazılım mühendisliği bilgi tedariğinin en önemli sorunu verilerin çok farklı veri depolarında söz-dizimsel olarak tutulmasıdır. Artık haline gelmiş olan bu veriler, yazılım mühendisliği uygulamalarının tekrarlı verileri kullanmasına ve her yazılım geliştirme etkinliğinin sil baştan yapılmasına neden olmaktadır. Kazanılan tecrübelerde yeniden kullanılabilirliğin ya da geri dönüşlerin olmaması yazılım mühendisliği uygulamalarının, yazılım süreç modellerinde olduğu gibi, yazılım geliştirme sürecinde de verimsiz zaman ve kaynak kullanımını beraberinde getirmektedir. Yazılım geliştirme süreçlerinde tecrübelerin ve aynı zamanda bilginin yeniden kullanımın sağlanması için yazılım mühendisliği süreç modellerinde ontoloji yaklaşımının kullanılması yazılım mühendisliği yönetim sürecine fazlaca bağlı olan yazılım mühendisliği projelerinde bilgi ve tecrübeye dayalı başarıyı getirecektir. Bu çalışmanın amacı, yazılım mühendisliği standartlarının söz varlıklarının incelenmesi ve yazılım mühendisliği taksonomisinin geliştirilmesi, bu taksonomi ile birlikte elde edilen ilişkiler ve üst-veriye dayanarak kavramsal ontoloji modelinin çıkarılmasıdır.

1. Giriş

Günümüzde Web, kullanıcılarına yapısal olmayan, dinamik, dağıtık ve hızla büyüyen veri yığını sunmaktadır. Bu yığın, verinin ne anlama geldiğini ifade etmeye çalışan mantıksal işlenebilirlikten uzak, standartlaştırılmış bir artık (garbage) halini almıştır. Buradaki temel sorun, farklı merkezlerde bulunan verilerin birbirleri arasında bilmediğimiz mantıksal ilişkilerinin kullanıcılarına sunulmuyor olmasıdır. Verinin sunumu önemlidir ve bu arzu edilen sunum, veriler arası anlamsal ilişkilerin yaratılması ve yapılandırılması ile mümkün olmaktadır.

Anlamsal Web, ilk olarak 1994 yılındaki Uluslararası Web Konferansı1 'nda Tim Berners-Lee2 tarafından ortaya atılmıştır. Temel amaç, Web'deki boş duran bağlantıların sadece istekleri bir noktan diğer bir noktaya yönlendirmesinin yanı sıra bilgi erişimi için de kullanılabileceğinin düşünülmesidir. Sözü geçen veri yığını aslında Web'in dışındaki verilerdir; uygulamalar tarafından yönetilirler, ancak bir veri ağının parçası değildirler. Bu veri boşluğu uygulamaların sadece, Web'de ya da değil, bağlanılabilir veri depolarında daha önceden tanımlanmış söz-dizimsel veriler ile beslenmelerine neden olmaktadır.

Yazılım mühendisliği süreçleri, tecrübe yani bilginin yeniden kullanımı mümkün olmadığından dolayı her ürün/proje başlangıç safhasında sil baştan kullanılan veriler ile modellenmektedir. Bu geri dönüşümün ya da yeniden kullanımın sağlanabilmesi için yazılım mühendisliği süreçleri modellenirken her sürecin diğer süreç ile arasındaki, her süreç içindeki faaliyetlerin de diğer faaliyetler ile anlamsal ilişkilerinin belirlenmesi gerekmektedir.

Bu bildiride, giriş, sonuç ve kaynakça dışında iki ana bölüm ile birlikte çalışmanın yol haritasını belirleyen üç ana başlık bulunmaktadır. İkinci bölümde, Anlamsal Web hakkında kısa bir bilgi verilmiştir. Üçüncü bölümde ise varolan süreçlerin zayıf yönleri ile birlikte varolan yazılım mühendisliği süreç modellerine alternatif olabilecek bir model öne sürülmüştür.

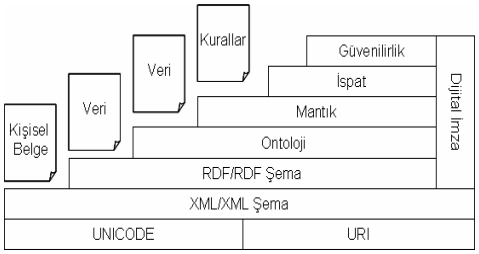
2. Anlamsal Web

Anlamsal Web [1,2,3,4,5,9,13,22,24,29,30,31,32,33,53], bir kısım kullanıcıların varolan iletişim ağı içerisinde bildiklerini eklemelerine izin veren, bir kısım kullanıcıların da sorularına cevap vererek gelişen bir bilgi kümesidir. Anlamsal Web'de bilgi, doğal dil metininden farklı olarak, yapısal devam ettirilen ve bu sayede insan ve makineler tarafından kolaylıkla kullanılabilmektedir [16]. Bu bilgi kolleksiyonu, sadece Web'den tanıdığımız ortam nesneleri (Web sayfaları, görüntü, klipler, vs.) için değil bunun yanında insan, yer, organizasyon ve olaylar için de kaynak teşkil etmektedir [15].

Anlamsal Web RDF [6,19,22,23,24,25,34] üzerine inşaa edilmiştir. RDF (Resource Description Framework), verilerin sunumunu sağlamak için üst-verileri [7,17,26,28,35,36,37,38] ifade etmeye yarayan URI (Uniform Resource Identifier) [2,23,24,34]' ları kullanmaktadır. RDF tarafından biçimlendirilen ve ilişkilendirilen bu veri kümeleri ontolojiler [20,39,40,41,42,43,44,45,46] yolu ile saklanmaktadır. Ontolojiler, dağıtık verilerin üst-verilerine dayalı olarak modellenmiş ortak iletişim araçlarıdır. Terminolojideki anlamının yanı sıra ontolojiler, bir takım aracı ya da ajanlar içinde de veri grupları olarak ele alınmaktadır [5,14,24]. Bilgi, ontolojilerde çıkarsama [1,22,24,42,43,47,48,49,50,51,52,53,54] amacı ile tutulmaktadır. Çıkarsama, farklı veri kümelerinden elde edilen veriler ile bilginin sağlanması yöntemidir. Çıkarsama sayesinde pek çok artık verinin saklanması ihtiyacı ortadan kalkmaktadır.

Şekil 1, Anlamsal Web' in çok iyi bilinen mimarisini göstermektedir. Bu mimari, Anlamsal Web araştırmacıları tarafından bir yol haritası olarak kabul edilmiştir. Anlamsal uygulamalar, uygulama yazarları tarafından bu mimari temel alınarak gerçekleştirilmektedir.

[1] http://www.iw3c2.org/
[2] http://www.w3.org/People/Berners-Lee/



Şekil 1: Anlamsal Web mimarisi [4,11]

Anlamsal Web sadece yeni bir veri modeli olarak tasarlanmamıştır; aynı zamanda dağıtık olarak varolan pek çok üst-veri modelinin de bağlantısını sağlamaktadır [21]. Anlamsal Web'in en önemli yanı, farklı veri depolarına bağlı olarak pek çok verinin saklanabilmesi ve bu veriler üzerinde karmaşık işlemlerin yapılıyor olmasıdır [1,3].

3. Anlamsal Yazılım Süreçleri

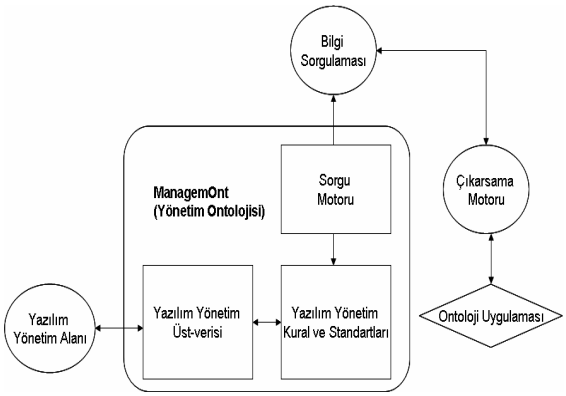
Yazılım mühendisliği süreçleri [18,27,55,56,57,58] farklı gruplarda ifade edilmektedir; bu süreçler girdiler ve çıktılar bazında birbirine sıkı sıkıya bağlıdır. Yazılım mühendisliği süreçlerinin ifade edilmesinde kullanılan pek çok model, süreç bilgi mal varlıkları [59] ile süreçlerin otomatikleştirilmesi [8] yerine iyileştirilmesine çaba harcamaktadır. Sözü geçen süreç bilgi mal varlıkları basitçe Şekil 2' de gösterilmiştir.

Şekil 2: Yazılım mühendisliği bilgi mal varlıkları

Sözü geçen süreç modelleri üst-seviye faaliyetlere odaklanmış modellerdir. Şekil 2' deki mal varlıkları arasındaki ilişkilerin tanımlanmamış olmasından dolayı pratik olamayacak kadar fazla soyut yapılardır [10]. Bu süreç modelleri, durağan tarzda bir sürecin hangi adımlarının ya da safhalarının olması gerektiğini ifade etmeye çalışmaktadır. Bir sürecin modellenmesi ya da yeniden tasarlanması [60] için *nasıl?* sorusuna cevap aranmalıdır [12]. *Nasıl?* sorusuna verilecek cevaplar, süreçler arası anlamsal ilişkileri tanımlayarak süreçler arası dinamik uyarlamaları [61,62] sağlayacaktır. Bu cevaplar, süreçler arası iletişimi kuvvetlendirecek, aynı zamanda da yazılım mühendisliği tecrübe [59] yani bilgilerinin yeniden kullanımını, süreçlerin olgunlaşabilmesini ve süreç modelleme [10] için standartların oluşmasını sağlayacaktır.

Geleneksel yazılım mühendisliği süreçleri, yazılım mühendisliği yönetim sürecine, risk, gereksinimler, kalite, maliyet, belgeleme, sınama ve doğrulama, ve zaman yönetimi faaliyetlerine sıkıca bağlıdır. Bu yönetim faaliyetlerinin sonuçları büyük kişisel belgeler halinde saklanmakta ve her yeni ürün ya da proje başlangıç safhasında bu belgeler, başarıların yeniden kullanılabilirliğini sağlamak amacı ile tekrar tekrar ayrıştırılmaya çalışılmaktadır. Bu ayrıştırma son derece zahmetli ve zor olduğundan dolayı her yeni ürün ya da proje sil baştan süreç planlamasına girmektedir. Bu da verimsiz zaman ve kaynak kullanımına neden olmaktadır.

Yazılım mühendisliği yönetim sürecinin anlamsal olarak modellenmesi risk tecrübelerinin, maliyet ve gereksinim analizlerinin başarılı olarak sonlandırılmış projelerden yeniden kullanımına olanak sağlamaktadır. Başarılı olmuş projelerde kullanılan proje planları, istatistiksel veriler her yeni projede zaman kaybı olmaksızın proje planlarının üretilmesine, buna bağlı olarak da proje süresince kaynak ve zamanın verimli kullanımına yardımcı olmaktadır. Anlamsal ilişkilerin belirlnmesi yolu ile doğru ya da doğruya yakın zaman çizelgeleri, görev atama ve zamanlamaları ile kalite güvence yöntemlerinin projelere kolay adaptasyonu sağlanabilmektedir. Bu adaptasyon, belgeleme, sınama ve doğrulama süreçlerinin daha önceki başarılı projelerde varolan çıktılarının yeniden kullanılabilmesi ile mümkün olmaktadır.



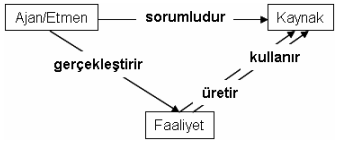
Şekil 3: Yazılım mühendisliği yönetim süreci anlamsal modeli

Şekil 3' de belirtilen yazılım mühendisliği alanı yukarıda bahsedilen diğer yazılım mühendisliği süreçlerini belirtmektedir. Bü süreçler, üst-verisi yolu ile yazılım mühendisliği yönetim süreci [18,27,55,56,57,58] ile ilişkilidirler. Bu üst-veri, IEEE [63,64] tarafından belirlenmiş kural ve standartlar ile biçimlendirilmiştir. Sorgu motoru ile kural ve standart tabanlı üst-veri üzerinde gerekli sorgulama yapılabilmekte ve sonuçlar çıkarsama motoruna iletilmektedir. Çıkarsama motoru sayesinde sorgulama sonucu elde edilmiş olan verilerden, akıl yürütme ile ontoloji uygulamasının ihtiyacı olan yazılım mühendisliği yönetim bilgilerine ulaşılmaktadır. Yazılım mühendisliği kural ve standartları, sorgu motoru ve yazılım mühendisliği yönetim süreci üst-verisi, yazılım mühendisliği yönetim süreci ontolojisini oluşturmaktadır. Yazılım mühendisliği süreç ontolojisi kısaca ManagemONT olarak isimlendirilmiştir.

3.1. Yazılım Mühendisliği Süreci Üst-veri Modeli

Her yazılım mühendisliği süreci dinamik [61,62] ve otomatikleştirilmiş [65] bir bileşen olarak düşünülmelidir. Bu bileşenler, diğer bileşenlerden gelen çıktıları girdi olarak kabul ederler, içsel faaliyetlerini gerçekleştirip başka bir

bileşene girdi olması için gerekli çıktıları üretirler. Bu bileşenler arasındaki girdi/çıktı alışverişleri diğer taraftan da ilişkileri belirler. Şekil 4' de belirtildiği gibi bileşen, süreç, faaliyet adı verilen içsel görevlerini gerçeklemektedir. Sürecin sahibi, ajan ya da etmen, bu faaliyetleri gerçeklemekle yükümlüdür. Her etmen ya da ajana ait bir rol belirlenmiştir. Şekil 4' de sürecin rolü ilgili kaynaktan sorumlu olmaktır. Bu kaynaklar faaliyetler tarafından üretilir ya da kullanılırlar.



Şekil 4: Somut süreç modeli

SPEM (Software Process Engineering Metamodel) [66] yazılım mühendisliği faaliyet ve bağımlılıkları için temel bir üst-model sunmaktadır. Bu soysal (generic) model, proje planını kaynak olarak kullanır. Yazılım mühendisliği yöneticisini ajan ya da etmen olarak tanımlar. Faaliyet olarak ilgili planı üretir. Bu sayede yazılım mühendisliği süreçlerine kolaylıkla uygulanabilmektedir.

Yazılım mühendisliği organizasyonel süreçlerinden [18,27,63,64] yönetim süreci aşağıdaki şekilde listelenebilir:

- Başlatma ve kapsam tanımı
- Planlama
- Yürütme ve kontrol
- Gözden geçirme ve değerlendirme
- Kapatma

Yukarıda listelenen tüm faaliyetlerin farklı kaynaklar ile birlikte bir ajan ya da etmeni mevcuttur. Bu ajan ya da etmen yazılım mühendisliği yöneticisi ya da ilgili faaliyeti yürüten nesnedir. Yazılım mühendisliği yönetim süreci taksonomisi, yazılım mühendisliği yönetim süreci üst-verisinin tanımlanmasına da yardımcı olacaktır. Şekil 5' de yazılım mühendisliği yönetim süreci taksonomisi verilmiştir.

Şekil 5: Yazılım mühendisliği yönetim süreci faaliyet taksonomisi

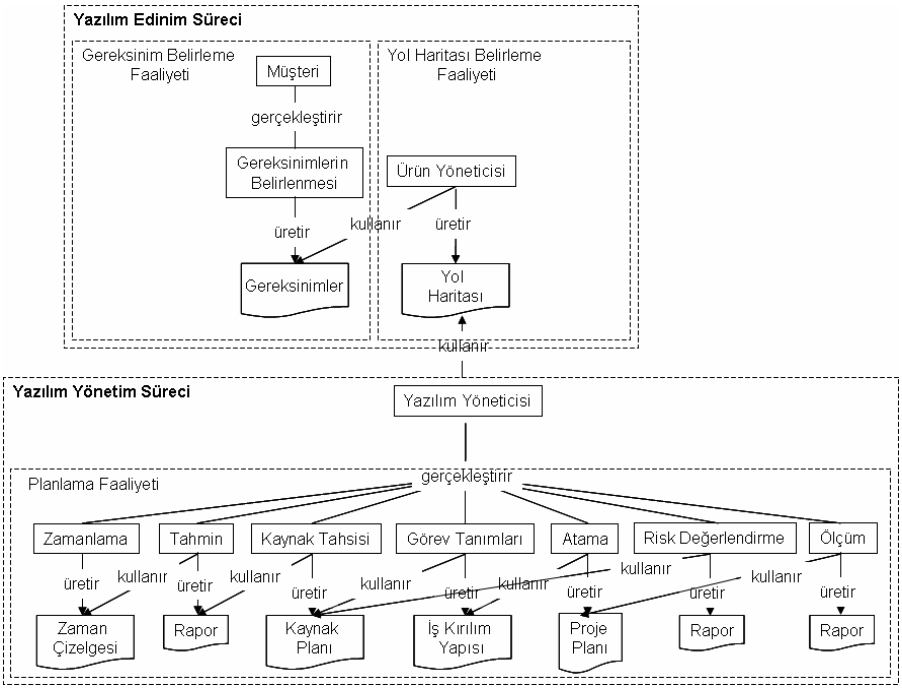
Şekil 5'te gösterilen taksonomi oldukça büyük olduğundan dolayı sözü geçen sürecin planlama faaliyeti ele alınmış ve Şekil 6' da yazılım mühendisliği yönetim sürecinin planlama faaliyetine ilişkin üst-veri *somut süreç modeli* baz alınarak belirtilmiştir.

Şekil 6, yazılım mühendisliği yönetim süreci planlama faaliyetine ilişkin üst-veriyi yazılım mühendisliği yönetim süreci taksonomine bağlı olarak sunmaktadır. Üst-veride belirtilen her faaliyet ortak bir ajan ya da etmene sahiptir. Her faaliyetin bir girdi ve bir çıktısı bulunmaktadır. Yazılım mühendisliği yöneticisi, yazılım edinim süreci [18,27,63,64] yolu ile belirlenen gereksinimlerin ürün yöneticisi tarafından yol haritası haline getirilmiş çıktısını kullanmaktadır. Yazılım mühendisliği yöneticisi, bu yol haritasını girdi olarak kullanmakta, bu girdi yolu ile planlama faaliyetini ve bu faaliyete bağlı olarak diğer alt faaliyetleri gerçeklemektedir. Proje zamanlama faaliyeti ile elde edilne zaman çizelgesi tahmin yönetimi için girdi olarak kullanılmaktadır. Tahmin yönetimi sonucu elde edilen çıktı kaynak kullanımı için bir girdi teşkil etmektedir. Kaynak kullanımı faaliyeti ile elde edilen çıktı ile ilgili etmen ya da ajan iş kırılım yapısı (WBS-Work Breakdown Structure) [67] uygulayarak görev tanımlarını ve iş atamalarını gerçekleştirmektedir. Görev tanımları ve iş atamaları sonucu elde edilen çıktı, yazılım mühendisliği yöneticisi tarafından proje planına girdi olarak kullanılmaktadır. Diğer taraftan, üretilen kaynak ve proje planları ilgili ajan ya da etmen tarafından risk değerlendirme ve ölçümler için birer girdi, bu girdi ile birlikte risk değerlendirme ve ölçümler sonucu elde edilen çıktılar, yani sayısal değerler, kalite güvence süreci için birer girdi olarak tanımlanmaktadır.

3.2. Yazılım Mühendisliği Yönetim Süreci Ontolojisi

Yazılım mühendisliği yönetim süreci için tanımlanan üst-veri modeline ait her ajan/etmen, girdi, çıktı ve faaliyet yazılım mühendisliği yönetim ontolojisi için bir kavramı ifade etmektedir. Her kavramın kendine ait ozellikleri ve diğer kavramlar ile tanımlanmış ilişkileri mevcuttur.

Yazılım mühendisliği yönetim süreci için tanımlanan üst-veri modeli ilgili sürecin diğer süreçler ile anlamsal ilişkilerini de belirtmektedir. Örnek model tüm yazılım mühendisliği süreçleri için genişletildiği takdirde her sürecin kendi içinde ve diğer süreçler ile arasında benzer ilişkilerin varlığı göze çarpmaktadır. Şekil 6' daki örnek modelde bu ilişkilerden bir tanesi *kullanır* ilişkisi ile gösterilmiştir.



Şekil 6: Yazılım mühendisliği yönetim süreci planlama faaliyeti üst-veri modeli

Yazılım mühendisliği yönetim süreci diğer süreçler ile olan anlamsal ilişkileri nedeni ile diğer yazılım mühendisliği süreçlerinin merkezinde yer almaktadır. Yazılım mühendisliği yönetim süreci, yazılım geliştirme sürecini yönetmekte, yazılım geliştirme süreci de kaynak kodu üretmektedir. Diğer bir taraftan yazılım yönetim süreci, yazılım konfigürasyon yönetimi ile *istemek* anlamsal ilişkisi yolu ile kaynak koda ilişkin yazılım yükünün hazırlanması, *bilgi vermek* anlamsal ilişkisi ile belgeleme sürecinde gerekli belgelemelerin yapılmasını sağlamaktadır.

Yazılım mühendisliği sınama ve doğrulama süreci *kullanmak* anlamsal ilişkisi ile yazılım konfigürasyon yönetimi sürecinin ürettiği yazılım yükünün kullanılması faaliyetini gerçekler. Belgeleme süreci, sınama ve doğrulama sonuç raporunu çıktı olarak yazılım mühendisliği belgeleme sürecine girdi olarak vermektedir.

Yazılım mühendisliği süreçleri yukarıda söz edilen süreçler ile sınırlı değildir. Sözü edilen süreç ilişkileri ontolojinin modellenmesinde temel olarak alınacak ilişkilerdir. Yazılım mühendisliği yönetim sürecinin kendi içinde dinamik olarak varolan faaliyetleri, girdi, çıktı ve rolleri ile belirlenecek olan yazılım mühendisliği yönetim süreci anlamsal ilişkileri, yazılım mühendisliği yönetim süreci ontolojisi olarak adlandırılacaktır.

4. Gelecek Çalışma

ManagemONT isimli çalışma IEEE 12207.0' da detaylı olarak belirlenmiş olan yazılım mühendisliği standartlarının incelenmesi ile başlamıştır. Bu standartlar halen yazılım mühendisliği yönetim sürecinin üst-verisinin modellenmesinde kullanılmaktadır. Elde edilecek olan üst-veri yolu ile yazılım mühendisliği yönetim süreci ve diğer süreçler arası anlamsal ilişkilerin belirlenmesi hedeflenmektedir. Bu anlamsal ilişkiler yolu ile elde edilecek olan kavramlar ile birlikte sürece ait kurallar, IEEE 12207.0 standartlarına uygun olarak tanımlanmıştır. İlgili sürecin tanımlanan kuralları yolu ile sürecin ajan/etmen, faaliyet ve kaynaklardan oluşan durağan kavramları Şekil-6' da belirtilen üst-veri modeli baz alınarak ontoloji tabanlı olarak modellenecektir. Yazılım mühendisliği yönetim sürecine ait girdi ve çıktıların modele dahil edilmesi yolu ile Şekil-4' te belirtilen somut süreç modeli kullanılarak yazılım mühendisliği yönetim sürecine ait dinamik ontoloji modeli elde edilecektir.

Verilerin sorgulanabilmesi için sorgu motoru geliştirilmesi ve elde edilen verilerden bilgiye erişecek olan çıkarsama motorunun yazılım mühendisliği uygulama alanındaki kurallar çerçevesinde kullanılması hedeflenmektedir.

Son olarak, elde edilen çıkarsanmış bilgilerin kullanılacağı Anlamsal Yazılım Yönetim Birimi isimli ontoloji uygulaması geliştirilecektir. Uygulamanın amacı, günümüz yazılım mühendisliği süreç ve yönetim araçlarından farklı olarak doğru veri ile söz-dizimsel sonuçlar üreten değil, veri ile anlamsal sonuçlar üretebilen bir alternatif sunmaktır.

5. Sonuçlar

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 eldeki bilgi de yararsız hale gelmektedir.

Yazılım kuruluşlarında tamamlanan projeler hakkındaki 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 ile yeni projelere uyarlanamamaktadır.

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

Elde edilen üst-veriler yazılım mühendisliği yönetim ontolojisinin verileri olarak kullanılacaktır. Varolan iş akış yönetimi ve belgeleme yönetimi modellerine benzer olarak yazılım mühendisliği yönetim süreci üst-verisi elde edilirken somut sureç modeli kullanılmıştır. Ancak sözü edilen modellerden farklı olarak somut süreç modelinden elde edilmiş olan üst-veri modeli baz alınarak gerçekleştirilecek olan yazılım mühendisliği yönetim süreci ontolojisi ile olmayan süreç verilerinden çıkarsama yolu ile yeni bilgiler elde edilebilecektir.

Çıkarsama yolu ile başlanacak yazılım mühendisliği 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 mühendisliği projeleri geçmişte başarılı olmuş projlerin verileri kullanılarak sağlanabilecek, yazılım mühendisliği yönetim bilgileri doğru veri – doğru bilgi yargısından uzaklaşarak elde veri olmasa da doğru bilgiye ulaşabilmeyi sağlayacaktır.

6. Kaynakça

[1] Allen, J., Making a Semantic Web, http://www.netcrucible.com/semantic.html, 2001

[2] Barners-Lee, T., Uniform Resource Identifiers (URI): Generic Syntax, W3C Publications, http://www.ietf.org/rfc/rfc2396.txt, 1998

[3] Berners-Lee, T., What The Semantic Web can Represent, W3C Publications on Semantic Web, http://www.w3.org/1999/xhtml, 1998

[4] Berners-Lee, T., Semantic Web on XML, W3C Publications on Semantic Web, http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html, 2000

[5] Berners-Lee, T. et. al., The Semantic Web, The Scientific American Publications on Semantic Web, http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70 84A9809EC588EF21, 2001

[6] Brickley D., ve Guha R. V., Resource Description Framework (RDF) Schema Specification, W3C Publications on Semantic Web, http://www.w3.org/TR/1999/PR-rdf-schema-19990303/, 1999

[7] Ceri, S., ve Pelagatti, G., Distributed Databases, McGraw Hill, US, 1984

[8] Curtis, W. et. al., Process Modelling, ACM Publications, 1992

[9] Daconta, M. C., Obrst, L. J. ve Smith, K. T., The Semantic Web, Wiley Inc., US, 2003

[10] Doheny, J. G., ve Filby, I. M., A Framework and Tool for Modelling and Assessing Software Development Processes, Artificial Intelligence Applications Institute (AIAI) Publications, 1996

[11] Dumbill, E., Building the Semantic Web, The XML.COM Publications on Semantic Web, http://www.xml.com/pub/a/2001/03/07/buildingsw.html, 2001

[12] Eric S. K. Yu ve Mylopoulos J., Understanding why in Software Process Modelling, Analysis and Design, 16th International Software Engineering Conference Publications, 1994

[13] Fensel, D., Hendler, J., Lieberman, H. ve Wahlster, W., Spinning the Semantic Web, MIT Press, US, 2003

[14] Gruber, T., What is an Ontology, , http://ksl-web.stanford.edu/people/gruber/, 1993

[15] Guha, R. ve McCool R., TAP: A Semantic Web Platform, http://tap.stanford.edu/tap.pdf, 2003

[16] Hawke, S., How the Semantic Web Works, W3C Publications on Semantic Web, http://www.w3.org/2002/03/semweb/, 2002

[17] Hay, D. C., Data Model Patterns, Dorset House Publishing, US, 1996

[18] Humphrey, W. S., Managing the Software Process, SEI Series, US, ISBN: 0-201-18095-2, 1998

[19] Miller et. al., Resource Description Framework (RDF) v1.173, W3C Publications on Semantic Web, http://www.w3.org/RDF/, 2006

[20] Noy F. N. ve McGuiness D. L., Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Uni. Publications, 2004

[21] Özsu, M. T. ve Valduriez, P., Distributed Database Systems, Prentice Hall, US, 1991

[22] Palmer, S. B., The Semantic Web: An Introduction, W3C Publications on Semantic Web, http://www.w3.org/, 2001

[23] Palmer, S. B., The Semantic Web: Taking Form, Semantic Web Agreement Group Publications, http://infomesh.net/2001/06/swform/, 2001

[24] Swartz, A., The Semantic Web in Breadth, http://logicerror.com/semanticWeb-long, 2002

[25] Swick, R., Resource Description Framework (RDF) Model and Syntax Specification, W3C Publications on Semantic Web, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, 1999

[26] Tannenbaum, A., Metadata Solutions, Addison Wesley, US, 2002

[27] Thayer, R. H. ve Sommerville, I., Software Engineering Volume 2: Supporting Processes, IEEE Series, US, 2002

[28] Tozer, G., Metadata Management, Artech House, US, 1999

[29] Berners-Lee, T. ve Miller, E., The Semantic Web lifts off, ERCIM News Issue: 51, ERCIM EIGG, France, 2002

[30] Antoniou, G., Harmelen, van F., A Semantic Web Primer, The MIT Press, UK, ISBN: 0-262-01210-3, 2004

[31] Passin, B. T., Explorer's Guide to the Semantic Web, Mining Co., US, ISBN: 1-932394-20-6, 2004

[32] Alesso, P. H. ve Smith, C. F., Thinking on the Web, Wiley, US, ISBN: 0-471-76814-6, 2006

[33] Javies, D., Fensel, D., Harmelen, van F., Towards the Semantic Web, Wiley, US, ISBN: 0-470-84867-7, 2003

[34] Powers, S., Practical RDF, O'Reilly, US, ISBN: 0-596-00263-7, 2003
[35] Sumpter, R. M., Data Management, Lawrence Livermore National Lab., 1994
[36] Thangarathinam, T., Wyant, G., Gibson, J. ve Simpson, J., Metadata Management: The Foundation for Enterprise Information Integration, Intel Technology Journal Volume: 8 Issue: 04, US, 2004
[37] Sun Microsystems, Metadata Management: An Essential Ingredient for Information Lifecycle Management, Sun Mic., 2005
[38] NISO Press, Understanding Metadata, NISO Press Booklets, US, ISBN: 1-880124-62-9, 2004
[39] Spyns, P., Meersman, R. ve Jarrar, M., Data Modelling versus Ontology Engineering, STARLab, Belgium, 2002
[40] Kalinichenko, L., Missikoff, M., Schiapelli, F. ve Skvortsov, N., Ontological Modelling, Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003, Russia, 2003
[41] Gardner, S. P., Ontologies ve Semantic Data Integration, DDT Volume: 10 Number: 14, 2005
[42] Cruz, I. F. ve Xiao, H., The Role of Ontologies in Data Integration, ADVIS Lab., US, 2005
[43] Mizoguchi, R. ve Ikeda, M., Towards Ontology Engineering, Technical Report AI-TR-96-1, I.S.I.R, Jp, 1998
[44] Mizoguchi, R., Tutorial on Ontological Engineering, I.S.I.R, Jp, 2004
[45] Ding, Y. ve Fensel, D., The Thematic Network for the Semantic Web, ERCIM News Issue: 51, ERCIM EIGG, France, 2002
[46] Valle, D. E. ve Brioschi, M., An Ontology-Oriented Solution for Knowledge-Intensive Organization, ERCIM News Issue: 51, ERCIM EIGG, France, 2002
[47] Ramakrishnan, R., Gehrke, J., Database Management Systems 2nd Edition, McGraw Hill, US, ISBN: 0-07-246535-2, 2000
[48] Djuric, D., Gasevic, D., Damjanovic, V. ve Devedzic, V., MDA-based Ontological Engineering, GOOD OLD AI research Group, Serbia and Montenegro, 2005
[49] Cranefield, S. ve Purvis, M., UML as an Ontology Modelling Language, IJCAI 99 Proceedings, Nz, 1999
[50] Patel-Schneider, P. F. ve Horrocks, I., Position Paper: Comparison of Two Modelling Paradigms in the Semantic Web, WWW2006, US, 2006
[51] Dieng-Kuntz, R., Corporate Semantic Webs, ERCIM News Issue: 51, ERCIM EIGG, France, 2002
[52] Pahl, C., Ontologies for Semantic Web Components, ERCIM News Issue: 51, ERCIM EIGG, France, 2002
[53] Staab, S., The Semantic Web Revisited, IEEE Computer Society 1541-1672/06, Germany, 2006
[54] Berners-Lee, T., The Semantic Web Roadmap, W3C, US, 1998
[55] Pressman, R. S., Software Engineering: A Practitioner's Approach 5th Edition, McGraw Hill, US, ISBN: 0-07-709677-0, 2000
[56] Pfleger, S. L., Software Engineering: Theory ve Practice 2nd Edition, Prentice Hall, US, ISBN: 0-13-029049-1, 2001
[57] Behrofoz, A. ve Hudson, F. J., Software Engineering Fundamentals, Oxford Press, UK, ISBN: 0-19-510539-7, 1996
[58] Abran, A., Moore, J. W., Bourque, P., Dupuis, R. ve Tripp, L. L., SWEBOK: Software Engineering Body of Knowledge Trial Version, IEEE Computer Society, US, ISBN: 0-7695-10000-0, 2001
[59] Mohame, A. H., Lee, S. P. ve Salim, S. S., An Ontology-based Knowledge Model for Software Experience Management, Journal of Knowledge Management Practice, 2004
[60] Scacchi, W., Understanding Software Process Redesign Using Modeling, Analysis and Simulation, Software Process Improvement and Practice, Wiley, US, 2000
[61] Gnatz M., Marschall, F., Popp, G., Rausch, A. ve Schwerin, W., Common Meta-Model for a Living Software Development Processes, Munich Technical University ZEN research project, Germany, 2002
[62] Ramil, J. F., Lehman, M. M., Modeling Process Dynamics in Software Evolution Processes, SCE 99 Proceeding, US 1999
[63] ISO/IEC 12207, Software Lifecycle Processes – Implementation Considerations, IEEE/EIA Guide 12207.2-1997, US, 1998
[64] ISO/IEC 12207, Software Lifecycle Processes, IEEE/EIA Guide 12207.0-1996, US, 1998
[65] Mi, P. ve Scacchi, W., A Knowledge-based Environment for Modeling and Simulating Software Engineering Processes, IEEE Transactions on Knowledge and Data Engineering Volume: 2 No:3, 1041-4347/90/0900-0283$01.00, US, 1990
[66] OMG, The Software Process Engineering Metamodel (SPEM), OMG Documents: ad/2001-03-08, US, 2001
[67] Gustafson, D., Software Engineering, McGRaw Hill, US, ISBN: 0-07-137794-8, 2002