

Bağlı Veri Teknolojileri Kullanılarak Üniversite Verisinin Bütünleştirilmesi ve Yayınlanması

Tayfun Gökmen Halaç, Emrah İnan, Damla Oğuz, Bahtiyar Erden, Pınar Göçebe, Oğuz Dikenelli

Bilgisayar Mühendisliği Bölümü, Ege Üniversitesi,
35100 Bornova, İzmir, Türkiye
{tayfunhalac, bahtiyar.erden, gocebepinar}@gmail.com
{emrah.inan, oguz.dikenelli}@ege.edu.tr
Bilgisayar Mühendisliği, İzmir Yüksek Teknoloji Enstitüsü
35430 Urla, İzmir, Türkiye
damlaoguz@iyte.edu.tr

Özet Üniversiteler, öğrenci, öğretim üyesi, ders, proje, fakülte konumları, kütüphane gibi birçok kavram hakkında büyük miktarda veriyi sahip kurumlardır. Üniversitelerde öğrenci bilgi sistemi, kütüphane arşiv yazılımı, araştırma projeleri yönetim sistemi, ders yönetim sistemi, personel yönetim sistemi, coğrafi bilgi sistemi gibi her biri farklı bir gereksinimi karşılayan birçok yazılım bir arada kullanılmaktadır. Genellikle farklı kuruluşlar/kişiler tarafından geliştirilmiş olan bu yazılımlar birçok veriyi paylaşma gereksinimi duymaktadırlar. Yazılımlar arasındaki bu veri bütünleştirme gereksinimini karşılamak amacıyla kullanılan veb servisleri gibi teknolojiler büyük miktarda eşgüdüm ve bakım maliyeti getirirler. Son yıllarda gelişen bağlı veri (diğer adıyla anlamsal veb) teknolojisi, bu sorunları ortadan kaldırmayı hedeflemiş ve üniversiteler dahil olmak üzere birçok kuruluştaki kullanılmaya başlanmıştır. Bu doğrultuda, Ege Üniversitesi, farklı yazılımların veritabanlarında kapalı kalan verileri bağlı veri teknolojilerine dayanarak bütünleştirmiş, yayınlamış ve tüm yazılımların kullanımına açık hale getirmiştir. Bu çalışmada, bağlı veri temelli veri bütünleştirme mimarileri değerlendirilmiş ve Ege Üniversitesi'ndeki yazılım mimarisinin bağlı veri teknolojileri kullanılarak yeniden şekillendirilmesinin yol haritası tanıtılmıştır. Bu yol haritası, bağlı veri teknolojilerinin üniversite alanında veri bütünleştirme için kullanımına bir örnek oluşturmaktadır.

Anahtar Sözcükler. bağlı veri, anlamsal veb, veri tümleşimi, üniversite verisi, veri yayınlama mimarisi

1 Giriş

Yaklaşık 40 bin öğrenciye sahip Ege Üniversitesi'nde günlük iş süreçleri sürekli olarak veri üretir. Öğrenci, öğretim üyesi, ders, proje, fakülte konumları, kütüphane gibi birçok kavram hakkındaki bu veriler, üniversitenin eğitimsel ve yönetsel ihtiyaçlarını karşılayan farklı yazılım sistemleri tarafından üretilir ve yönetilirler. Bu yazılımlara örnek olarak ders yönetim sistemi, öğrenci bilgi sistemi,

kütüphane arşiv sistemi, araştırma projeleri yönetim sistemi, personel yönetim sistemi ve coğrafi bilgi sistemi verilebilir. Bu yazılımlar arasında üniversitenin sahip olduğu birçok verinin paylaşılması gereksinimi ortaya çıkmaktadır. Örneğin, akademik personel ve kuruluş yapısı hem öğrenci bilgi sisteminde, hem ders yönetim sisteminde hem de araştırma projeleri yönetim sisteminde bulunan verilerdir. Bir başka örnek olarak ders planlarında verilen kaynak kitapların üniversite kütüphanesine bağlanabilirliği verilebilir. Bu gibi durumlar farklı yazılımların veritabanlarında verinin tekrar edilmesine (buna ilişkin olarak tekrar verilerde tutarsızlık oluşmasına) ve kullanıcıların farklı yazılımlara dağılmış ilişkili verilere ulaşamamasına neden olmaktadır.

Birçok büyük kuruluşta olduğu gibi üniversitelerde de gereksinim duyulan yazılımlar farklı özel şirketler veya çeşitli üniversite personeli tarafından geliştirilmektedir. Birçok farklı katılımcının bulunduğu böyle bir yazılım ekosisteminde veri tekrarının önüne geçmek ve yazılımlar arası bütünlüğü sağlamak için veb servisleri gibi teknolojiler kullanılabilir. Bu durumda bir yazılım, diğer yazılımların verisine veb servisleri üzerinden erişebilir. Ancak, bu servisler geliştirici bağımlı olduğundan veriyi kullanan yazılımcılar veri modelini anlamakta ve isteklerine uyarlamakta güçlük çekebilirler. Bunu ortadan kaldırmak için veriyi paylaşan taraflar arasında bir eşgüdüm sağlanması gerekmektedir. Bu durum da yazılım sahibine veriyi paylaşacak diğer tüm yazılımcılar ile birebir eşgüdüm maliyeti getirmektedir. Bu yöntemeye dayanan bütünleştirmelerde bir diğer önemli nokta sağlanan servislerin yazılım güncellemeleri ve yeni gereksinimler karşısında güncelliğini korumasıdır. Bu gereksinim ise yazılım sahibine bir bakım maliyeti doğurmaktadır. Söz edilen eşgüdüm ve bakım maliyetleri yazılımlar arasında bütünleştirmelerin etkin bir şekilde yapılmasının önüne geçmektedir.

Ege Üniversitesi'nde kullanılan yazılımlar arasında veri bütünleştirmenin yanı sıra bu verileri kullanacak yeni yazılımlara gereksinim duyulmaktadır. Üniversite kuruluşu, bürokratik süreçlerin de etkisiyle yeni yazılım gereksinimlerini karşılamakta güçlük çekmektedir. Ayrıca, üniversite içi veri paylaşımının yanı sıra bir üst seviyede üniversiteler arasında veya üniversite ile kamu/özel kuruluşlar arasında da benzer gereksinimler bulunmaktadır. Örneğin, Ege Üniversitesi ile İzmir Büyükşehir Belediyesi arasında her dönem başında öğrencilere özel olarak sağlanan toplu ulaşım kartlarının geçerliliğinin kontrolü, bürokratik ve insana dayalı süreçler dolayısıyla hataya açık ve yavaş bir şekilde gerçekleşmektedir. Bir başka örnek olarak öğrenci veya öğretim üyelerinin çeşitli sosyal ağlarda bulunan verilerinin de kullanılacağı yenilikçi uygulama fikirleri verilebilir. Çoğaltılabilecek olan bu örnekler, hem üniversite içinde hem de üniversite dışı kurumlar ile veri bütünleştirme üzerine bir çalışma yapılması gereksinimini açıkça ortaya koymaktadır.

Son yıllarda gelişmekte olan bağlı veri¹ teknolojileri [2] kurum-içi ve kurumlar arası veri tümleşimi için veb standartlarına dayanan bir yaklaşım or-

¹ Bağlı veri, yıllar önce ortaya konan anlamsal veb kavramının yalnızca veb üzerinde etkin veri paylaşımına odaklanan kısmına verilen addır. Bu kavram, anlamsal veb hedefine ilerleyebilmek için veb üzerinde yazılımlar tarafından tüketilebilen bir veri ağı oluşturulması amacıyla ortaya atılmıştır.

taya koymaktadır [6,9,8]. Üniversitelerde veri bütünleştirme amacıyla da kullanılmakta olan [7,5,11] bu teknoloji, diğer yaklaşımların eksik yanlarının aksine, açık standartlara dayanarak ve açık veri şemalarını kullanarak farklı kaynaklardaki verilerin birbirine bağlanmasına olanak tanımaktadır. Bağlı veri teknolojileri, verinin standart olarak RDF² biçiminde HTTP³ protokolü ile paylaşılmasını ve SPARQL⁴ sorgu dili ile yine HTTP protokolü ile sorgulanabilmesini öngörmektedir. Bu teknolojinin en temel özelliği farklı sunucularda bulunan verilerin birbirine bağlanabilmesidir. Bu bağlantılar, farklı veb sunucularından yayınlanan veb sayfaları arasındaki bağlantılardan ilham alınarak tasarlanmıştır. Örneğin, BBC⁵ kuruluşunun Wikipedia tanımını içeren veb sayfasından bu kuruluşun kendi veb sitesindeki kuruluş şemasının tanıtıldığı veb sayfasına⁶ bir bağlantı bulunmaktadır. Kullanıcılar bir veb sayfasından diğerine ulaşmak için bu bağlantıları izlerler ve veb üzerinde bu bağlantılar sayesinde oluşturulmuş büyük ağ içinde dolaşabilirler. Benzer şekilde veriler de veb sayfaları gibi HTTP ile erişilebilir olduğunda ve ilişkili verilere bu erişilebilir adresleri kullanarak bağlandıklarında veb üzerinde bir veri ağı oluşturulabilecektir. Böylece farklı yazılımlar arasında ilişkili verilere ulaşmak için gerekli bilgi verinin içinde tanımlı bağlarda olacak ve tüm yazılımlar bu bağları izleyerek diğer yazılımların verilerine kolayca ulaşabilecektir. Bağlı veri, bu temel fikir doğrultusunda tasarlanmış ve yazılımların kullanımına sunulmuştur.

Bağlı veri yaklaşımı veri şemalarını açık bir şekilde paylaşmayı, var olan şemaları yeniden kullanmayı öngörür. Böylece uygulamalar ortak verileri yorumlarken aynı dilden konuşabilecektir. Ayrıca SPARQL sorgu standardı sayesinde veriyi kullanan taraf her türlü karmaşık sorgu gereksinimini karşılayabilir. Dolayısıyla diğer teknolojilerde olduğu gibi yeni sorgu gereksinimlerinde veri sahibi tarafında bir bakım gereksinimi doğmaz. Bağlı veri yaklaşımı doğrultusunda veri bütünleştirme yapabilmek için gerçekleştirilecek temel adımlar vardır. Literatürdeki bağlı veri bütünleştirme süreçlerinde [10,4,1] ele alınan 3 temel adım aşağıda tanıtılmıştır.

1. *Üst-veri tanımlama*: Bu adımda kavramlar ve aralarındaki ilişkilerin ifade edildiği veri şemaları RDFS⁷/OWL⁸ gibi bağlı veri standartları kullanılarak tanımlanır. Bu aşamada genel sözlüklerin yeniden kullanımı birlikte işlerlik açısından önemlidir.
2. *Veri yayınlama*: Belirlenen sözlüğe dayalı olarak verilerin RDF biçimine dönüştürülmesi ve yayınlamasıdır. Yayınlama, verinin HTTP protokolü ile erişilebilir ve sorgulanabilir olmasını sağlamaktır.

² RDF, <http://www.w3.org/RDF/>. (Son erişim 8/7/2013)

³ HTTP, <http://www.w3.org/Protocols/HTTP/>. (Son erişim 8/7/2013)

⁴ SPARQL, <http://www.w3.org/TR/rdf-sparql-query/>. (Son erişim 8/7/2013)

⁵ BBC Wikipedia sayfası, <http://wikipedia.com/wiki/BBC>. (Son erişim 8/7/2013)

⁶ BBC kuruluş şeması sayfası, <http://www.bbc.co.uk/aboutthebbc/insidethebbc/managementstructure/bbcstructure/>. (Son erişim 8/7/2013)

⁷ RDFS, <http://www.w3.org/TR/rdf-schema/>. (Son erişim 8/7/2013)

⁸ OWL, <http://www.w3.org/TR/owl-overview/> (Son erişim 8/7/2013)

3. *Uygulama Geliştirme:* Yayınlanan RDF verisini kullanan uygulamalar geliştirilir. Verinin yayınlanma biçimi uygulama geliştirmeyi doğrudan etkiler. Uygulama geliştirici veri yayınlayıcıdan bağımsız olarak veriyi başka veriler ile ilişkilendirebilir.

Bu sürecin veri yayınlama adımımda birçok farklı mimari kullanılabilir. Veriler arasındaki bağların oluşturulması, yayınlanan verinin güncelliğinin sağlanması gibi gereksinimler ile ilgili kararlar veri yayınlama mimarisini etkiler. Burada tanıtılan süreç döngüsel olarak yürütülür. Her döngüde yeni veri eklendikçe üst-veri genişler ve gerektiği takdirde veri yayınlama ve uygulama geliştirme mimarileri yeniden ele alınır.

Bu çalışmada, Ege Üniversitesi'nde kullanılan yazılımlardan 4 tanesinin verileri baz alınarak bir bağlı veri altyapısı oluşturulmuştur. EgeLOD⁹ adı verilen bu altyapı, Bilgi Yönetim Sistemi (EBYS), Bilimsel Araştırma Projeleri Sistemi (BAP), Duyuru Panosu (EgeDuyuru) ve Kampüs Haritası (ECBS) yazılımlarının verisini bağlı veri standartlarıyla açık bir biçimde yayımlar. EBYS, öğrencilerin ders kayıtları için kullanılır ve bu nedenle ders ve akademisyen bilgisini tutar. BAP, bilimsel araştırma projelerinin süreçlerini yönetmek için kullanılır ve üniversitede yürütülen araştırma projeleri ve bu projelerde çalışan kişiler hakkında bilgi saklar. EgeDuyuru üniversitedeki fakültelerin ve bölümlerin duyurularını yapmak için kullanılırken, ECBS kampüs içerisindeki binaların coğrafi konumlarını gösterir. Üniversitede kullanılan bu sistemlerden üçü (EBYS, BAP ve EgeDuyuru) verilerini ilişkisel veritabanlarında saklarken, ECBS verisini bir XML dosyasında saklamaktadır.

Üniversite verisinin bir kısmı veb sayfalarından herkese açık olarak yayımlanırken bir kısmı da güvenlik ve gizlilik gibi nedenlerle yalnızca bazı yazılımlar arasında paylaşılabilir. Kurum-İçi veya kurumlar arası farklı bütünleştirme gereksinimlerine göre açık veya kapalı olarak bağlı veri paylaşımı yapılabilir. Bu çalışmada açık olarak yayımlanabilecek veriler seçilerek EgeLOD altyapısı geliştirilmiştir. Bu altyapının geliştirilmesi sürecinde farklı veri yayınlama mimari seçenekleri deneyimlenmiş ve edinilen deneyimler bu çalışmada açıklanmıştır.

Makalenin geri kalanı şu şekilde düzenlenmiştir. 2. bölümde farklı bağlı veri yayınlama mimarileri değerlendirilmiştir. 3. bölümde EgeLOD altyapısının dağıtık bağlı veri yayınlama mimarisi kullanarak oluşturulması ve bu mimari üzerinde bir uygulama geliştirilmesi, 4. bölümde ise geliştirilen mimarinin merkezleştirilmiş mimariye dönüşümü açıklanmıştır. 5. bölümde sonuç ve geleceğe dönük çalışmalar bulunmaktadır.

2 Bağlı Veri Yayınlama Mimarileri

Girişte tanıtılan bağlı veri bütünleştirme sürecinin en yoğun adımı bağlı veri yayınlamadır. Bağlı veri yayınlamada iki temel görev gerçekleştirilir. Bunlardan ilki ilişkisel veritabanı, XML, CSV vb. farklı biçimlerde saklanan verileri ortak bağlı veri dili olan RDF biçimine dönüştürmektir. İkincisi ise RDF biçimindeki

⁹ EgeLOD, <http://data.ege.edu.tr>. (Son erişim 8/7/2013)

ilişkili verilerin aralarındaki bağların keşfedilmesidir. Böylece, hem var olan veri hem de veri arasında olması gereken bağlar birlikte RDF biçiminde erişilebilir hale getirilir. Ancak, bu noktada RDF biçimindeki bu verinin hangi kanaldan yayınlanacağına karar vermek gerekir. Farklı bağlı veri yayınlama yöntemleri, farklı araçlar ve etkinlikler içerir. Bunun için Heath&Bizer tarafından da açıklanmış olan bu yöntemler aşağıda sıralanmıştır[3].

- **Veb içeriğine gömülü yayınlama:** HTML, XHTML gibi standartlar kullanılarak sunulmuş olan veb sayfalarına eklemeler yapılarak bağlı veri yayınlanabilir. RDFa¹⁰, Microdata¹¹ veya Microformats¹² standartlarıyla içeriğe yapılacak eklemeler insanlar için hazırlanan sayfaların yazılımlar tarafından da işlenebilmesine olanak tanımaktadır. Bu yöntem kullanıldığında var olan veb sayfalarından hızlıca RDF biçiminde veri yayınlanabilir ancak RDF verileri veb sayfalarının içine gömülü olacağından SPARQL ile sorgulanamaz. Bu yöntemle yayınlanan verileri sorgulamak isteyen uygulamalar belirli aralıklarla verileri toplayıp depolayarak sorgulayabilirler.
- **Yığın dosya halinde yayınlama:** Veritabanında saklanan ya da farklı bir biçimde tutulan veriler yığın halinde bir RDF dosyasına aktarılarak diğer yazılımların kullanımına sunulabilir. Bu yöntemle yayınlanmış verinin kullanılabilmesi için yığın dosyanın veb adresinden indirilerek bir RDF sunucusuna yüklenmesi ve böylece ilgili yazılımın sorgulayabileceği bir hale getirilmesi gerekmektedir. Bu yöntem, hem veri sağlayıcısının hem de veri tüketicisinin verinin güncellenmesi için bir çaba göstermesini gerektirir.
- **Erişilebilir RDF dosyaları halinde yayınlama:** RDF biçimindeki veriler dosyalar halinde saklanarak bir sunucudan erişilebilir hale getirilebilir. Bu durumda her RDF dosyası bir veri kaydını veya birden çok kaydı temsil eden kaynakları ifade eder. Her veri kaydına bir bağlı veri kaynağı olarak erişilebilmesi için her biri bir kayıt ile ilgili bilgileri içeren farklı dosyalar halinde yayınlanır. Bu yöntem halihazırda dosya olan verilerin dönüştürülüp yayınlanması için uygundur ve teknik olarak basittir. Ancak, bu yöntemle yayınlanan verilerin sorgulanması için, dönüştürülmüş verilerin veri tüketicisi tarafından toplanarak bir RDF sunucusuna kaydedilmesi gerekir. Ayrıca, çok değişen ve büyük veriler için birçok dosyanın yönetiminin zorluğu ve dosya-okuma yazma performansı, bu yöntemi görece az değişen ve az miktardaki veriler için daha uygun kılar.
- **RDF sunucularından yayınlama:** RDF sunucuları, büyük miktarda RDF verisinin SPARQL ile sorgulanabilmesi ve veri içindeki kaynakların HTTP protokolü ile erişilmesini sağlayan araçlardır. Yani RDF sunucuları hem bir SPARQL uç noktası hem de bir bağlı veri uç noktası içerirler. RDF sunucuları aynı zamanda bir RDF saklayıcı olabilir. İlişkisel veritabanlarının veri saklama yeteneklerini RDF biçimine özelleşmiş olarak sağlayan araçlara RDF (üçlü/dörtlü) saklayıcı adı verilmektedir. RDF saklayıcılar bu biçimde

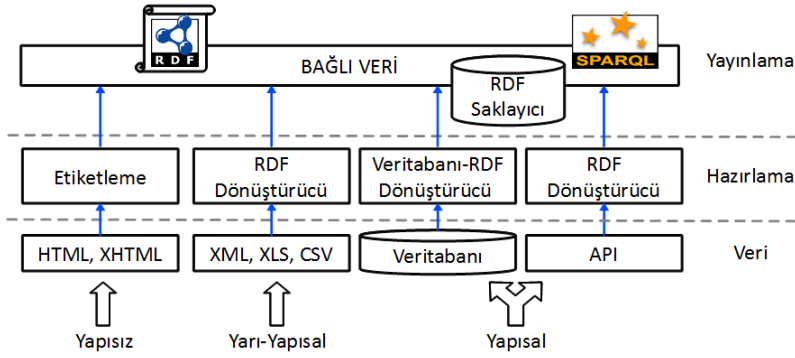
¹⁰ RDFa 1.1 Primer, Rich Structured Data Markup for Web Documents, <http://www.w3.org/TR/rdfa-primer/>. (Son erişim 8/7/2013)

¹¹ HTML Microdata, <http://www.w3.org/TR/microdata/>. (Son erişim 8/7/2013)

¹² Microformats, <http://microformats.org/>. (Son erişim 8/7/2013)

tanımlanmış veri saklar/sunular ve SPARQL ile sorgulanmasına olanak tanirlar. Baęlı veri uç noktası içermeyen RDF saklayıcılar için ek araçlar¹³ kullanılabilir. RDF sunucuları kullanarak baęlı veri yayınlamak en çok başvurulan yöntemdir. RDF sunucuları tüm verisini RDF biçiminde kullanan uygulamalar için de veritabanı görevi yapabilir. Ancak, yayınlanan veri başka biçimlerden RDF biçimine dönüştürülüyor ise RDF sunucunun aynı zamanda RDF dönüştürücü desteęi vermesi önemlidir. Bu desteęi veren sunuculara örnek olarak Virtuoso¹⁴ verilebilir. Bir başka örnek olan D2RQ¹⁵ ise doğrudan ilişkisel veritabanlarına baęlı veri ve SPARQL uç noktaları aracılığıyla erişim sağlayan bir RDF sunucudur¹⁶.

Farklı veri yayınlama yöntemleri Şekil 1'de bir arada resmedilmiştir. Veriler yapırsız, yapısal ve yarı-yapısal olmak üzere 3 biçimde temsil edilebilmektedir. HTML gibi metin biçimleri yapırsız, veritabanı veya uygulama programlama arayüzlerinden (API) erişilen veriler yapısal, XML ve CSV gibi yapırsını içinde barındıran veriler ise yarı-yapısaldır. Her veri biçimine özel dönüştürücüler yardımıyla baęlı veri meydana getirilir ve yukarıdaki açıklanan yöntemlerden biriyle sunulur.



Şekil 1. Baęlı veri yayınlama yolları

Bu çalışmada temel olarak RDF sunucularından baęlı veri yayınlama yöntemi kullanılmıştır. Ancak bu seçim yapıldığında da birçok farklı veri kaynağında bulunan verinin bütünleştirilerek yayınlaması için iki farklı yaklaşım ortaya

¹³ RDF saklayıcıları baęlı veri uç noktası olarak davranmasını sağlayan araçlara Pubby örnek verilebilir, <http://wifo5-03.informatik.uni-mannheim.de/pubby/>. (Son erişim 8/7/2013)

¹⁴ Virtuoso RDF Store, <http://virtuoso.openlinksw.com/rdf-quad-store/>. (Son erişim 8/7/2013)

¹⁵ D2RQ, <http://d2rq.org/>. (Son erişim 8/7/2013)

¹⁶ Bu gibi araçlar SQL sorguları kullanarak gereksinim duyulan veriye erişerek RDF biçiminde yayınlılar. Bu araçlar için R2RML (<http://www.w3.org/TR/r2rml/>) standardı tanımlanmıştır.

çıkılmaktadır: dağıtık yayınlama ve merkezi yayınlama. İki yaklaşımda da veri RDF sunucularından yayınlanır. Ancak, dağıtık yaklaşımda her veritabanı için bir RDF sunucusu kullanılırken, merkezi yaklaşımda tüm veritabanları bir RDF sunucuda birleştirilerek yayınlanır. Veri kaynaklarının çeşitliliği, veri tazeliği gereksinimi ve erişim hızı gibi etkenler dağıtık veya merkezi yayınlama seçiminde rol oynar. Bu noktada bu yöntemleri incelemek ve birbiriyle karşılaştırmak yararlı olacaktır.

Dağıtık Yayınlama: Bu yöntemde her farklı veri kaynağı için ayrı şemalar belirlenir ve bu şemalar doğrultusunda veri kaynakları RDF biçimine dönüştürülür. Dönüştürülen her veri kaynağı kendine özgü bir URI uzayında erişilebilir ve sorgulanabilir şekilde yayınlanır. Yayınlama verinin bulunduğu yerden canlı olarak dönüştürülmesi ya da yığın halinde dönüştürülerek başka bir yere kopyalanmasıyla yapılabilir. Daha sonra verilerde aynı nesnelere belirten kaynaklar bulunarak *owl:sameAs* ile; ilişkili olan diğer veriler de ilgili ilişkiler ile birbirine bağlanır. İki veri kümesi arasındaki bağlar, ayrı bir veri kümesinde veya ilgili veri kümelerinin içinde yayınlanabilir. Dağıtık yayınlama yöntemiyle yayınlanan veri üzerinde yazılım geliştirirken dağıtık sorgular kullanılır. Farklı SPARQL uç noktalarına gönderilen sorgu sonuçları birleştirilerek gerekli bilgiye ulaşılır.

Dağıtık yayınlamanın kazançları şöyle sıralanabilir:

- Her veri kaynağı birbirinden bağımsız olarak dönüştürülüp ilişkili veri kümelerine bağlanabildiği için esnek bir mimari ortaya çıkar. Kolayca genişleyebilir.
- Veriler canlı olarak dönüştürülürse her an güncel bağlı veriye erişilebilir.

Diğer yandan dağıtık yayınlamanın bazı sorunları da bulunmaktadır. Bunlar aşağıda sıralanmıştır:

- Dağıtık SPARQL uç noktaları üzerinde sorgu işletmek ağ maliyetinden dolayı bir performans kaybı yaratır.
- Her bir veri kaynağının şemalarının ayrı olarak belirlenmesi, veri modelini karmaşıktırır. Ayrıca, kurulan bağların yönü ve yayınlama biçimi gibi varsayımlar yazılımların işleyişine veya sorgulara gömülme zorundadır.

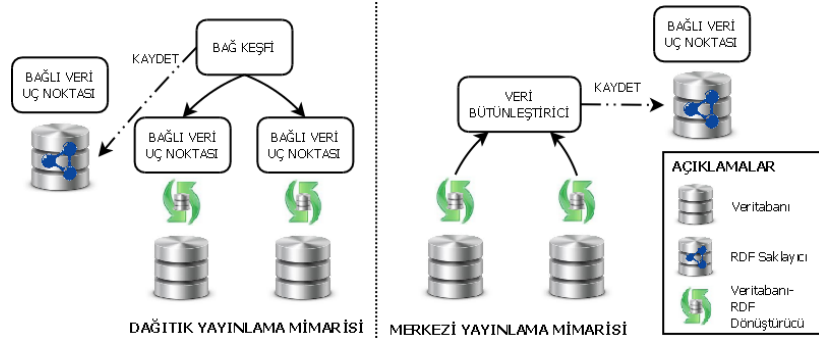
Merkezi Yayınlama: Bu yöntemde veriler tek bir RDF sunucuda toplanır ve buradan gereksinim duyan uygulamalara sunulur. Bunun için tüm veriyi kapsayan bütünlük bir veri şeması oluşturulur. Her veri kaynağından bu veri şemasının ilgili kısımlarına dönüşümler yapılır. Farklı veri kaynaklarında tekrar eden veriler var ise bunlar keşfedilir ve ortak RDF sunucusunda tek bir nesne temsil edecek şekilde bir araya getirilir. Bununla birlikte diğer bağlar da keşfedilir ve aynı sunucuda veri ile birlikte yayınlanır. Merkezi yayınlama yaklaşımında veriler ortak bir noktada kopyalanarak saklandığından kopyalanan verinin güncelliğini sağlayacak yöntemler uygulanır. Merkezi yayınlama yaklaşımının kazançları şöyle sıralanabilir:

- Daha büyük bir alanı temsil eden anlaşılır bir şema ve veri elde edilmiş olur.
- Veri tek kaynaktan sorgulandığı için daha performanslı olur.

Ancak verinin merkezi bir yere toplaması bazı sorunlara da yol açar:

- Veri kopyalanarak yayınlanacağından kopya verinin güncellenebilmesi için çeşitli mekanizmalara gereksinim duyulur.
- Çok fazla veri toplanması durumunda büyük verinin sorgulanması ile ilgili bazı sorunlar ortaya çıkabilir.

Dağıtık ve merkezi bağlı veri yayınlama yöntemleri Şekil 2’de gösterilmiştir. Şekilde, iki veritabanının, iki farklı mimari kullanılarak yayınlanması örneklendirilmiştir. Dağıtık bağlı veri yayınlama mimarisinde her bir veritabanının önünde RDF dönüştürücü ve sunucu görülmektedir. İki veritabanı arasındaki bağlantılar da örnek olarak üçüncü bir kaynaktan sunulmuştur. Merkezi yayınlama mimarisinde ise iki veri kaynağının önünde RDF dönüştürücü bulunmakta ve veriler tek bir RDF sunucuya kaydedilmektedir. Verinin kaydedilmesi sırasında keşfedilen ilişkiler kullanılarak veriler bütünleştirilmektedir.



Şekil 2. Dağıtık ve Merkezi Bağlı Veri Yayınlama Mimarileri

Dağıtık ve merkezi veri yayınlama mimarilerinin kayıp ve kazançları doğrultusunda bütünleştirilecek veriye göre seçim yapılır. Örneğin, bütünleştirilecek veri kaynaklarının tek bir sahibi var ise veya koordinasyon mümkünse merkezi yayınlama yöntemi kullanılabilir. Ayrıca, veri kümeleri arasında çok fazla bağlantı var ise veya veriler çok fazla tekrar ediyorsa merkezi mimari daha kullanışlı bir veri ortaya çıkarabilir. Ancak veri kaynaklarının çok çeşitli sahipleri varsa veya ortamdaki veri kaynaklarında sürekli artış/azalma olabiliyorsa dağıtık mimari esneklik sağlar. Üniversite alanından örnek vermek gerekirse üniversite içi yazılımların verilerini bütünleştirilmesi veri sahibi üniversite olduğu için merkezi mimari kullanılarak yapılabilirken birden çok üniversitenin verisini bütünleştirmek için dağıtık mimari daha uygun görünmektedir.

Bu makalede anlatılan EgeLOD bağlı veri altyapısının üretilmesi dağıtık ve merkezi veri yayınlama mimarilerinin deneyimlendiği bir ortam doğurmuştur. Çalışma iki ana aşamadan oluşmaktadır. İlk aşamada dağıtık yayınlama yöntemi kullanılarak veriler yayınlanmış, bağlanmış ve bu veri üzerinde bir uygulama

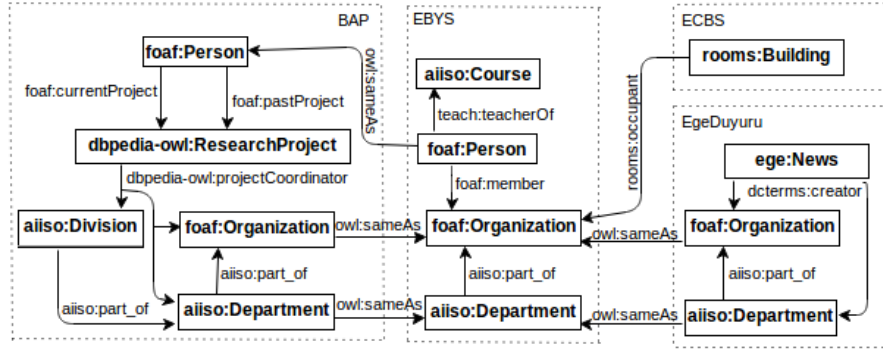
geliştirilmiştir. ikinci aşamada ise edinilen deneyimler doğrultusunda EgeLOD altyapısı elden geçirilerek bağlı veri yayınlama mimarisi merkezileştirilmiştir. Sonraki bölümlerde EgeLOD mimarisinin şekillenme süreci anlatılmıştır.

3 EgeLOD Dağıtık Bağlı Veri Mimarisi

EgeLOD bağlı veri altyapısının geliştirimine başlanırken ilk aşamada dağıtık yayınlama mimarisi hedeflenmiştir. Bu mimarinin yukarıda tanımlanan bağlı veri bütünleştirme sürecine uygun olarak gerçekleştirilen adımlar bu bölümde açıklanmıştır. Bu sürecin son adımında kullanıcıların üniversitedeki ilgili veriye kolayca erişmesini sağlayan Üniversite Kampüs Asistanı adında bir uygulama geliştirilmiş ve bundan edinilen deneyimler anlatılmıştır.

3.1 Üst-verilerin Tanımlanması

EgeLOD altyapısının ilk sürümünde EBYS, BAP, EgeDuyuru ve ECBS yazılımlarının verisinin yayınlanması hedeflenmiştir. Bu amaçla her bir yazılımın verisi analiz edilmiş ve 4 farklı üst-veri tanımlanmıştır. Tanımlanan üst-verilerde yer alan kavramlar ve aralarındaki ilişkiler Şekil 3'de gösterilmiştir.



Şekil 3. EgeLOD dağıtık yayınlama mimarisi üst-veri sözlükleri

Üst-veri sözlükleri belirlenirken daha önce tanımlanmış ve vebde var olan sözlükler yeniden kullanılmıştır. Academic Institution Internal Structure Ontology (aiiso), Friend of a Friend (foaf), Teaching Core Vocabulary (teach), Dublin Core (dcterms), DBpedia Ontology (dbpedia-owl), ResumeRDF Ontology (cv) ve DERI Buildings and Rooms Vocabulary (rooms) EgeLOD üst-verilerinde yeniden kullanılan kavramları içeren sözlüklerdir.

3.2 Bağlı Veri Yayınlama

Bağlı veri kullanılarak bütünleştirilecek yazılımlardan EBYS, BAP ve EgeDuyuru verilerini ilişkisel veritabanlarında saklarken ECBS verisini XML biçiminde bir

dosyada saklanmaktadır. İlgili üst-veri tanımlamalarının ardından bu verilerin üst-veri sözlüklerine uygun olarak RDF biçiminde dönüştürülmesi gerçekleştirilmiştir. Bu amaçla ilişkisel veritabanlarından üst-veri sözlüklerine D2RQ eşlemeleri oluşturulmuş ve her bir veritabanı D2RQ sunucusu yardımıyla bağlı veri biçiminde yayınlanmıştır. ECBS verisinin dönüşümü için ise Tripliser¹⁷ eşlemeleri kullanılmış ve Joseki¹⁸ SPARQL uç noktası, Pubby bağlı veri uç noktası ve Tripliser XML-RDF dönüştürücü bir arada kullanılarak XML verisi RDF biçiminde yayınlanmıştır. Şekil 4'te bulunan mimari 4 farklı veri kümesinin bağlı veri uç noktaları ile yayınlanmasını göstermektedir. Şekilde bağlı veri uç noktası SPARQL sorgularına ve RDF kaynağı erişim isteklerine yanıt verme yeteneklerine sahip bütünlük bir bileşen olarak ele alınmış ve 4 bağlı veri uç noktası tek bir katman olarak gösterilmiştir.

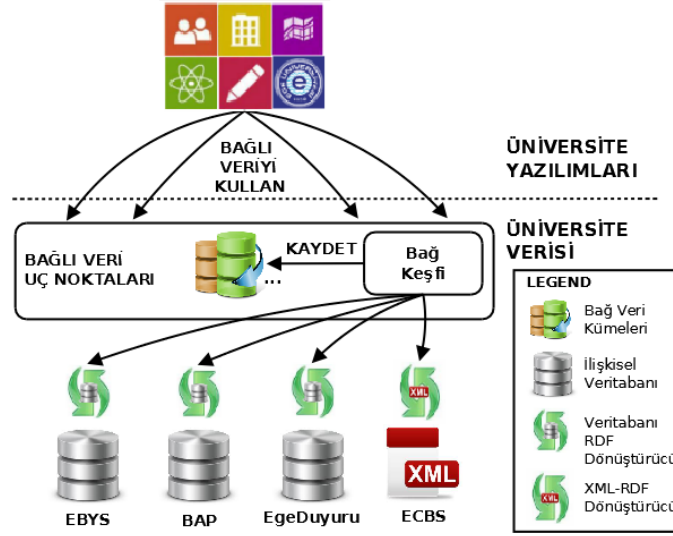
Yayınlanan bağlı verilerin ilişkilendirilmesi ve tekrar eden verilerin belirlenmesi mimaride Bağ Keşfi olarak gösterilmiştir. Bu amaçla Silk¹⁹ bağ keşif çerçevesi kullanılmıştır. EYBS ve BAP veritabanları arasında *foaf:Person* sınıfının örnekleri birbirlerine, EgeDuyuru ve BAP veritabanlarındaki fakülte ve bölümler ise EBYS veritabanındaki fakülte ve bölümlere bağlanmıştır. Bu kavramlar farklı sistemlerde tekrar ettiği için bu bağlarda *owl:sameAs* ilişkisi kullanılmıştır. ECBS veritabanındaki bina bilgileri de *rooms:occupant* ilişkisi ile EBYS verisindeki bölüm ve fakültele bağlanmıştır. Verinin bütünlüğünü sağlayan bu bağlar keşfedildikten sonra Şekil 4'te gösterildiği gibi bağlı veri uç noktaları katmanında RDF biçiminde yayınlanmıştır. EBYS, BAP ve EgeDuyuru D2RQ ile yayınlandığından bölüm/fakülte bağları ve öğretim üyeleri bağları bu veri kümeleri içinde yayınlanamamış; onlar için iki yeni SPARQL uç noktası oluşturulmuştur. Coğrafi bilgiler ile bölümler/fakülteler arasındaki bağlar ise doğrudan ECBS verisinin içinde saklanmıştır.

Silk ile bağların keşfedilmesini örneklemek için EBYS ve BAP arasındaki öğretim üyelerinin bağlanması ele alınabilir. Silk, bu iki veri kümesinde bulunan *foaf:Person* sınıfının tüm örneklerini birbirleriyle karşılaştırmak için ayarlanmıştır. İki veri kümesinde de *foaf:firstName* ve *foaf:familyName* özellikleri eşleme için kullanılmıştır. Karşılaştırma yapabilmek için isimlerin içindeki alfabetik olmayan karakterler *alphaReduce* dönüştürücü fonksiyonu ile kaldırılmıştır. Büyük harfle başlayan isimlerdeki harflerin hepsi küçük harf dönüştürme fonksiyonu ile küçük harflere çevrilmiştir. İsimler bu şekilde eşleştirme için hazırlandıktan sonra karşılaştırma operatörleri iki kaynağı karşılaştırır. İsimlerin benzerlikleri hesaplanırken uzaklık ölçümü ve eşik değerinden faydalanılır. Harfler üzerine uzaklık ölçümü yapan Jaro-Winkler, adlar için uygun olduğundan bu çalışmada kullanılmıştır. Benzerlik ölçümünün daha hassas olması adına eşik değeri 0.2 belirlenmiştir. Aynı zamanda benzerlik ölçümünde hem adların hem de soyadların aynı ağırlığa sahip olması için iki özelliğin de karşılaştırmadaki ağırlıklarına 1 değeri atanmıştır.

¹⁷ Tripliser XML-RDF dönüştürücü, <http://daverog.github.io/tripliser/>. (Son erişim 8/7/2013)

¹⁸ Joseki SPARQL uç noktası, <http://joseki.sourceforge.net/>. (Son erişim 8/7/2013)

¹⁹ Silk bağ keşif çerçevesi, <http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/>. (Son erişim 8/7/2013)



Şekil 4. EgeLOD Dağıtık Bağlı Veri Yayınlama Mimarisi

Bu örneğe ek olarak EgeDuyuru ile EBYS arasında bölüm/fakülte bağları ve ECBS ile EBYS arasındaki bina ilişkileri benzer şekilde Silk fonksiyonları kullanılarak oluşturulmuştur.

3.3 Uygulama Geliştirme Deneyimleri

Bağlı veri bütünleştirme sürecinin şimdiye kadarki kısımları bağlı veri standartları ile yayınlanmış bütünleşik bir bağlı veri ağı ortaya çıkarmıştır. Bağların yayınladığı veri kümeleri dahil 6 farklı SPARQL uç noktasını sorgulayarak yazılımlar bu bütünleşik veriden yararlanabilmektedir. Şekil 5'te örnek ekran görüntüsü verilen Kampüs Asistanı veb uygulaması²⁰ ile bütünleşik veri öğrencilere sunulmuştur. Daha önce ilişkili verilere ulaşmak için defalarca farklı veb sayfalarında arama yapmak zorunda kalan kullanıcılar bu uygulama yardımıyla 4 farklı sistemdeki ilişkili bilgilere bir kavram üzerinden erişebilmektedir. Örneğin, şekilde görülen bir öğretim üyesinin sayfasında öğretim üyesinin verdiği dersler, öğretim üyesinin yürüttüğü araştırma projeleri, öğretim üyesinin ders verdiği bölümün coğrafi konumu ve bu bölüm ile ilgili duyurular bir arada görüntülenebilmektedir. Yine bu veriye proje, ders, bölüm gibi farklı bakış açılarından da ulaşılabilir.

Bir lisans öğrencisi tarafından dağıtık SPARQL sorguları kullanılarak geliştirilen Kampüs Asistanı, seçilen bağlı veri yayınlama mimarisinin iki güçlüğünü göz önüne sermiştir. Bunlardan birincisi dağıtık sorguların yanıt zamanının uzun olmasıdır. Bunun bir nedeni veritabanından canlı dönüştürme yoluyla RDF üretilmesi, ikincisi ise dağıtık veri kümeleri arasında dolaşmanın bir ağ maliyeti oluşturmasıdır.

²⁰ <http://data.ege.edu.tr/app/> (Son erişim 26/8/2013)



Şekil 5. Kampüs Asistanı uygulamasından bir ekran görüntüsü

Şekil 6'da öğretim üyesi üzerinden ilişkili verilere ulaşan sorgular gösterilmiştir. Her sorgu EBYS veri kaynağından bağları kullanarak diğer veri kaynaklarına atlamakta ve ilişkili veriyi getirmektedir. Uygulamanın bu sürümünde tüm bu sorguların işletilmesi yaklaşık olarak 10 saniye sürmektedir. Veritabanı şeması ile kullanılan üst-veri sözlüğü arasındaki farklılık nedeniyle verinin anlık olarak dönüştürülmesi bu kayıpta önemli bir rol oynamaktadır.

Uygulamanın işleyişi sırasında karşılaşılan ikinci güçlük, veri kümeleri arasındaki bağların eskimesi olmuştur. Şimdiye kadar üretilmiş tüm verinin hızlıca bağlanması için bağlar verinin kopyası üzerinde oluşturulduğundan veriler güncelendikçe yeni bağların da üretilme gereksinimini ortaya çıkarmıştır.

Bu aşama sonucunda, dağıtık yayınlama mimarisinin getirdiği yeni veri kümesi ekleme esnekliği ve veritabanlarından canlı olarak güncel verinin yayınlanması kazançları elde edilmiştir. Ancak, veri kaynaklarının çok fazla bağ içermesi ve dağıtık veri kümeleri arasında bağ keşfi gereksiniminin tam olarak karşılanamaması nedeniyle söz edilen eksiklikler ortaya çıkmıştır. Bu eksiklikleri ortadan kaldırmak için bağlı veri bütünleştirme sürecinde yeni bir döngüye başlanarak yayınlama mimarisi merkezleştirilmiştir. Bu döngünün gerçekleştirimi bir sonraki bölümde açıklanmıştır.

4 EgeLOD Merkezleştirilmiş Bağlı Veri Mimarisi

İlk döngü sonunda ortaya çıkan uygulama dağıtık yayınlama mimarisinin sorunlarını yaşadığı için bağlı veri yayınlama mimarisinin merkezleştirilmesine karar verilmiştir. Bağlı veri yayınlama merkezleştirilirken ortak bir noktadan yayınlanan verinin günceliğini koruması ve veri bütünleştirmeyi sağlayan bağların keşfinin güncellenmesi de birlikte ele alınmıştır. Bu amaçla Şekil 7'de gösterilen EgeLOD altyapısının yeni mimarisindeki Değişiklik İşleyici adlı araç geliştirilmiştir.

```

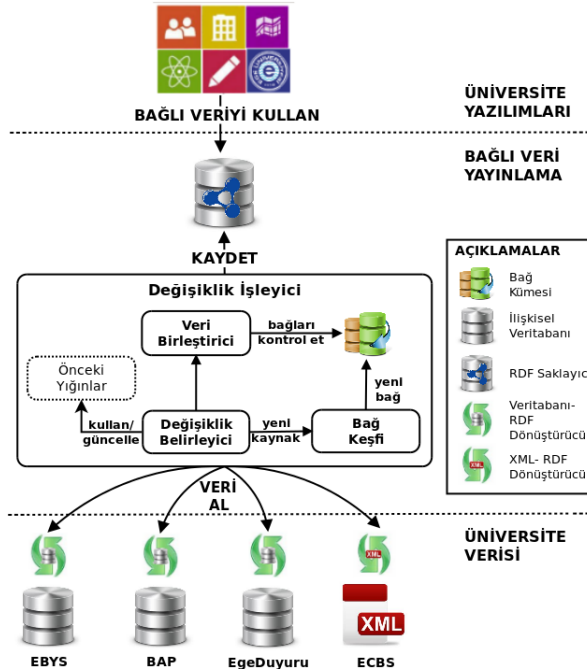
SELECT * {
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/lecturerlinks> {
    ?projectAcademician owl:sameAs ogr:1730 .
  }
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/projeler> {
    ?projectAcademician foaf:currentProject ?project .
    ?project ?pred ?obj .
  }
}

SELECT * {
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/dersler> {
    ogr:1730 teach:teacherOf ?course .
    ?department aiiso:teaches ?course .
  }
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/organizationlinks> {
    ?announcer owl:sameAs ?department.
  }
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/duyurular> {
    ?news dcterms:creator ?announcer .
    ?news ?pred ?obj .
  }
}

SELECT * {
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/dersler> {
    ogr:1730 teach:teacherOf ?course .
    ?department aiiso:teaches ?course .
  }
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/geodata> {
    ?building rooms:occupant ?department .
    ?building geo:lat ?lat .
    ?building geo:long ?long .
  }
}

```

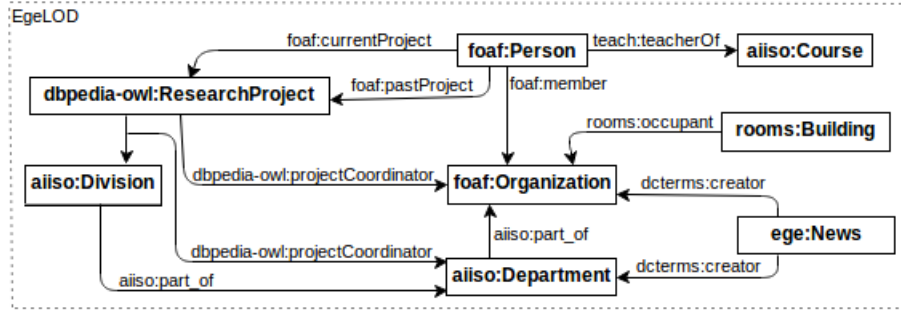
Şekil 6. Bir öğretim üyesinin bilgisini toplayan dağıtık sorgular



Şekil 7. Veriyi tek bir RDF saklayıcıda birleştiren yayınlama mimarisi

Bağlı veri yayınlama mimarisinin değiştirilmesi amacıyla bağlı veri bütünleştirme döngüsüne tekrar başlanarak öncelikle tanımlanan üst-veri sözlükleri bütünleştirilmiştir. Şekil 8’de gösterilen üst-veri sözlüğünde *owl:sameAs* bağlantıları kaldırılmış ve veri modeli tek bir sözlük haline getirilmiştir. Bu doğrultuda veri de birleştirilmiş ve farklı veritabanlarındaki her bir fakülte, bölüm ve öğretim üyesi tek bir kaynakla temsil edilmiştir. Böylece ilişkili veriye ulaşmak için veri kaynakları üzerinde geçiş yapma zorunluluğu ortadan kalkmıştır. Bunun bir örneği olarak Şekil 9’da bir öğretim üyesinin bölümünün duyurularını getiren sorgunun nasıl değiştiği gösterilmiştir. Sorgu, tek bir uç noktaya gönderilmekte ve *owl:sameAs* kullanma gereği olmadan ilişkili bilgiyi verebilmektedir.

Üst-verinin elden geçirilmesinden sonra bu döngüde yayınlama mimarisine katılan Değişiklik İşleyici, veri kaynaklarından veriyi alarak yeni veriler için bağ keşfini gerçekleştirmekte ve belirlenen tekrar verileri tek bir kaynaktan birleştirmektedir. Birleştirme sırasında ortak kaynak olarak EBYS kaynakları kullanılmıştır. Değişiklik İşleyici birleştirdiği veriyi bir RDF sunucusuna kaydetmektedir.



Şekil 8. Merkezileştirilmiş EgeLOD mimarisi üst-veri sözlüğü

```

SELECT * {
  SERVICE <http://data.ege.edu.tr/endpoint/repositories/egelod> {
    ogr:1730 teach:teacherOf ?course .
    ?department aiiiso:teaches ?course .
    ?news dcterms:creator ?department .
    ?news ?pred ?obj .
  }
}

```

Şekil 9. Öğretim üyesinin bölümünün duyurularını getiren sorgunun değişimi

Veri bütünleştirmeye katılan yazılımlar, ilişkisel veritabanlarını güncelleme-ye devam etmektedir. Değişiklik İşleyici bileşeni, günlük olarak her veritabanını N-Triples²¹ biçiminde bir RDF yığın dosyası haline getirir. İlişkisel veritaban-

²¹ <http://www.w3.org/TR/rdf-testcases/#ntriples> (Son erişim 26/8/2013)

ları D2RQ ile, XML biçimindeki veri kaynağı ise Tripliser aracı kullanılarak yığın dosyaya saklanmıştır. Verinin eldeki sürümü ile aradaki değişiklikleri belirleyebilmek için yeni yığın dosya oluşturulmadan önce bir önceki yığın dosya saklanır. Bağlı veri altyapısında bulunan 4 veri kaynağındaki verinin toplam miktarı yaklaşık 800 bin üçlüdür. 2 GB RAM ve UltraSPAC T1 işlemciye sahip sunucumuzda tüm yığın dosyaların oluşturulması 7 dakikanın altında tamamlanmaktadır.

Şekil 7’de görüldüğü gibi, Değişiklik İşleyici, 3 alt-bileşenden oluşur: Değişiklik Belirleyici, Bağ Keşfi ve Veri Birleştirici. Değişiklik İşleyici bir veri kaynağının yeni yığın dosyasını elde ettikten sonra, Değişiklik Belirleyici güncellemelerin belirlenmesi için bu yığını bir önceki yığın dosya ile karşılaştırılır. Bu karşılaştırma, iki yığının da bellekte Jena²² Model’lerine okunması ve bu sınıfın *difference* metodu kullanılması ile yapılır. 4 veri kaynağının güncellemelerinin belirlenmesi toplamda 30 saniyeden daha kısa bir sürede tamamlanmaktadır.

Değişiklik Belirleyici, güncelleme içinde yeni kaynaklar ile karşılaştığında bunları Bağ Keşfi at-bileşenine aktarır. Bağ Keşfi, Silk çerçevesini ve verilmiş bağ eşleme tanımlarını kullanarak yeni gelen bir kaynak ile var olanlar arasındaki bağlantıları araştırır. EgeLOD altyapısının bu sürümünde BAP, EgeDuyuru ve ECBS veri kaynakları EBYS veri kaynağına bağlandığından EBYS yığını doğrudan ortak RDF sunucusuna aktarılır. Ardından diğer 3 veri kümesindeki yeni kaynakların EBYS ile bağlantısı belirlenir. Belirlenen bağlantılar bir başka RDF sunucuda saklanır ve böylece tekrar bağ keşfi gereksiniminin önüne geçilir. Keşif performansı altyapının var olan durumunda yeterlidir. Örneğin, BAP veri kümesindeki 10 yeni akademisyenin EBYS’deki 4 bin akademisyen ile karşılaştırılarak bağlantılarının keşfedilmesi 2 saniyenin altında tamamlanmaktadır. BAP veri kümesindeki 2 bin akademisyen kaynağının EBYS’deki 4 bin akademisyen kaynağı ile karşılaştırılması da yaklaşık 3 dakika sürmektedir.

Son olarak veriyi ortak RDF sunucuya kaydetmeden önce Veri Birleştirici tüm bilgiyi EBYS URI’lerinde birleştirir. Bu amaçla keşfedilen *owl:sameAs* bağlantıları doğrultusunda kaydedilen veri içindeki kaynak EBYS URI’si ile değiştirilir. ECBS verisi, *rooms:occupant* özelliği kullanılarak bağlandığından birleştirilmeye ihtiyaç duymaz. Diğer yandan BAP ve EgeDuyuru veri kümelerinde yer alan kuruluş ve kişi tanımları eş olduğu keşfedilen EBYS kaynaklarına birleştirilir. Veri Birleştirici, güncelleme sırasında kaldırılan üçlüler için de birleştirme uygulayarak bütünlük veriyi günceller.

EgeLOD altyapısının bu sürümünde bütünlük veri Sesame²³ sunucusunda saklanmıştır. Verideki kaynakların veb üzerinden erişilebilir olması da Pubby ile sağlanmıştır.

Dağıtık mimaride, bağ keşfi, verinin kopyası üzerinde yapılmıştır. Veri yayınlama mimarisinin bu sürümünde, veri güncellendikçe yeni kaynaklar için bağ keşfi yapılmaktadır. Kampüs Asistanı uygulamasının veriye tek bir RDF sunucudan erişmesi performans sorunlarını büyük ölçüde gidermiştir. Geliştirilen Değişiklik İşleyici bileşeni ile verinin tazeliği sağlanmaya çalışılmıştır. Bu bileşenin veriyi

²² Apache Jena RDF Framework, <http://jena.apache.org/>. (Son erişim 26/8/2013)

²³ Sesame RDF Deposu, <http://www.openrdf.org/>. (Son erişim 8/7/2013)

kontrol etme aralığının kısa olması daha taze bir bağlı veri sağlarken değişiklik işleyici görevini üstlenen makine üzerinde bir iş gücü ortaya çıkarmaktadır.

5 Sonuç

Bu makalede tanıtılan bağlı veri yaklaşımı ile veri yayınlama ve bütünleştirme yöntemleri doğrultusunda Ege Üniversitesi'nin bağlı veri yayınlama mimarisinin gelişimi açıklanmıştır. Geliştirilen altyapı üzerinde Kampüs Asistanı adında bir uygulama geliştirilerek bağlı veri teknolojileri ile veri bütünleştirme süreci deneyimlenmiştir. Bu çalışmadan edinilen deneyim bağlı veri yayınlama mimarisinin yaşayan bir ürün olduğu ve gereksinimler doğrultusunda evrilme gereksinimi olduğunu göstermiştir. Ayrıca, bu çalışma ile üniversite içi ve üniversiteler arası veri bütünleştirme gereksinimlerini karşılayabilecek mimari seçenekler tanıtılmıştır.

Veb üzerinde veri bütünleştirme yeteneğinin açık veri kavramı ile birleşmesi, şimdiye kadar maliyetli olduğu için gerçekleştirilememiş birçok gereksinimin karşılanması için uygun bir ortam oluşturmaktadır. Bu makalede tanıtılan altyapı ile bölüm duyurularının mobil uygulamalardan erişilmesi, bölüm/fakültelerin bilimsel araştırma projeleri bazında analiz edilmesi gibi bazı çalışmalar üniversite yönetiminden ve bürokratik süreçlerden bağımsız olarak başlamıştır. Bu ortam, verinin bütünleştirilmesi ile ortaya çıkacak yaratıcı fikirlere de açık durumdadır. Ayrıca, bağlı veri yayınlama süreci, üniversite verisinin başlı başına bir değer olarak ortaya konmasını sağlamıştır. Tanımlanan üst-veri sözlüğü, zaman içinde genişleyerek üniversite için geliştirilen birçok yazılıma ışık tutabilecektir.

Ege Üniversitesi bağlı veri bütünleştirme çalışması doğrultusunda gelecekte, öncelikle kütüphane gibi veri kaynaklarını da bağlı açık verisine dahil etmeyi ardından açık olmayan veriler arasında da bütünleştirmeler yapmayı hedeflemektedir. Bunların yanında, başka üniversiteler ile de bağlı açık veri ağı örnekleri oluşturarak teknolojinin yeteneklerini deneyimlemek hedeflenmektedir.

Kaynaklar

1. Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N. Mendes, Bert Van Nuffelen, Claus Stadler, Sebastian Tramp, and Hugh Williams. Managing the life-cycle of linked data with the lod2 stack. In *International Semantic Web Conference (2)*, pages 1–16, 2012.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.
4. Bernadette Hyland and David Wood. The Joy of Data - A Cookbook for Publishing Linked Government Data on the Web Linking Government Data. In David Wood, editor, *Linking Government Data*, chapter 1, pages 3–26. Springer New York, New York, NY, 2011.

5. Carsten Kessler and Tomi Kauppinen. Linked open data university of muenster—infrastructure and applications. In *Demos the Extended Semantic Web Conference 2012 (ESWC2012)*, Heraklion, Crete, Greece, May 2012.
6. Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media meets semantic web - how the bbc uses dbpedia and linked data to make connections. In *ESWC*, pages 723–737, 2009.
7. Yuanchao Ma, Bin Xu, Yin Bai, and Zonghui Li. Building linked open university data: Tsinghua university open data as a showcase. In *JIST*, pages 385–393, 2011.
8. Sean O’Riáin, Edward Curry, and Andreas Harth. XBRL and Open Data for Global Financial Ecosystems: A Linked Data Approach. *International Journal of Accounting Information Systems*, 13(2):141–162, 2012.
9. François-Paul Servant. Linking enterprise data. In *LDOW*, 2008.
10. Boris Villazón-Terrazas, Luis Vilches, Oscar Corcho, and Asunción Gómez-Pérez. Methodological guidelines for publishing government linked data. In David Wood, editor, *Linking Government Data*, chapter 2. Springer, 2011.
11. Fouad Zablith, Mathieu d’Aquin, Stuart Brown, and Liam Green-Hughes. Consuming linked data within a large educational organization. In *COLD*, 2011.

EK A. TERİMLERİN İNGİLİZCE KARŞILIKLARI

Türkçe	İngilizce
Anlamsal Veb	Semantic Web
Bağ Keşfi	Link Discovery
Bağı Veri	Linked Data
Dağıtık Sorgulama	Federated Query
Eşgüdüm	Coordination
RDF Dönüştürücü	RDF Converter
RDF Saklayıcı	RDF Store
RDF Sunucu	RDF Server
Sözlük	Vocabulary
Uç nokta	Endpoint
URI uzayı	URI space
Üçlü	Triple
Üst-veri	Meta-data
Veri kümesi	Dataset
Yarı-yapısal	Semi-structured
Yapısal	Structured
Yapısız	Unstructured
Yayınlama	Publishing
Yığın dosya	Dump File