

# Ontoloji Yeniden Kullanımı için Bulut Tabanlı Bir Yazılım Mimarisinin Gerçekleştirilmesi

Cemil Abiş<sup>1</sup>, Görkem Giray<sup>2</sup>, Murat Osman Ünalır<sup>3</sup>

<sup>1</sup> Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir  
91110019563@ogrenci.ege.edu.tr

<sup>2</sup> gorkemgiray@gmail.com

<sup>3</sup> Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir  
murat.osman.unalir@ege.edu.tr

**Özet.** Ontolojiler, uygulamalar arası birlikte çalışabilirliğin ve bilgi paylaşımının önemli bileşenlerinden biridir. Başta tıp ve biyoinformatik olmak üzere, büyük boyutlardaki bilginin etiketlenmesi, getirilmesi ve paylaşımı için başarılı bir şekilde kullanılmaktadır. Ancak birlikte çalışabilirliğin daha etkin bir şekilde sağlanabilmesi ve geliştirim maliyetlerinin azaltılabilmesi için mevcut ontolojilerin yeniden kullanımı en önemli geliştirme adımıdır. Birçok ontoloji geliştirme süreci yeniden kullanıma dikkat çekerken bu gereksinimi tüm aşamalarıyla karşılayacak bir yöntem bulunmamaktadır. Bu çalışmada, ontoloji yeniden kullanımının tüm aşamalarını kapsayacak bulut tabanlı bir yöntem ve uygulama mimarisi önerilmektedir.

**Anahtar Kelimeler:** Ontoloji, Ontoloji geliştirme, Ontoloji yeniden kullanımı, Alt ontoloji çıkarma, Anlamsal ilintililik.

## Implementing a Cloud-Based Software Architecture for Ontology Reuse

**Abstract.** Ontologies are one of the important components of interoperability between applications and knowledge sharing. They are used for annotation, retrieval and sharing of huge volume of data in many areas, especially in medicine and bioinformatics. However, to provide an efficient interoperability and decrease costs of ontology development, reuse of existing ontologies is a critical step for development process. While most of the ontology development methodologies emphasize the importance of reuse, there is no method that completely supports all phases of reuse process. In this work, we recommend a cloud-based method and application architecture that includes and supports all steps of ontology reuse.

**Keywords:** Ontology, Ontology development, Ontology reuse, Sub-ontology extraction, Semantic relatedness.

## 1 Giriş

Yazılım geliştirme başta olmak üzere birçok disiplinde mevcut bileşenlerin ve teknolojilerin yeniden kullanımı, süreçlerin kalitesini arttıran ve maliyetleri azaltan temel unsurdur. Benzer koşullar ontoloji geliştirme süreçleri için de geçerlidir. Bir alanda geliştirilmiş mevcut ontolojilerin yeniden kullanımı geliştirilecek yeni ontolojilerin kalitesini yükselteceği gibi uygulamalar arasındaki bilgi paylaşımı ve birlikte çalışabilirliği de arttıracaktır [1].

Ontoloji geliştirme süreçlerinin birçoğu mevcut ontolojilerin yeniden kullanımını önem verilmesi gereken bir adım olarak tanımlasa da bu alandaki çalışmaların sayısı kısıtlıdır. Ontolojilerin daha modüler hale getirilmesi üzerine geliştirilmiş çalışmalar ontoloji yeniden kullanımındaki problemleri çözmeye çalışmaktadır. Ancak tüm yeniden kullanım sürecini bütünlük olarak ele alan bir yaklaşım bulunmamaktadır.

Bu çalışmada, ontoloji yeniden kullanımını daha etkin hale getirecek bir yöntem ve bu yöntemi destekleyen bir yazılım mimarisi önerilmektedir. Yöntemin ve dayandığı yazılım mimarisinin temel özellikleri bulut tabanlı olması ve ölçeklenebilirliktir. Ayrıca kullanılabilirlik problemi de çözülmeye çalışılmaktadır. 2. bölüm, ontoloji yeniden kullanım bağlamını ve ilgili çalışmaları aktarmaktadır. 3. bölümde geliştirilen yöntemin ve veri mimarisinin detaylarına değinilecektir. 4. bölümde yeniden kullanımı destekleyen Protege eklentisi hakkında bilgi verilecektir. Son bölümde ise yapılan çalışmalar değerlendirilecek ve ilerleyen dönemde gerçekleştirilmesi planlanan çalışmalar hakkında bilgi verilecektir.

## 2 Bağlam ve İlgili Çalışmalar

Bu bölümde, ontoloji yeniden kullanımını desteklemek için önermiş olduğumuz yöntemin kapsamında yer alan temel bilgilere değinilecektir. Mevcut ontolojilere nasıl erişilebileceği konusu ontolojiler ve ontoloji kaynakları başlığı altında aktarılacaktır. Ontoloji geliştirme süreçleri ve yeniden kullanım diğer bölümlerin içeriğini oluşturacaktır.

### 2.1 Ontolojiler ve Ontoloji Kaynakları

Temel amacı bilgi paylaşımı ve yeniden kullanım olan ontolojiler bilgi yönetimi, bilgi edinimi, doğal dil işleme ve yapay zeka gibi birçok alanda kullanılmaktadır [1]. Uzun zamandır var olan ontoloji ve ontoloji geliştirme kavramları, Tim Berners Lee tarafından ortaya konulan anlamsal web vizyonu ile daha çok ilgi çekmiş ve bu alandaki çalışmaların sayısı hızla artmıştır [2].

Ontoloji sayısının artması ile birlikte bu ontolojilere erişim sağlayacak, kullanıcıların ontolojileri aramasını ve kullanmasını kolaylaştıracak kaynaklara ihtiyaç duyulmuştur. Swoogle [3] ve Watson [4] gibi ontoloji arama motorları web üzerindeki ontolojileri tarayıp indeksleyerek arama imkanı sağlamışlardır. Arama motorları dışında, ontoloji deposu veya ontoloji kütüphanesi olarak adlandırılan, bir dizi ontolojiyi barındıran ve kullanıcıların bu ontolojileri bulup kullanabilmesini sağlayan web tabanlı

sistemler de ortaya çıkmıştır [5]. Bu sistemler daha çok belirli bir kapsam ve alan dahilindeki benzer ontolojileri barındırmaktadır. Örneğin; Bioportal<sup>1</sup>, OBO Foundry<sup>2</sup> ve OLS<sup>3</sup> gibi ontoloji kütüphaneleri tıbbi ontolojilere erişim için önemli kaynaklardır. Bu kaynaklar dışında kendi sunucuları üzerinde barındırılan Dbpedia<sup>4</sup>, Yago<sup>5</sup> ve Geonames<sup>6</sup> gibi büyük ontolojiler de bulunmaktadır. Bu ontolojiler SPARQL uç noktaları ve ontoloji dosyalarının doğrudan indirilmesi gibi yöntemlerle erişim sunmaktadır.

## 2.2 Ontoloji Geliştirme Süreçleri

Ontolojilerin de yazılım ürünlerinde olduğu gibi, önceden tanımlı adımları içeren bir metodoloji kullanılarak geliştirilmesi, ontolojinin kalitesi ve geliştirme sürecinin yönetimi bakımından önemlidir. Bu gereksinim doğrultusunda ontoloji geliştirmeye yönelik çok sayıda metodoloji önerilmiştir.

Uschold ve King [6], ontoloji geliştirme süreci için ilk yöntemlerden birini önermiştir. Bu yöntem, Enterprise ontolojisinin geliştiriminde uygulanmıştır. İlk yöntemlerden bir diğeri Gruninger ve Fox [7] tarafından önerilen ve TOVE (Toronto Virtual Enterprise) projesi kapsamında uygulanan yöntemdir. Uschold ve King, yöntemlerinde ontoloji yeniden kullanımına açık bir şekilde yer verirken, Gruninger ve Fox yeniden kullanıma değinmemiştir. Methontology [8], ontoloji geliştirme kapsamında önerilen detaylı süreçlerden bir diğeridir. Ontoloji yeniden kullanımı, kapsamlı bir şekilde değerlendirilmektedir. Ontology Development 101 [9], literatürde yer alan önemli çalışmalardan bir diğeridir. Bu çalışmada vurgulanan en önemli nokta, ontoloji geliştirme sürecinin yinelemeli ve artışlı olma zorunluluğudur. Yeniden kullanım da bir adım olarak belirtilmiştir. DILIGENT [10] yöntemi ise daha çok birden fazla paydaşın aynı anda dahil olabileceği iş birliği bir yaklaşım üzerinde durmaktadır.

Ontoloji geliştirme süreçlerinin birçoğu yeniden kullanımın önemini belirtmekte ve ilgili adımların işletiminde bazı öneriler sunmaktadır. Ancak yeniden kullanım sürecini bütünüyle ele alan bir süreç görülmemiştir.

## 2.3 Ontoloji Yeniden Kullanımı

Ontoloji yeniden kullanımı, geliştirme maliyetinin azalması ve ontoloji kalitesinin artırılmasından çok birlikte çalışabilirliğin sağlanması bakımından önemlidir [1]. Bu nedenle, kullanılacak ontoloji geliştirme metodolojisinin en önemli adımlarından biri yeniden kullanım olmalıdır.

---

<sup>1</sup> <https://bioportal.bioontology.org/>

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

<sup>3</sup> <https://www.ebi.ac.uk/ols/index>

<sup>4</sup> <https://wiki.dbpedia.org/>

<sup>5</sup> <https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>

<sup>6</sup> <http://www.geonames.org/>

En basit ve geleneksel ontoloji yeniden kullanımı, mevcut ontolojinin bütün olarak yeni ontolojiye dahil edilmesi ile sağlanır. Ontolojilerin bütünsel olarak yeniden kullanımı, FOAF<sup>7</sup>, BFO<sup>8</sup> (Basic Formal Ontology) ve GFO<sup>9</sup> (General Formal Ontology) gibi daha küçük ölçekli ve üst seviye ontolojilerde geçerli olabilir. Bütünsel olarak dahil edilen ontoloji, yeni ontolojiyi işleyecek olan kütüphane ve/veya araç tarafından bütünüyle ele alınacak ve yüklenecektir.

Ontolojilerin bütün olarak yeniden kullanılması küçük boyuttaki ontolojiler için uygun olsa da Dbpedia, Yago, SNOMED CT ve FMA gibi çok sayıda kavram ve ilişkiyi barındıran büyük ve kapsamlı ontolojiler için uygulanabilir değildir. Bu ontolojilerdeki kavram ve ilişki sayısı çok fazladır ve tek bir parça halinde işlenebilmesi oldukça zordur (Örneğin; Dbpedia ontolojisinde yaklaşık 5 milyon kavram ve 125 milyon ilişki bulunmaktadır<sup>10</sup>).

Büyük ontolojilerin bir bütün olarak yeniden kullanılması yerine, bu ontolojilerde bulunan ve sadece ihtiyaç duyulan kavramların yeniden kullanımı bir seçenek olarak karşımıza çıkmaktadır. Böyle durumlarda, yeniden kullanılacak kavramların ve bu kavramlar arasındaki ilişkilerin bulunduğu bir ontoloji alt kümesinin oluşturulması ve yeni ontolojiye dahil edilmesi uygun olacaktır. Bu bağlamda, modüler ontoloji geliştirme ve mevcut alt ontoloji çıkarma yaklaşımları, büyük ontolojilerin yeniden kullanım probleminde çözüm getirmeye çalışmaktadır. Ancak ontoloji geliştirme süreçleri ile bütünleştirilebilecek ve araç desteği sağlayan mevcut bir yöntem bulunmamaktadır.

### 3 Ontoloji Yeniden Kullanımı için Bir Yazılım Mimarisi

Ontoloji yeniden kullanımının önemine ve faydalarına rağmen getirdiği ek maliyetler ve zorluklardan da bahsetmek gerekmektedir. İlk problem, yeniden kullanıma dahil edilecek ontolojilerin tespit edilebilmesi ve içlerinde arama yapılabilmesidir. Ardından bu ontolojilerin değerlendirilebilmesi ve yeniden kullanılacak parçalarının uygun bir biçimde yeni ontolojiye aktarılması sağlanmalıdır [1]. Bu zorlukları, etkin bir yöntem ve bu yöntemi destekleyen bir araç ile asgari seviyeye indirebilmek mümkündür.

Yeniden kullanımı destekleyen yöntemler, genellikle büyük ontolojilerin modülerleştirilmesi üzerine odaklanmaktadır. Ontoloji modülerleştirme, ontoloji ayırma ve alt ontoloji çıkarma olmak üzere iki kategoriye ayrılmaktadır [11]. Ontoloji modülerleştirme, ontolojiyi belirli sınırlar dahilinde birden fazla alt bölüme ayırır. Alt ontoloji çıkarma yöntemlerinde ise kullanıcının seçmiş olduğu kavramlara bağlı dinamik bir modül oluşturulur [11]. Alt ontoloji çıkarma yöntemlerinin dinamik yapısı, ontoloji yeniden kullanımı bakımından daha etkin ve avantajlıdır.

Alt ontoloji çıkarma üzerine önerilmiş çok sayıda yöntem bulunmaktadır [12][13][14][15]. Bu yöntemlerin birçoğu kullanıcının seçmiş olduğu kavramları ön

---

<sup>7</sup> <http://xmlns.com/foaf/spec/>

<sup>8</sup> <http://ifomis.uni-saarland.de/bfo/>

<sup>9</sup> <http://www.onto-med.de/ontologies/gfo/>

<sup>10</sup> <http://downloads.dbpedia.org/2016-10/statistics/stats-general-stats.json> (Sadece İngilizce sürümünün istatistikleri dikkate alınmıştır)

tanımlı veya yine kullanıcı tarafından belirlenen kurallar dahilinde işleyerek alt ontolojiyi oluşturmaya çalışmaktadır. Kural tabanlı yöntemlerin en belirginini, Bhatt ve arkadaşları [12] tarafından önerilmiş Materialized Ontology View Extractor (MOVE) isimli çalışmadır. Ranwez ve arkadaşlarının yöntemi [15], kural tabanlı yöntemlerden farklı olarak çizge dolaşımı üzerine yoğunlaşmaktadır. Ancak sadece alt sınıf ve üst sınıf ilişkilerini kullanmaktadır.

Mevcut alt ontoloji çıkarma yöntemlerinin temel sorunu kural tabanlı olmalarıdır. Kural tabanlı yöntemler, seçilen kavramlar arasındaki en anlamlı patikayı bulmaktan çok bu kavramları herhangi bir şekilde bağlamaya çalışmaktadır. İki kavram arasında birden fazla patika olması durumunda hangisinin kullanılacağına yönelik bir hesaplama yapılmamaktadır. Bu durum, kullanıcının seçmiş olduğu kavramlar arasındaki en ilintili patikaların ve ihtiyaç dahilindeki en küçük alt ontolojinin çıkarılmasını engellemektedir.

Mevcut yöntemlerin, yeniden kullanım sürecinin diğer adımları bakımından da eksiklikleri bulunmaktadır. Bu yöntemlerin neredeyse tamamında kullanıcının seçmiş olduğu kavramlar temel alınmaktadır. Ancak hiçbir yöntemde, bu kavramların nasıl seçileceği ve bu adımın nasıl kolaylaştırılacağı ile ilgili bir öneride bulunulmamıştır. Ölçeklenebilirlik sorunu da sadece [12] çalışmasında dikkate alınmıştır. Ancak bu çözüme yönelik bir veri mimarisi önerilmemiştir. Kullanılabilirlik, üzerinde durulmayan diğer bir önemli problemdir. Kullanılabilirlik ile bahsetmek istediğimiz, yöntemin ontoloji geliştirme süreci ile nasıl bütünleştirileceği ve kullanıcılara sunulacak araç desteğidir. Sadece Noy ve Musen [13] tarafından önerilen yöntemde, Protege üzerinde çalışan bir eklenti desteği söz konusudur.

Çalışmamızda, yeniden kullanımı destekleyecek ve yukarıda ifade edilen sorunları çözebilecek bir yöntem önerilmiştir. Bu yöntemin temel işleyişi, veri mimarisi, servis bileşenleri ve eklenti desteği detaylandırılmıştır.

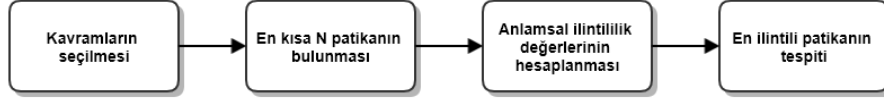
### **3.1 Alt Ontoloji ve Anlamsal İlintililik Tabanlı Yeniden Kullanım Yöntemi**

Ontoloji yeniden kullanımını desteklemesini hedefleyen çalışmamız, kavramların ve ontolojilerin aranmasından, ilgili alt ontolojinin çıkarılmasına kadar olan tüm süreç adımları için çözümler sunmaktadır. Bu nedenle de alt ontoloji çıkarma yönteminin yanı sıra bu yöntemi destekleyecek ve ölçeklenebilirlik problemini çözecek bir veri mimarisi de önerilmiştir.

Öncelikli olarak kural tabanlı yöntemlerin getirdiği eksiklikleri giderebilmek amacıyla anlamsal ilintililik tabanlı bir alt ontoloji çıkarma yöntemi geliştirilmiştir. Bu yöntem, seçilen kavramlar arasındaki en ilintili anlamsal patikaları [16] tespit ederek, seçilen kavramlardan oluşan en küçük ve en ilintili alt ontolojiyi çıkarmak üzerine odaklanmaktadır. Burada bahsedilen anlamsal ilintililik, ilgili ve ilişkili olma durumunu ifade etmektedir [16].

Çalışmamızda öncelikli olarak kavram çiftleri arasındaki en ilintili patikanın bulunması üzerine odaklanılmıştır. Burada kullanılan yöntem, öncelikli olarak ontolojilerin genel amaçlı çizgeler gibi değerlendirilerek kavramlar arasındaki en kısa N patikanın bulunmasıdır. N değeri, ontolojinin büyüklüğü ve kavramlar arasındaki ortalama ilişki sayıları gibi parametrelere göre belirlenmektedir. Bulunan en kısa N patika-

nın anlamsal ilintililik değerleri hesaplandıktan sonra en anlamlı olduğu tespit edilen patika seçilmektedir. Bu süreç Şekil-1 üzerinde gösterilmiştir.



Şekil 1. Kavramlar arasındaki en ilintili patikaların bulunması süreci.

Literatürde, anlamsal ilintililik değerlerinin hesaplanması için farklı yöntemler bulunmaktadır. Ancak OWL (Web Ontology Language) bileşenlerini desteklemesi ve patika tabanlı yaklaşıma daha uygun olması nedeniyle [16] çalışmasında yer verilmiş olan yöntem ve bu yöntemin tanımladığı ilintililik değerleri kullanılmıştır. Bu çalışmada, her bir OWL bileşeni için 0 ile 1 arasında yer alan bir ilintililik değeri verilmiştir. Anlamsal ilintililiğin hesaplanması, genel amaçlı çizgelerden farklı olarak maliyetlerin toplanması şeklinde yapılmamaktadır. Patika üzerinde ilerlendikçe, anlamsal ilintililik değerinin düşmesi gerektiğinden, her bir bileşenin ilintililik değeri çarpılarak ilerlenmektedir [16]. İlintililik hesaplaması, (1) numaralı formülde gösterilmiştir.  $R_p$  tüm patikanın ilintililiğini, her bir  $r$  ise patikada yer alan ilişkilerin ilintililiğini ifade etmektedir.

$$R_p = r_1.r_2.r_3...r_n \quad (1)$$

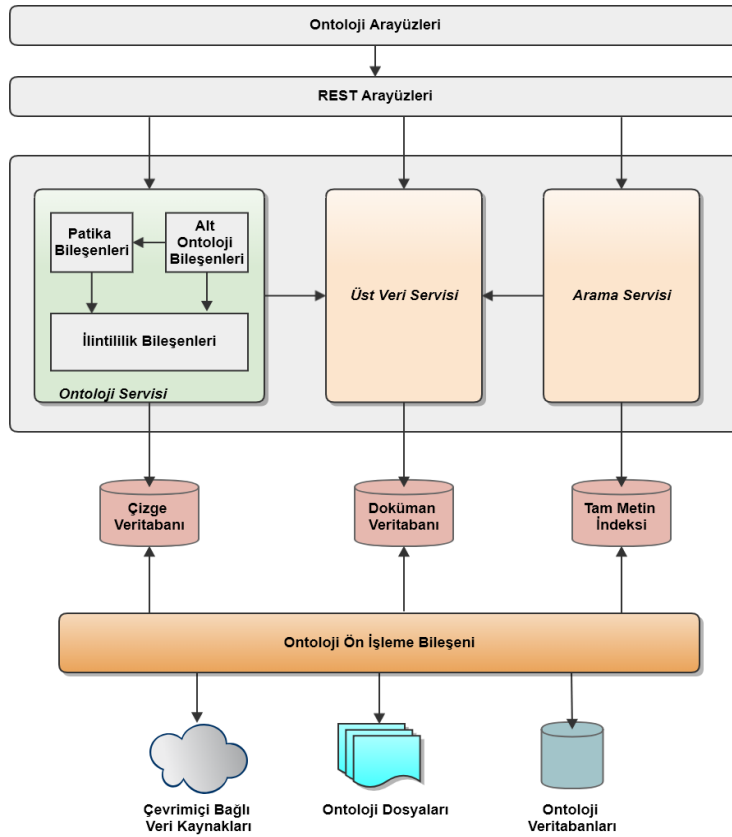
Her bir kavram çifti arasındaki en anlamlı patikanın bulunması, alt ontolojinin oluşturulabilmesi için tek başına yeterli değildir. Yöntemimizdeki amaç, en anlamlı ve en küçük alt ontolojinin çıkarılmasıdır. Bu gereksinim, genel amaçlı çizgelere indirildiğinde Steiner Tree problemi ile karşılaşılmaktadır. Steiner Tree, seçilen düğümlerin en küçük maliyetle birleştirilerek bir alt çizge oluşturulmasını tanımlayan problemdir [17]. Steiner Tree probleminde belirtilen maliyet, anlamsal ilintililik ile değiştirilmiş ve çalışmamıza uyarlanmıştır. Steiner Tree problemi için önerilmiş çok sayıda yöntem bulunmaktadır. Ancak çizge dolaşımına uygun ve patika tabanlı olması nedeniyle, Takahashi ve Matsuyama [18] tarafından önerilen yöntem tercih edilmiştir.

Takahashi ve Matsuyama yöntemi, seçilmiş düğümler arasından birinin rastgele olarak belirlenmesi ile başlar. Bu düğüm, alt çizgeye dahil edilir. Daha sonra kalan düğümler ve alt çizgedeki düğümler arasındaki tüm olası patika çiftlerinin maliyeti hesaplanır. En kısa maliyete sahip olan patika eklenir ve bu patikadaki tüm düğümler kalan düğümlerin listesinden çıkarılır. Bu işlem, kalan düğümler bitene kadar devam eder. Bu maliyet hesaplamaları, (1) numaralı formülde belirtilen ilintililik hesabı ile değiştirilerek yöntemimize uygun hale getirilmiştir. Ayrıca, patika ilintililik hesaplamaları dağıtık hale getirilerek ve önbelleklenerek performansın artırılması sağlanmıştır. Steiner Tree hesaplaması sonucunda elde edilen alt ontoloji, taslak ontoloji olarak kullanıcıya sunulmaktadır.

### 3.2 Ontoloji Yeniden Kullanımını Destekleyen Veri Mimarisi

Anlamsal ilintililik tabanlı alt ontoloji çıkarma yöntemi, ontoloji yeniden kullanımının temelinde yer almaktadır. Ancak bu yöntemin işlerliğini ve ölçeklenebilirliğini sağlamak için uygun veri mimarisinin ve bu mimari üzerinde çalışacak servis bileşenlerinin gerçekleştirimi de kritik öneme sahiptir.

Veri mimarisinin ilk gereksinimi, bulut veri modeline uygun ve ölçeklenebilir olmasıdır. Böylelikle yöntemin yüzbinlerce kavramdan oluşan büyük ölçekli ontolojiler için de performanslı çalışabilmesinin ilk adımı sağlanmış olacaktır. Şekil-2 üzerinde, veri mimarisi ve bu mimari üzerinde çalışan servislerin genel yapısı görülmektedir.



Şekil 2. Yazılım mimarisi.

Veri mimarisi, 3 ayrı veri modelini içermektedir. Çizge veri modeli, yeniden kullanım kapsamında işlenecek ontolojilerin saklanacağı alandır. Doküman veri modelinde, ontolojilere ait üst veriler ve önbellek yer alacaktır. Tam metin indeksi ise ontolojiler içerisinde yapılacak aramalarda kullanılacaktır.

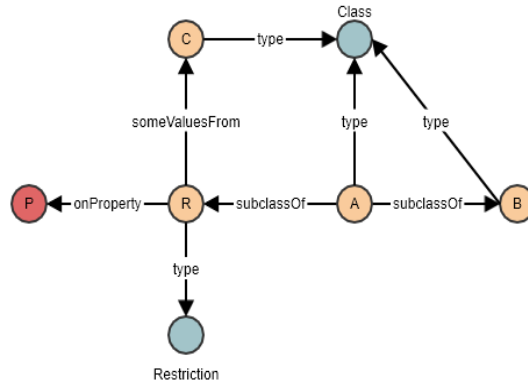
Her bir veri modeli üzerinde çalışan ve mikro servis mimari yaklaşımına göre ayrıştırılmış servisler yer almaktadır. Böylelikle arama ve alt ontoloji çıkarma gibi işlemleri ayrı olarak ölçekleyebilmek mümkün olacaktır. İlerleyen bölümlerde veri mimarisinin alt modelleri ve servis bileşenleri detaylandırılmıştır.

### Çizge Veri Modeli

Ontoloji veri modeli farklı biçimlerde saklanabilir. Özellikle büyük ontolojiler için en uygun çözüm RDF veya ontoloji veritabanlarıdır. Bu veritabanları genellikle çizge veri modeli üzerine kurulmuştur ve SPARQL sorgularının çalıştırılmasını bir uç noktası üzerinden desteklemektedir.

Ontoloji veritabanları SPARQL gereksinimlerini desteklese de ontolojinin genel amaçlı bir çizge gibi dolaşımını sağlama da yetersiz kalmaktadır. Bunun ilk nedeni SPARQL dilinde böyle bir sorgulama bileşeninin olmayışıdır. Bu nedenle, ontoloji veri modelinin genel amaçlı bir çizge veri modeli kullanılarak saklanması yoluna gidilmiştir. Yapılan değerlendirme çalışmaları sonucunda esneklik ve performansından dolayı Neo4j<sup>11</sup> çizge veritabanının kullanımı tercih edilmiştir.

Ontoloji veri modelinin genel amaçlı bir çizge olarak nasıl temsil edilebileceği Şekil-3 üzerinde gösterilmiştir. Kavramlar, nitelikler ve kimliksiz bileşenler birer çizge düğümü olarak aktarılmıştır. Bu bileşenlerin arasındaki tüm ilişkiler ise çizgedeki kenarlar tarafından temsil edilmiştir.



Şekil 3. Ontoloji veri modelinin çizge olarak temsili.

### Tam Metin Arama Modeli

Kullanıcılar, hangi ontolojilerin kendi yeni ontolojilerinde kullanılacağını belirlerken, analiz çalışmalarında elde ettiği anahtar terimleri kullanır. Bu terimlerin hangi ontolojilerde yer aldığını hızlı ve etkin bir şekilde belirleyebilmelidir. Ontolojinin aktarıldığı çizge veri modelinin bu gereksinimi tek başına sağlayabilmesi mümkün

<sup>11</sup> <https://neo4j.com/>



değildir. Bu nedenle tam metin arama yeteneğine sahip ve ölçeklenebilir bir çözüm tercih edilmelidir.

Arama ihtiyaçlarının karşılanabilmesi için bir tam metin arama motoru olan ElasticSearch<sup>12</sup> kullanılmıştır. Ontoloji içerisinde yer alan bileşenler tekil tanımlayıcıları ve etiketleri ile birlikte tam metin arama motorunda indekslenmiştir.

### **Doküman Veri Modeli**

Doküman veri modeli, ontolojilere ait üst verilerin saklanması, yapılandırma bilgilerinin saklanması ve patika ve alt ontolojilere ait önbellek oluşturulması amacıyla kullanılmaktadır.

Ontoloji yapılandırma verileri, sistem içerisinde yer alan her bir ontoloji için çizge ve tam metin veri kaynaklarına ulaşım bilgilerini içerecektir. Böylelikle ölçekleme ihtiyaçları doğrultusunda anlık olarak değişebilecek veri kaynaklarına ulaşım herhangi bir aksama olmadan sağlanabilecektir. Ontoloji üst verileri ise ontolojiler ile ilgili temel istatistikleri (kavram ve nitelik sayıları, yeniden kullandığı ontoloji sayıları vb.) ve ontolojiler arasındaki kullanım ilişkilerini saklayacaktır. En önemli bileşen olan önbellek için de doküman veri modelinden faydalanılacaktır. Kavramlar arasında bulunan ve bir kez işlenen en ilintili patikalar tüm sürecin tekrar etmemesi amacıyla önbellek içerisinde saklanacak ve yöntemin başarımı arttırılacaktır. Doküman veri modelinin gerçekleştirimi için MongoDB<sup>13</sup> veritabanından faydalanılmıştır.

### **3.3 Uygulama Bileşenleri**

Yöntem ve veri mimarisinin ölçeklenebilirliğinin tam anlamıyla sağlanabilmesi için servis mimarisinin de bu yapıya uygun olması gereklidir. Bu nedenle 3 temel servis modellenmiş ve geliştirilmiştir. İlk ve en önemli servis patika ve alt ontoloji süreçlerini işletecek olan ontoloji servsidir. Arama servisi, tam metin destekli arama işlemlerini gerçekleştirecektir. Üst veri servsinden ise ontoloji üst verilerine ve önbelleklere erişim sağlanacaktır.

Servis bileşenlerinin ölçeklenebilir ve bulut mimarisine uygun olması iki önemli işlevsel olmayan gereksinimdir. Bu nedenle mikro servis mimarisine uygun bir yaklaşım izlenmiş ve farklı amaçlı bileşenler farklı servisler olarak geliştirilmiştir. Böylelikle yükün oluştuğu noktalarda daha esnek bir yatay ölçekleme sağlanabilecektir.

Çalışmamızın bir diğer amacı da yöntemin farklı uygulamalar ve araçlar tarafından kolaylıkla bütünleştirilebilir olmasını sağlamaktır. Bu nedenle tüm servisler REST (REpresentational State Transfer) mimarisine uygun olarak HTTP protokolü üzerinden JSON veri biçimi ile hizmet vermektedir.

---

<sup>12</sup> <https://www.elastic.co/>

<sup>13</sup> <https://www.mongodb.com/>

### 3.4 Ontoloji Veri Kaynaklarının Aktarımı

Mevcut ontoloji veri kaynakları, ontolojilerin sorgulanabilmesi ve işlenebilmesi için SPARQL dahil olmak üzere çeşitli uç noktaları sunmaktadır. Bu uç noktaları üzerinde SPARQL sorguları çalıştırılarak ontoloji işlenebilir. Ancak bu ontolojileri, kendi veri modellerine ve kullandıkları teknolojilere bağlı olarak yöntemimiz kapsamında ölçeklenebilir ve yüksek performanslı olarak işleyebilmek mümkün değildir. Bu nedenle kullanılacak kaynakların kendi veri modelimize aktarımı sağlanmalıdır.

Aktarım çalışmaları kapsamında bir araç geliştirilmiştir. Bu araç ontoloji dosyaları üzerinden eriştiği ontoloji kaynaklarını işlemekte, filtrelemekte ve ontoloji ve tam metin arama veri modellerine aktarımını sağlamaktadır. Aktarım aracının geliştirim çalışmaları kapsamında SNOMED CT, FMA, Dbpedia ve Yago ontolojilerinin aktarımı sağlanmıştır.

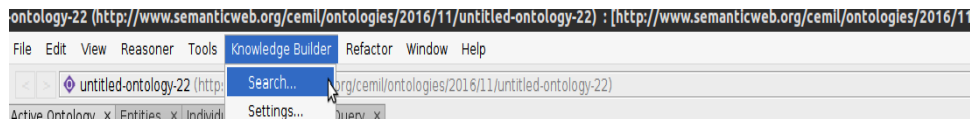
Mevcut durumunda, kullanılacak veri kaynaklarının tarafımızdan aktarımının yapılması gereklidir. Ancak ilerleyen çalışmalarda kullanıcıların kullanmak istedikleri ontolojileri aktarabilecekleri daha etkin bir çözüm sunulması planlanmaktadır.

## 4 Ontoloji Yeniden Kullanım Aracı

Ontoloji yeniden kullanımını destekleyen yöntemimizin servis bileşenleri REST modeline uygun uç noktaları ile dış erişime açılmıştır. Böylelikle, farklı uygulamalar tarafından bütünleştirilebilmesine olanak sağlanmaktadır. Ancak ontoloji mühendislerinin ve kullanıcıların geliştirme süreci boyunca kullanabilecekleri görsel bir araca da ihtiyaçları vardır. Bu durum, yöntemimizin bir araç ile desteklenmesi gereksinimini doğurmaktadır.

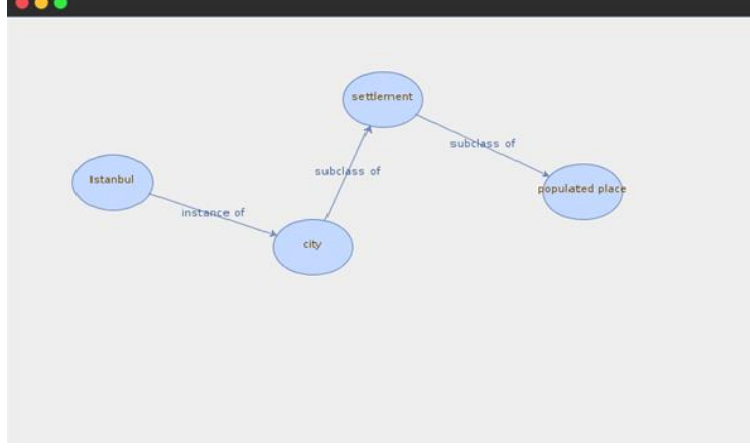
Geliştirilecek aracın tüm ontoloji geliştirme süreçleri ile bütünleşik olması kullanılabilirlik bakımından önemlidir. Bu nedenle mevcut bir ontoloji geliştirme editörüne yeniden kullanım süreçlerini dahil eden bir eklenti geliştirilmesi daha uygun olacaktır. Editör olarak, ontoloji geliştirme çalışmaları için standart haline gelmiş Protege tercih edilmiştir. Protege, esnek ve uyarlanabilir bir eklenti mimarisi sunmaktadır [19].

Geliştirilen eklentimizin yüklenmesi ile birlikte Protege üst menüsünde “*Knowledge Builder*” isimli menü ve bu menünün altında yer alan “*Search*” ve “*Settings*” alt menüleri oluşmaktadır (Bkz. Şekil-4).



Şekil 4. Yeniden kullanım eklentisi.

*Settings* menü öğesi, yeniden kullanım servislerine ulaşım adresleri ve kullanılması istenen ontoloji kaynakları ile ilgili yapılandırma adımlarını içermektedir. *Search* alt menü öğesi, ontoloji yeniden kullanım sürecini başlatmaktadır. Açılan kullanıcı arayüzü üzerinden ontolojilerde aramalar yapılmakta ve seçilen kavramlar görsel gezinim arayüzüne aktarılmaktadır. Gezinim arayüzü Şekil-5 üzerinde görülmektedir.



Şekil 5. Gezinim arayüzü.

Geliştirdiğimiz eklenti, servis bileşenlerinin sunduğu REST uç noktaları ile yöntemimizle iletişim sağlamaktadır. Ontolojileri sorgulayabilmekte, kavramlar arasındaki ilişkileri örüp alt ontoloji çıkararak Protege üzerinde tanımlanmış olan ontolojiye dahil edebilmektedir.

## 5 Sonuçlar ve Gelecek Çalışmalar

Bu çalışmada, ontoloji yeniden kullanımının önemi ve zorlukları üzerinde durulmuş ve ontoloji yeniden kullanımını kolaylaştıracak bir yöntem ve yazılım mimarisi önerilmiştir. Ayrıca geliştirilen yeniden kullanım aracı, bir Protege eklentisi olarak sunulmuş ve kullanılabilirliğinin artırılması amaçlanmıştır.

Geliştirilen alt ontoloji çıkarma yönteminin anlamsal ilintililik tabanlı oluşu, kural tabanlı mevcut yöntemlerin eksiklerini gidermeyi amaçlamaktadır. Ayrıca mevcut yöntemlerin, ontoloji yeniden kullanım sürecindeki tüm adımları destekleyecek bir araç sunmamış olması, çalışmamızı önemli kılan bir diğer faktördür. Ölçeklenebilirlik, ontoloji yeniden kullanımının önündeki önemli bir engeldir. Yüzbinlerce kavram ve milyonlarca ilişkiden oluşan ontolojilerin işlenip yeni ontolojiye dahil edilebilmesi, bu alandaki yöntemleri kısıtlamaktadır. Yöntemin ölçeklenebilirliğinin ilk kriteri ise ölçeklenebilir veri mimarisinin kullanılmasıdır. Çalışmamızda, üzerinde önemle durulan noktalardan bir diğeri de ölçeklenebilir bir veri mimarisi geliştirmek olmuştur.

İlerleyen dönemlerde, aracın geliştirilmesine yönelik bazı çalışmaların yapılması planlanmaktadır. Mevcut durumda, ontolojilerin tarafımızca geliştirilen veri modeline aktarımı gereklidir. Bu durum kullanıcıları sisteme aktarılmamış ontolojileri kullanabilmesi açısından bir eksikliktir. Bu sürecin kullanıcıların taleplerine bağlı ve otomatik bir hale getirilmesi amaçlanan öncelikli çalışmadır.

## Kaynakça

1. Simperl, E.: Reusing ontologies on the Semantic Web: A feasibility study. *Data & Knowledge Engineering*. 68, 905–925 (2009).
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American*. 284, 28–37 (2001).
3. Ding, L., Finin, T., Joshi, A., Pan, R., Scott Cost, R., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: A Search and Metadata Engine for the Semantic Web. In: 13th ACM Conference on Information and Knowledge Management, Washington D.C. (2004).
4. M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, E. Motta, WATSON: a gateway for the semantic web, In: Proceedings of the Fourth European Semantic Web Conference (ESWC), Poster Session, Austria, 2007.
5. D'Aquin, M., Noy, N.F.: Where to publish and find ontologies? A survey of ontology libraries. *Journal of Web Semantics*. 11, 96–111 (2012).
6. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95. (1995).
7. Gruninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. In: IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing. , Montreal (1995).
8. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: Proc. Symposium on Ontological Engineering of AAAI (1997).
9. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory. 25 (2001).
10. Pinto, H.S., Tempich, C., Staab, S.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolInG Engineering of oNTologies. In: de Mantaras, R.L. and Saitta, L. (eds.) Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), pp. 393-397. IOS Press, Valencia, Spain (2004).
11. Doran, P.: Ontology Modularization: Principles and Practice, (2009).
12. Bhatt, M., Flahive, A., Wouters, C., Rahayu, J.W., Taniar, D., Dillon, T.S.: A Distributed Approach to Sub-Ontology Extraction. In: AINA (1). pp. 636-641. IEEE Computer Society (2004).
13. Noy, N.F., Musen, M.A.: Specifying Ontology Views by Traversal. In: International Semantic Web Conference. pp. 713-725. Springer Berlin (2004).
14. Seidenberg, J., Rector, A.: Web ontology segmentation: analysis, classification and use. In: WWW '06: Proceedings of the 15th international conference on World Wide Web. pp. 13-22. ACM Press, Edinburgh, Scotland (2006).
15. Ranwez, V., Ranwez, S., Janaqi, S.: Subontology extraction using hyponym and hypernym closure on is-a directed acyclic graphs. *IEEE Transactions on Knowledge and Data Engineering*. 24, 2288-2300 (2012).
16. Giray, G., Ünalir, M.O.: A method for ontology-based semantic relatedness measurement. *Turkish Journal of Electrical Engineering and Computer Sciences*. 21, 420–438 (2013).
17. Gilbert, E.N., Pollak, H.O.: Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*. 16, 1–29 (1968).
18. Takahashi, H., Matsuyama, a: An approximate solution for the Steiner problem in graphs. *Math Japonica*. 24, 573–577 (1980).
19. Anatomy of a Plugin, <https://protegewiki.stanford.edu/wiki/PluginAnatomy>, (Son Erişim: 17.06.2018).