Ontoloji Tabanlı Bilgi Sistemlerinde Politika Yönetimi

Özgü CAN, Murat Osman ÜNALIR

Bilgisayar Mühendisliği Bölümü, Ege Üniversitesi, Bornova-İzmir, Türkiye
ozgu.can@ege.edu.tr , murat.osman.unalir@ege.edu.tr

Özet— Bir bilgi uzayı olan web, sadece kişilerin haberleşmesi için değil, aynı zamanda bilginin makinelerce de anlaşılabileceği Anlamsal Web yapısına doğru gelişmektedir. Anlamsal Web verinin paylaşılmasına ve tekrar kullanımına izin veren genel bir çatı sağlamaktadır. Ancak, bu paylaşımın ve tekrar kullanımın beraberinde getirdiği tehditler nedeni ile güvenlik ve gizlilik kavramları önem kazanmıştır. Böylelikle, erişim politikalarının geliştirilmesi ihtiyacı ortaya çıkmıştır. Bu çalışmada Anlamsal Web politikalarının bilgiye güvenli bir biçimde erişim için nasıl kullanılacağına değinilmektedir. Bu amaçla, bilgiye erişimin ontoloji tabanlı bir yaklaşım ile gerçekleştirildiği Ontoloji Tabanlı Erişim Denetimi modeli geliştirilmiştir. Bu modelde, ontolojiler kullanılarak çeşitli politikaların yaratılması gerçekleştirilmektedir. Böylelikle, bu model, ontoloji tabanlı bilgi sistemlerinde erişim denetiminin sağlanması amacı ile kullanılabilecektir. Modelin uygulaması Jena Anlamsal Web çatısı kullanılarak gerçekleştirilmiştir.

Anahtar kelimeler— Anlamsal Web, ontoloji, politika, erişim denetimi

Policy Management in Ontology Based Information Systems

Abstract— The web as an information space improves towards Semantic Web, not only for the communication of people, but also for the information to be understood by machines. Semantic Web provides a common framework that allows data to be shared and reused. However, security and privacy concepts have become an important issue because of threats which the sharing and reusing brings with. Thus, the needs to develop access policies have emerged. In this work, how Semantic Web policies will be used for secure access to information is discussed. For this purpose, Ontology Based Access Control model is developed to carry out access to information with ontology based approach. In this model, ontologies are used to create various policies. Thus, this model could be used for the purpose of providing access control in ontology based information systems. Jena Semantic Web framework is used to implement the application of the model.

Keywords— Semantic Web, ontology, policy, access control

1. GİRİŞ

World Wide Web bilginin kolaylıkla paylaşılmasını sağlamasına rağmen bilginin ve kaynakların korunması için çok az seçenek sunmaktadır. World Wide Web'in geliştirilmesi ile, makineler tarafından da anlaşılabilecek web sayfalarına ve bilginin bütünleştirilmesi için ontolojilerin kullanımına duyulan ihtiyaç sonucunda ortaya çıkan Anlamsal Web teknolojilerinin de güvenliğini sağlayacak etkili mekanizmalara ihtiyaç duyulmaktadır. Bunun neticesinde, kaynak paylaşımının kontrol edilebilmesi için erişim denetim politikalarına dayanan mekanizmalara gereksinim duyulmaktadır.

Anlamsal Web'de politika yönetimi bir kaynağa erişim için kuralların tanımlanması, kullanıcıların bu kuralları yorumlaması ve kurallara uyması için kullanılmaktadır. Anlamsal olarak zengin bir biçimde tanımlanmış olan politikalar, insan hatalarını ve politika çelişkilerini azaltmakta, politika analizini ve birlikte işlerliği kolaylaştırmaktadır [1].

Anlamsal Web teknolojilerine dayanan politika dilleri, politikaların heterojen etki alanı verileri üzerinde tanımlanmasına izin vermekte ve aynı bilgi modelini kullanmayan katılımcılar arasında ortak anlamı desteklemektedir [2]. Son yıllarda yapılan erişim denetim çalışmalarında iki paralel konu ele alınmaktadır [2]: gerçek dünya uygulama etki alanlarının politika ihtiyaçlarını karşılamaya yönelik erişim denetim modellerinin geliştirilmesi ve erişim denetimi için politika dillerinin geliştirilmesi. Bu iki paralel konunun, erişim denetimi ve politika dilleri, güvenlik altyapısının gelişimini sağlamak için görevdeşlik yaratması gerektiği düşünülmektedir.

Anlamsal Web uygulamalarında, bilginin temsilinde ontolojiler kullanılmaktadır. Thomas R. Gruber tarafından yapılmış olan, daha sonra Rudi Studer tarafından yenilenmiş olan tanımda, ontoloji, kavramsallaştırmanın açık ve biçimsel bir belirtimidir [3]. Ontolojiler nesneler, kavramlar ve ilişkiler tanımlayarak belirli bir etki alanına ilişkin bilginin modellenmesini sağlamaktadırlar. Böylece sistemler arasında verilerin değiş tokuşu için standart kavramsal söz varlıkları belirtilmekte, bilgi yeniden kullanılabilmekte, sorgulamaların yanıtlanması için servisler sağlanmakta ve çok çeşitli sistemler arasında birlikte işlerliği kolaylaştıran servisler gerçekleştirilmektedir.

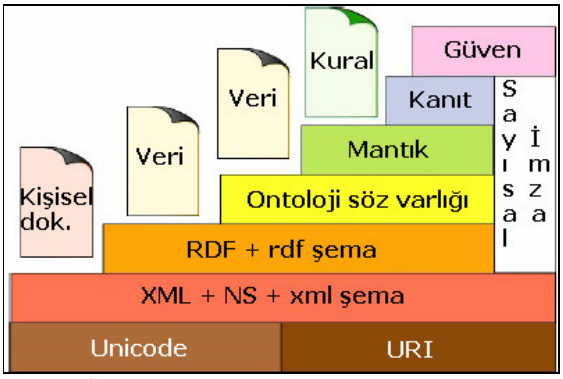
KaoS [4], Rei [5] ve Ponder [6] kaynakçada en sık rastlanan Anlamsal Web politika dilleridir. KAoS politika dilinde ontolojiler OWL dili ile tanımlanmaktadır. Web servisleri için politika ve etki alanı yönetimi servislerinden oluşmaktadır. KAoS politika ontolojisi, yetkiler (eyleme izin veren ya da yasaklayan kısıtlar) ve zorunluluklardan (bir durum meydana geldiğinde bazı eylemleri gerektiren ya da bu gereksinimden vazgeçilmesini belirten kısıtlar) oluşmaktadır [7]. KaoS grafiksel arayüz olarak KaoS Politika Yönetim Aracını (KaoS Policy Administration Tool - KPAT) sağlamaktadır. KPAT kullanıcılara politika tanımlamasında, düzeltme ve uygulamada yardımcı olmaktadır. Rei, OWL Lite temelli bir politika tanımlama dilidir. Kullanıcıların yetkiler, yasaklar, zorunluluklar ve özel izinler kavramlarını tanımlamasına izin vermektedir [7, 8]. Sistemdeki yetkiler ve zorunlulukların varlıklar arasında değiş tokuş edilebilmesi için Rei politika dilinin konuşma edimleri (speech acts) kümesi vardır. Ponder, bildirim deyimlerinden oluşan nesneye dayalı bir politika dilidir. Ponder politika yönetimi için yöntemleri desteklemektedir. Politikaları hazırlamak, güncelleştirmek, silmek ve taramak için çeşitli grafiksel araçlar sağlamaktadır. Ponder'da dört politika türü bulunmaktadır. Bunlar; izinler, zorunluluklar, sakınımlar ve yetki aktarımlarıdır.

Bu çalışmada ontoloji tabanlı bilgi sistemlerinde politika yönetiminin gerçekleştirilmesi amacı ile geliştirilen bir erişim denetimi modeli uygulaması anlatılmaktadır. İkinci kısımda n-katmanlı mimari ve Anlamsal Web hakkında bilgi verilmektedir. Üçüncü kısım ontoloji tabanlı politika yönetimi için geliştirilen modelin yapısını ve politika ontolojisi tanımlama adımlarını anlatmaktadır. Ontoloji Tabanlı Erişim Denetimi modelinin mimarisi ve geliştirilen uygulama yazılımı dördüncü kısımda incelenmektedir. Son olarak sonuçlar sunulmaktadır.

2. N KATMANLI MİMARİ ve ANLAMSAL WEB

World Wide Web insanların birbiri ile olan iletişim şeklinin ve iş yaşamının yönünün değişmesine neden olmuştur. Böylelikle, gelişen dünyanın bilgi toplumuna dönüşmesini sağlamıştır. Bugünkü webin içeriği insanların kullanımı için uygundur. Kişiler bilgi arayabilir, diğer kişiler ile iletişime geçebilir ve çevrimiçi alışveriş yerlerini gezip buralardan alışveriş yapabilirler. Günümüz webinin temel problemi web içeriğinin makine erişebilir (machine accessible) olmaması nedeni ile anlamsallıktan yoksun olmasıdır. Web kullanıcılarına daha iyi destek verebilmek için, web bilgisi makinelerce anlaşılabilir (machine understandable) olmalıdır. Bu da günümüz webinin Anlamsal Web olarak değiştirilmesidir. Burada önemli olan Anlamsal Web'in günümüz webine paralel bir şekilde gelişen yeni bir küresel bilgi paylaşımı ortamı olmak yerine, aşamalı olarak, varolan webden geliştirilecek olmasıdır [9].

Şekil 1'de [9] Tim Berners-Lee tarafından önerilen Anlamsal Web katmanlı yapısı görülmektedir. En alt katmanda yer alan XML (eXtensible Markup Language), kullanıcı tarafından tanımlanmış söz varlığı kullanılarak yapısal web belgeleri yazılmasını sağlayan bir dildir. RDF, web kaynakları ile ilgili olarak yalın ifadeler yazılan varlık-ilişki modeline benzer yalın bir veri modelidir. RDF, Anlamsal Web'in veri modelidir. Zorunlu olmamakla birlikte sıklıkla XML ile ifade edilmektedir. Bu nedenle XML katmanının üstünde yer almaktadır. RDF Şeması, web nesnelerini sıradüzen içerisine düzenleyen modelleme ilkelleri sağlamaktadır. Sınıflar ve özellikler, alt sınıf ve alt özellik ilişkileri, etki alanı ve erim kısıtlamaları temel ilkellerdir. RDF Şeması, RDF tabanlıdır ve ontoloji yazımı için bir ilkel dil olarak görülebilir. Ancak, RDF Şemasını genişleten ve web nesneleri arasında karmaşık ilişkilerin tanımlanmasına izin verecek daha güçlü ontoloji dillerine gereksinim vardır. Mantık katmanı, ontoloji dilini güçlendirmek ve uygulamaya özel bildirim deyimi bilgisinin yazımına izin vermek için kullanılmaktadır. Kanıt katmanı, hem tümdengelimli işlemleri hem de web dillerinde kanıtların temsil edilmesini ve kanıt onaylanmasını içermektedir. Son olarak Güven katmanı, sayısal imzaların ve güvenilir etmenler, oranlar, sertifika kuruluşları ve tüketicilerin önerilerini temel alan diğer bilgi türlerinin kullanımı ile ortaya çıkmaktadır. Güven, Anlamsal Web katmanlı yapısının en üstünde yer alması nedeniyle çok önemli bir kavramdır. Webin kullanılabilirliği, kullanıcılar işlemlerinde güvencede olduklarında ve sağlanan bilginin niteliğine bağlı olarak elde edilecektir [9].



Şekil 1. Anlamsal Web katmanlı yapısı

Mevcut web uygulamaları gibi Anlamsal Web uygulamaları da aynı zamanda n-katmanlı uygulamalardır. N-katmanlı mimari, uygulama

geliştiricilerinin esnek ve yeniden kullanılabilir bir uygulama için model yaratmasını sağlamaktadır. Uygulama geliştiriciler, uygulamayı baştan yazmak yerine, uygulamayı katmanlara ayırarak yeni bir katman ekleyebilir ya da mevcut katmanı güncelleyebilirler. 3-katmanlı mimaride sunum (presentation), iş (business) ve veri (data) katmanları yer almaktadır. Sunum katmanı dış varlıklar ile iletişimden sorumludur. Anlamsal Web'de dış varlıklar sadece kullanıcıları değil kullanıcıları temsil eden etmenleri de içermektedir. İş katmanı, sonuçların sunum katmanına iletilmeden önce verilerin işlenmesinden sorumludur. Veri katmanı ise bilginin ontoloji deposundan getirilmesi ve ontoloji deposuna yazılmasından sorumludur.

3. ONTOLOJİ TABANLI POLİTİKA YÖNETİMİ

Anlamsal Web bağlamında odaklanılan çalışmalardan biri de güvenlik ihtiyaçlarının nasıl tanımlanacağıdır. Bu amaçla, oluşturulacak güvenlik politikalarının tanımlanmasında ihtiyaç duyulan nesnelerin, erişim türlerinin ve konuşma edimlerinin modellenmesi için ontolojilerden yararlanılmaktadır.

Bu çalışmada, Anlamsal Web teknolojileri kullanılarak temel politika kavramlarını içeren bir Ontoloji Tabanlı Erişim Denetim Modeli (Ontology Based Access Control - OBAC) ortaya konulmaktadır [10]. Bu modelde bilgiye erişim ve politika yönetimi anlamsal tabanlı bir yaklaşım ile gerçekleştirilmektedir. OBAC'te; erişilecek nesne, bu nesne üzerinde gerçekleştirilebilecek eylemler ve bu eylemlerin hangi koşullar altında gerçekleştirilebileceğini belirten kısıtlar ontolojik olarak tanımlanmaktadır. Böylelikle, anlamsal olarak bütünlüğü sağlanması gereken ontoloji tabanlı erişim denetimi ve politika yönetimi sağlanmaktadır. Günümüzde, ontoloji tabanlı bilgi sistemleri konusunda gerçekleştirilen çalışmalar yeni olduğundan ontoloji tabanlı bilgi sistemlerinde politika yönetiminin sağlanması konusu, bu alanda yapılan çalışmaların başarısını arttıracak olan erişim denetiminin sağlanmasına yönelik önemli bir katkı sağlamaktadır.

İzleyen kısımda, OBAC modelinin yapısını incelemeden önce, politika yönetiminde kullanılan temel politika kavramlarına değinilmektedir.

3.1. Politika ve Politika Kavramları

Politika, sistemin davranış şeklini belirten bir durumdur. Bir servisi, kimin ve hangi koşullar altında kullanabileceğini, bilginin servise nasıl sağlanacağını ve sağlanan bilginin nasıl kullanılacağını belirtir [11]. Bir karar verme süreci olan politika, hem bir karar destek sistemi hem de bildirimsel deyim davranış sistemi olarak davranmaktadır [12].

Politika, politika nesnelerinden oluşan politika kurallarından oluşmaktadır. Politika nesnelerine deontik nesneler de denilmektedir. Kaynakçada dört politika nesnesi bulunmaktadır. Bunlar; izin (permission), yasak (prohibition), zorunluluk (obligation) ve özel izin (dispensation) tanımlamalarıdır. İzin, veriye ya da servise erişenin neleri yapabileceğini; yasak, neleri yapamayacağını; zorunluluk, neleri yapması gerektiğini; özel izin ise neleri yapmasına gerek kalmadığını belirten durumdur.

Politikalar ile ilgili bir başka kavram da konuşma edimleridir (speech acts). Konuşma edimi, politikaların daha az ayrıntılı olmasını ve merkezi olmayan güvenlik denetimini sağlar. Politikaların devingen olarak değiştirilmesine olanak verir. Konuşma edimlerinde gönderici (sender), alıcı (receiver), içerik (content) ve koşul (condition) olmak üzere dört özellik bulunmaktadır. Konuşma edimlerinin geçerli olabilmesi için göndericinin uygun izinleri olmalıdır. Dört konuşma edimi vardır. Bunlar: yetki aktarımı (delegate), istek (request), yetkinin geri alımı (revocation) ve iptal (cancel)'dir. Yetki aktarımında, gönderici alıcı için izin ekler. İstekte ise gönderici bir eylem ya da yetki için istekte bulunur. Yetkinin geri alımı durumunda gönderici bir izini siler ya da bir yasak ekler. İptal konuşma ediminde ise gönderici isteği etkisiz kılar.

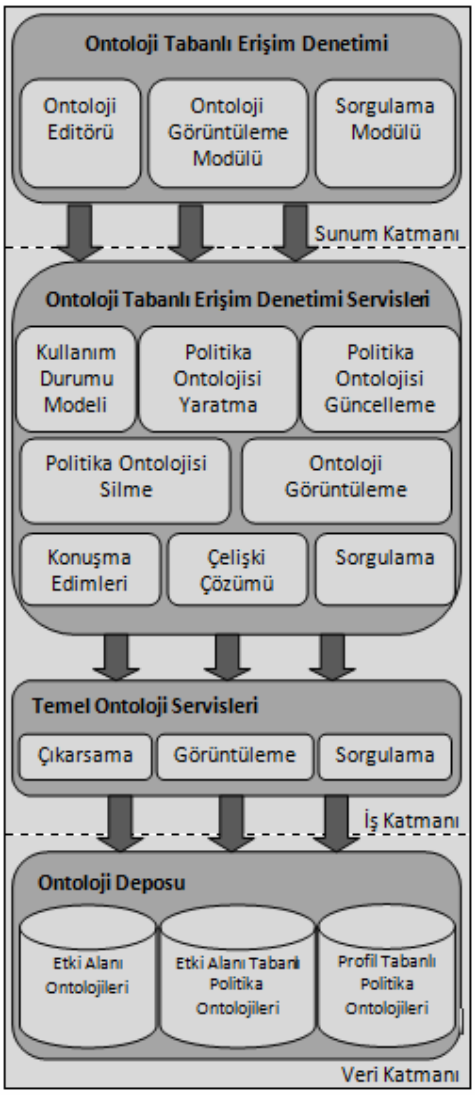
Çelişki çözümü, politika tanımlamalarında karşılaşılan bir diğer kavramdır. Aynı hedefteki aynı işlem için farklı kurallar/politikalar tanımlanmış olabilir. Böyle bir durumda çelişki meydana gelecektir. Birden fazla kuralın var olması durumunda hangi kuralın uygulanacağının ya da uygulanmayacağının belirlenmesi işlemi çelişki çözümüdür. Amaç çatışmanın sonlandırılmasıdır. Çözüm için öncelik belirlemesinden (specifying priority) ve öncelik ilişkilerinden (precedence relations) yararlanılmaktadır.

Politikalar yaratıldıktan sonra sistemde bu politikaların yürütülmesi işlemi için politika motoru kullanılmaktadır. Politika motoru politikaları yorumlar ve çıkarsama yapar. Konuşma edimleri ve etki alanı bilgisini kullanarak uygulanabilir. Yetkilerin, yasakların, zorunlulukların ve özel izinlerin kararlarını verir. OBAC modeli geliştirilirken temel alınan politika dili açık kaynak olarak bulunmasından dolayı Rei [5]'dir.

3.2. OBAC'in Yapısı

Kaynakçada yer alan modellerde erişilecek kaynak ve bu kaynağa erişecek varlık ile ilgili olarak herhangi bir üstveri bilgisi bulunmamaktadır. OBAC modelinde ise erişilecek kaynak ve bu kaynağa erişecek varlık ile ilgili üstveriler oluşturulmakta, politikalar da bu üstveriler temel alınarak yaratılmaktadır. Politikaların bir parçası olan üçlüler modelde; kaynağa erişecek varlık özne, kaynağın kendisi nesne ve politika nesneleri yüklem olarak temsil edilmektedir. OBAC modeli erişim denetimi düzeneğindeki tüm yapıtaşlarının (kaynak, özne, eylem) anlamsal olarak temsil edilebilmesini sağlamaktadır.

Şekil 2'de, Anlamsal Web tabanlı olarak geliştirilen Ontoloji Tabanlı Erişim Denetimi modeli 3-katmanlı mimari için uyarlanmıştır.



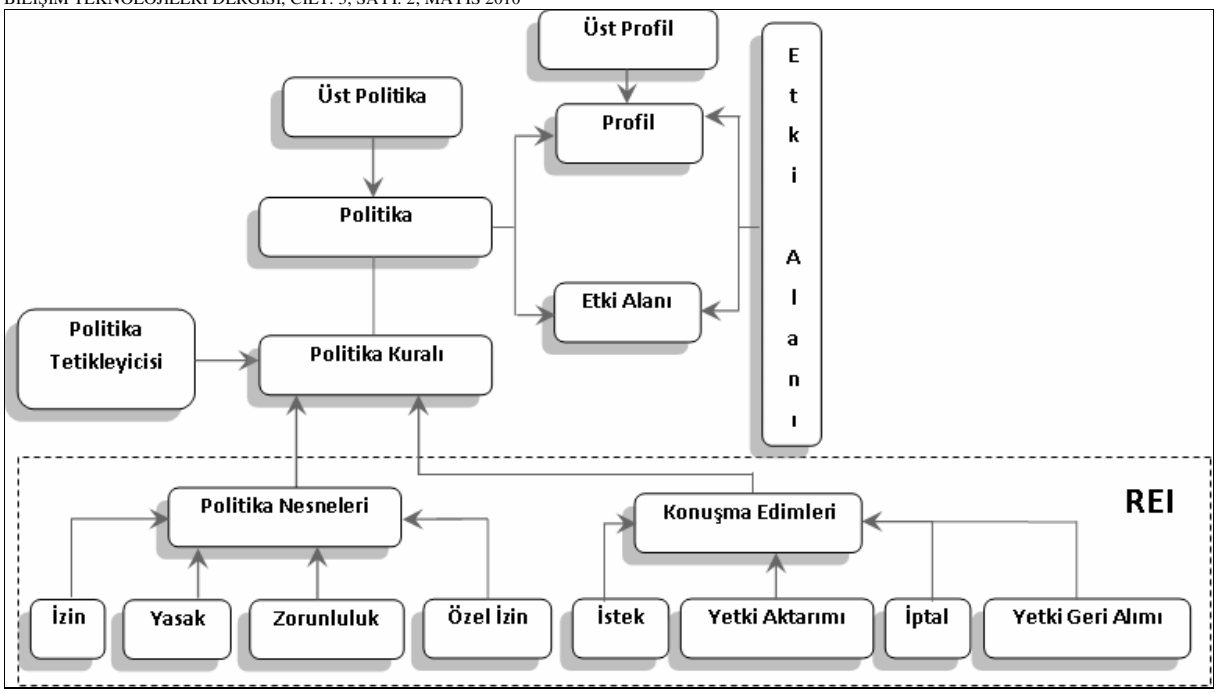
Şekil 2. Ontoloji Tabanlı Erişim Denetimi Modeli için 3-katmanlı mimari

Bu çalışma iş katmanı üzerine odaklanmaktadır. Ontoloji deposundan gelen bilginin uygulamanın ihtiyaçları için ne kadar uygun olduğunun belirlenmesi ve düzenlenmesi işlemleri bu katmanda gerçekleştirilmektedir. Bu çalışmada, iş katmanı, ontoloji dilini işleterek çıkarsama, görüntüleme ve sorgulama yapan bir kod şeklinde genişletilmektedir. Bu amaçla Jena Anlamsal Web çatısı kullanılmaktadır [13]. HP Laboratuarları tarafından geliştirilmiş olan Jena, Anlamsal Web uygulamalarının geliştirildiği bir Java çatısıdır. Anlamsal Web bilgi modeli ve dillerini kullanan uygulamaların kolay bir şekilde geliştirilmesini sağlamaktadır. RDF, RDFS, OWL ve SPARQL için programlama ortamını sağlamakta ve kural tabanlı bir çıkarsama motorunu içermektedir. Jena açık kaynak kodu Java kütüphanesine aktarılarak kullanılmaktadır. Jena'nın en önemli avantajı, Anlamsal Web uygulama geliştiricisine kullanılan ontoloji dilinden bağımsız olarak tutarlı bir programlama dili arayüzü sağlamasıdır. Bu çalışmada yer alan sorgulama işlemleri için SPARQL sorgu dili kullanılmaktadır [14].

OBAC modelinde geliştirilen politika ontolojileri iki türlüdür. Bu ontolojilerden biri, temel alınan etki alanı kurallarına yönelik olarak geliştirilen etki alanı tabanlı politika ontolojisi diğeri ise, kullanıcıya daha iyi bir kişiselleştirme sağlayabilmesi için etki alanında yer alan profillere yönelik olarak geliştirilen profil tabanlı politika ontolojisidir.

Kişiselleştirme kişiselleştirilmiş bilgi sistemlerinde bir gereksinimdir [15] ve ayrıca kullanıcı beklentilerini karşılamaktadır [16]. Bu nedenle, kişiselleştirme, ilgisiz bilgi süzülerek ve kullanıcının benzer ilgileri ile ilgili ek bilgi tanımlanarak sağlanabilir [15]. Kişiselleştirilmiş bir bilgi sisteminde kullanıcılar seçimlerini tanımlamakta ve gereksinimleri doğrultusunda etki alanını sorgulamaktadır. Kullanıcı bağlamını ve kişiselleştirilmiş bilgi sistemini kaydetmek için kullanıcı profilleme kullanılmaktadır [17]. Bu amaçla, OBAC modelinde oluşturulan profil tabanlı politikalar üst profil ontolojisini temel alan profil ontolojisini kullanmaktadır [10, 18, 19].

OBAC modelinde tanımlanan politika ontolojileri ve politika kavramları arasındaki ilişkiler Şekil 3'de yer almaktadır. Sistemin daha iyi bir kişiselleştirme için kullanıcı gereksinimlerine yönelik olarak davranabilmesi amacıyla etki alanı ve profil politikaları olmak üzere iki çeşit politika tanımlanmaktadır. Etki alanı verisi; etki alanı ve profil tabanlı politika ontolojilerini oluşturmak için kullanılmaktadır. Politika ontolojileri üst politika olarak aldığımız Rei politika dili [5] temel alınarak yaratılmaktadır. Ayrıca, profil tabanlı politikalar üst profil ontolojisi temel alınarak yaratılmaktadır [10, 18, 19]. Politika, politika nesnelerinden oluşan politika kurallarından oluşmaktadır. OBAC modelinde politika nesneleri olarak izin, yasak, zorunluluk ve özel izin tanımlanmaktadır. Modelde yer alan politika tetikleyicisi politika kurallarının yürütülmesini sağlamaktadır. Konuşma edimleri olarak da istek, yetki aktarımı, iptal ve yetki geri alımı tanımlanmaktadır.



Şekil 3. OBAC politika ontolojileri ve politika kavramları arasındaki ilişki

3.3. REI Terim Bilimi ve Ontolojileri

OBAC modelinde politika ontolojilerinin tanımlanmasında Rei ontolojileri kullanılmaktadır. Rei politika dilinde kullanılan terim bilimi Çizelge 1'de gösterilmektedir.

Her bir Rei ontolojisi etki alanı ile ilgili olan sınıfları (class) ve özellikleri (property) tanımlamaktadır. Rei politika dili aşağıdaki ontolojilerden oluşmaktadır:

- *ReiPolicy:* Politika etki alanı içerisindeki varlıkların davranışlarını belirler. Kuralları ve politika etki alanını tanımlayan bağlamı içerir.
- *ReiMetaPolicy:* Üst politikalar, politikaların nasıl yorumlandığı (interpreted) ve çelişkilerin nasıl çözümlendiği ile ilgili politikalardır. Rei, iki üst politika türünü modellemektedir: (i) varsayılanlar (ii) politikaların farklı gereksinimlerini karşılamaya yönelik olan çelişkilerin çözümlenmesi. Varsayılanlara yönelik olan üst politikalar *Behavior* ve *MetaMetaPolicy* sınıflarını, çelişkilerin çözümüne yönelik olan üst politikalarda *Priority* ve *ModalityPrecedence* sınıflarını içermektedir.
- *ReiEntity:* Herhangi bir kullanıcı, yazılım etmeni veya donanım kaynağı *entity:Entity* ile tanımlanmaktadır.
- *ReiDeontic:* Politika etki alanındaki varlıklar üzerinde izinler, yasaklar, zorunluluklar ve özel izinler yaratılması için kullanılmaktadır. Burada; hangi aktör ya da aktörler kümesinin hangi eylem ya da eylemler kümesini hangi kısıtlar altında gerçekleştireceği tanımlanmaktadır.

Çizelge 1. Rei Terimleri

Rei Terimi	**Tanım**
politika *(policy)*	Etki alanı içerisindeki varlıkların davranışlarını tanımlayan kurallar kümesidir.
politika etki alanı *(policy domain)*	Politika tarafından etkilenen varlıklar ve kaynaklar kümesidir.
deontik nesne *(deontic object)*	Deontik nesneler: İzin (*permission*), yasak (*prohibition*), zorunluluk (*obligation*) ve özel izindir (*dispensation*).
konuşma edimi *(speech act)*	Bazı eylemleri yerine getirmek için dilin kullanımıdır.
üst politika *(meta policy)*	Bir politikanın nasıl uygulanacağını belirten bir politikadır. Rei politika dilinde çelişkilerin çözümlenmesi için kullanılan düzenekleri belirtir.
kısıt *(constraint)*	Belirli bir politika, kural ya da deontik nesne ile ilgili bir koşuldur.

- *ReiConstraint:* Koşul, nesneler kümesini ve eylemler kümesini tanımlamak için kullanılmaktadır. İki çeşit koşul tanımlanmaktadır: *SimpleConstraint* ve *BooleanConstraint*. *SimpleConstraint*, RDF deyimleri gibi üçlü olarak biçimlenmektedir: özne (subject), yüklem (predicate) ve nesne (object). *BooleanConstraint*'de ise hem Simple hem Boolean koşulları *And*, *Or* ve *Not* boole işleci kullanılarak birleştirilebilir.
- *ReiAnalysis:* Tutarlı ve geçerli politikalar geliştirebilmek için kullanılan bu ontolojide Rei iki belirtim sağlamaktadır: *UseCase* yönetimi ve *WhatIf* çözümlemesi. *UseCase* yönetiminde, diğer politikalar kümesine karşı bir grup *UseCase*'in doğruluğu daima sağlanmaktadır. *WhatIf* çözümlemesi, politikada ya da varlıklarda geçici olarak meydana gelen değişiklikleri işleme koymadan önce etkilerini sınamak amacı ile kullanılmaktadır.
- *ReiAction:* Politikalar etki alanındaki eylemler üzerinden tanımlanmaktadır. Bu ontoloji, bütün eylemler için gerekli olan özellikleri içermekte, ayrıca eylemler ile ilgili olan etki alanına bağlı bilgilerin eklenmesine de izin vermektedir.

3.4. Politika Ontolojisi Tanımlama Adımları

Politika ontolojisi tanımlanırken izlenen adımlar aşağıda yer almaktadır:

1. Rei politika ontolojilerinin isim uzayı (namespace) tanımlamaları yapılmaktadır.
2. *entity:Variable* sınıfı yaratılmaktadır. Kurallar oluşturulurken kullanılacak olan aktörler bu sınıfın altında yaratılan örneklerdir.
3. *constraint:SimpleConstraint* sınıfı yaratılmaktadır. Bu sınıf altında oluşturulacak olan kısıt örnekleri yer almaktadır. Bu örnekler; *constraint:subject*, *constraint:predicate* ve *constraint:object* nesne özellikleri kullanılarak tanımlanmaktadır.
   Eğer karmaşık kısıt tanımlamaları yapılacaksa, yapılacak olan tanıma göre *constraint:And*, *constraint:Or* ve *constraint:Not* sınıfları yaratılmaktadır. *constraint:And* ve *constraint:Or* sınıflarının örnekleri *constraint:First* ve *constraint:Second* nesne özelliklerine sahiptir. *constraint:Not* sınıfının örnekleri ise sadece *constraint:First* nesne özelliğine sahiptir. Bu özellikler değerlerini *constraint:SimpleConstraint* sınıfının örneklerinden almaktadır.
4. *action:DomainAction* sınıfı yaratılmaktadır. Bu sınıfın altında oluşturulan örnekler kurallar için kullanılacak olan eylemlerin tanımlamalarıdır. Bu tanımlamalar, eylemin aktörünü belirten *action:actor*, eylemin yerini gösteren *action:location* ve *constraint:SimpleConstraint* sınıfından ya da karmaşık kısıt değerlerinden birini alan *action: precondition* nesne özelliklerini kullanmaktadırlar.
- Oluşturulacak politika nesnesinin türüne göre *deontic:Permission*, *deontic:Prohibition*, *deontic:Obligation* ya da *deontic:Dispensation* sınıfları yaratılmaktadır. Bu sınıfların altında oluşturulan örnekler, ilgili eylemi belirten *deontic:action*, aktör değerini alan *deontic:actor* ve *constraint:SimpleConstraint* sınıfından ya da karmaşık kısıt değerlerinden birini alan *deontic:constraint* nesne özelliklerini kullanmaktadırlar.
- Oluşturulan politika nesnelerinin onaylanması için *policy:Granting* sınıfı yaratılmaktadır. Bu sınıfın örnekleri, ilgili politika nesnesinin değerini alan *policy:deontic* ve politikanın aktör değerini alan *policy:to* nesne özelliklerini kullanmaktadırlar.
- Politikanın oluşturulmasın son adımında, *policy:Policy* sınıfı yaratılmaktadır. Bu sınıfın altında yaratılan örnekler, politikanın aktörünü belirten *policy:actor*, ilgili eylemi belirten *policy:action* ve bir önceki adımda yapılan onaylamayı belirten *policy:grants* nesne özelliklerini kullanmaktadırlar.

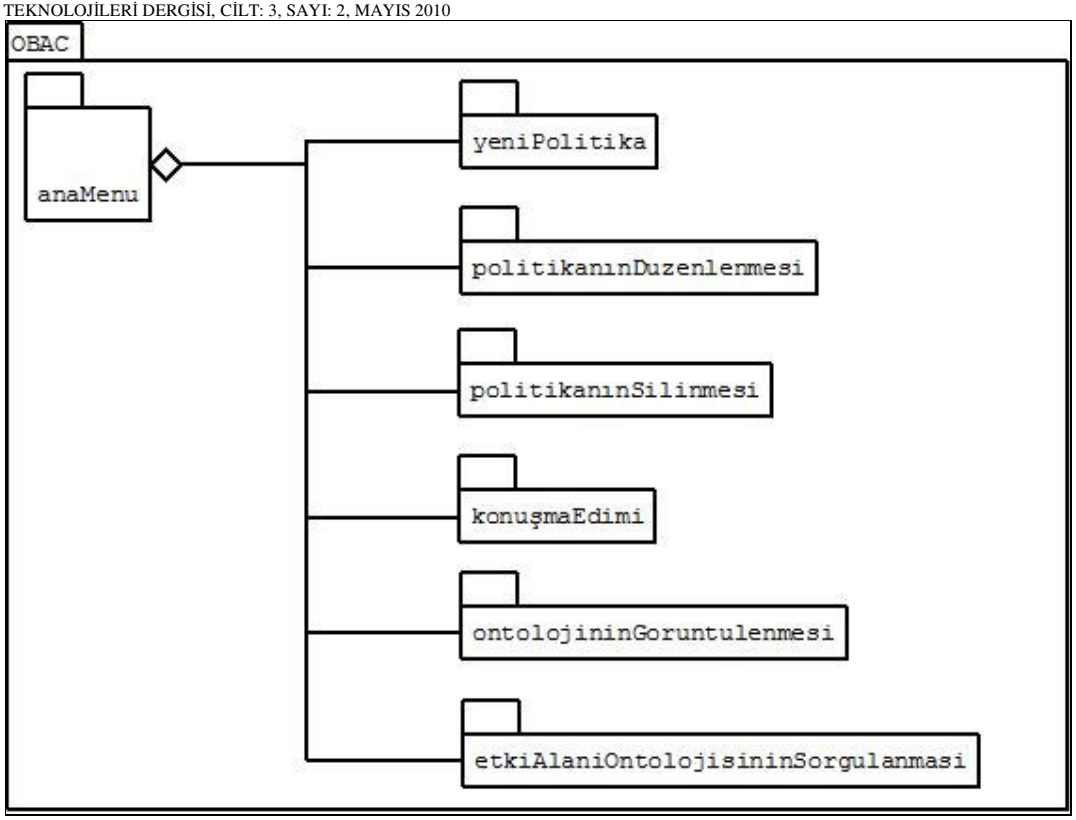
4. UYGULAMA MİMARİSİ VE YAZILIM

Bir kullanıcı arayüzü aracılığı ile politikaların yaratılması, düzenlenmesi, silinmesi ve politikalar üzerinde sorguların yapılabilmesi için Ontoloji Tabanlı Erişim Denetimi uygulaması geliştirilmiştir. Bu uygulama Java programlama dili kullanılarak gerçekleştirilmiştir. Bu amaçla, Anlamsal Web uygulamalarının geliştiriminde sıklıkla kullanılan Jena Anlamsal Web Çatısından yararlanılmıştır. Temel Jena işlemleri makalenin sonunda yer alan Ek kısmında örneklendirilmektedir. İzleyen kısımlarda politika motoru yapısı ve uygulama yazılımında yer alan işlemler anlatılmaktadır.

4.1. Politika Motoru Yapısı

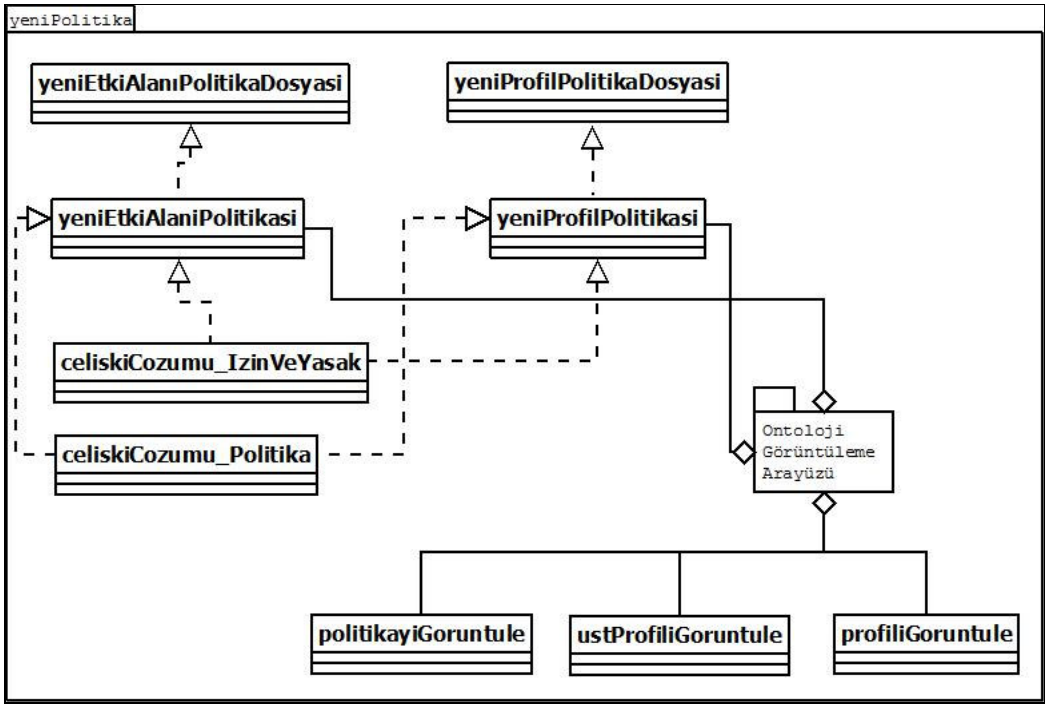
Ontoloji Tabanlı Erişim Denetimi için geliştirilen politika motorunun yapısal kullanım görünümü Şekil 4'de yer aldığı gibi 6 birimden oluşmaktadır. Bunlar:

- Yeni Politika (New Policy)
- Politikanın Düzenlenmesi (Edit Policy)
- Politikanın Silinmesi (Delete Policy)
- Konuşma Edimi (Speech Act)
- Ontolojinin Görüntülenmesi (View Ontology)
- Etki Alanı Ontolojisinin Sorgulanması (Query Domain Ontology)



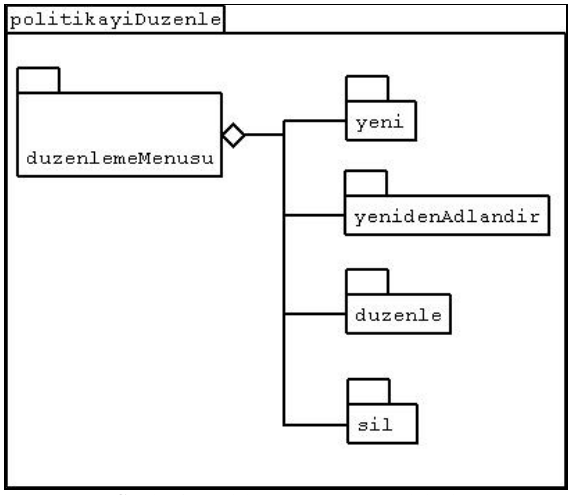
Şekil 4. Ontoloji tabanlı erişim denetimi politika motorunun yapısal kullanım görünümü

Şekil 5'de yer alan yeni politika tanımının yapıldığı birimde, *yeniEtkiAlanıPolitikaDosyasi()* ve *yeniProfilPolitikaDosyasi()* sınıfları sırası ile *yeniEtkiAlanıPolitikasi()* ve *yeniProfilPolitikasi()* sınıflarını kullanmaktadır. Bu sınıflar da *celiskiCozumu_IzinVeYasak()*, *celiskiCozumu_Politika()* sınıflarını ve ontoloji görüntüleme arayüzünü kullanmaktadır.



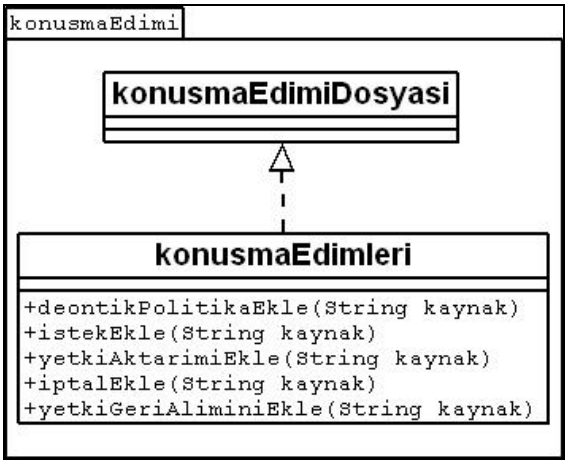
Şekil 5. Yeni politika tanımı birimi

Politika düzenleme birimi Şekil 6'da gösterilmiştir. Bu birim; yeni, yenidenAdlandir, duzenle ve sil birimlerinden oluşmaktadır.



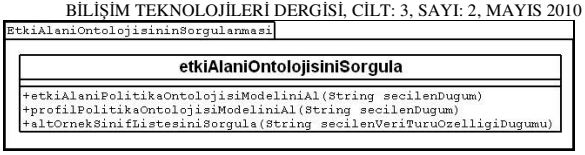
Şekil 6. Politika düzenleme birimi

Konuşma edimi biriminin yer aldığı Şekil 7'de *konusmaEdimiDosyasi()* sınıfı, *konusmaEdimleri()* sınıfını kullanmaktadır. Bu sınıfta; *deontikPolitikaEkle(String kaynak)*, *istekEkle(String kaynak)*, *yetkiAktarimiEkle(String kaynak)*, *iptalEkle(String kaynak)* ve *yetkiGeriAliminiEkle(String kaynak)* işlemleri gerçekleştirilmektedir.



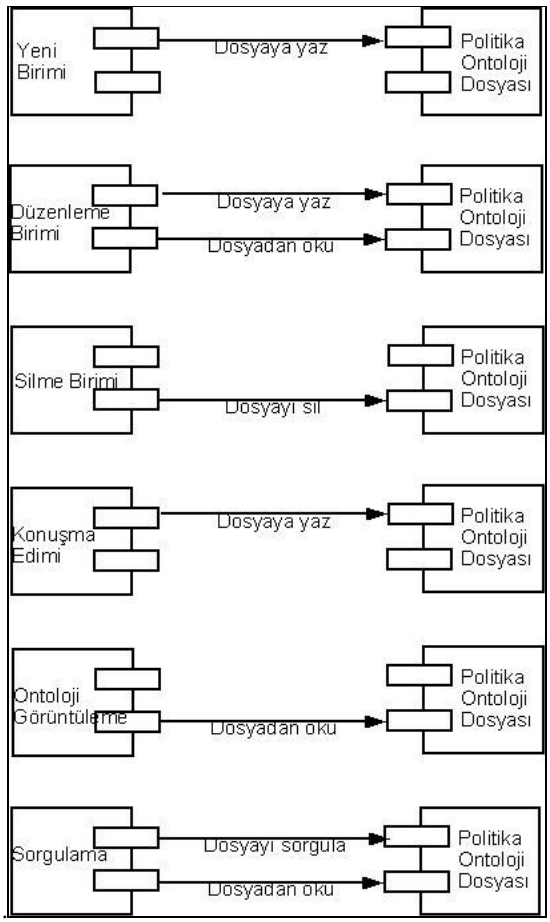
Şekil 7. Konuşma edimi birimi

Şekil 8, etki alanı ontolojisi sorgulama birimini göstermektedir. Bu birim *etkiAlanıOntolojisiniSorgula()* sınıfından oluşmaktadır. Bu sınıfta etki alanı ve profil politika ontolojileri kullanılarak gerçekleştirilen sorgulamalar için gerekli olan temel işlemler; *etkiAlaniPolitikaOntolojisiModeliniAl(String secilen Dugum)*, *profilPolitikaOntolojisiModeliniAl(String secilenDugum)* ve *altOrnekSinifListesiniSorgula(String secilenVeriTuruOzellligiDugumu)*'dır.



Şekil 8. Etki alanı ontolojisi sorgulama birimi

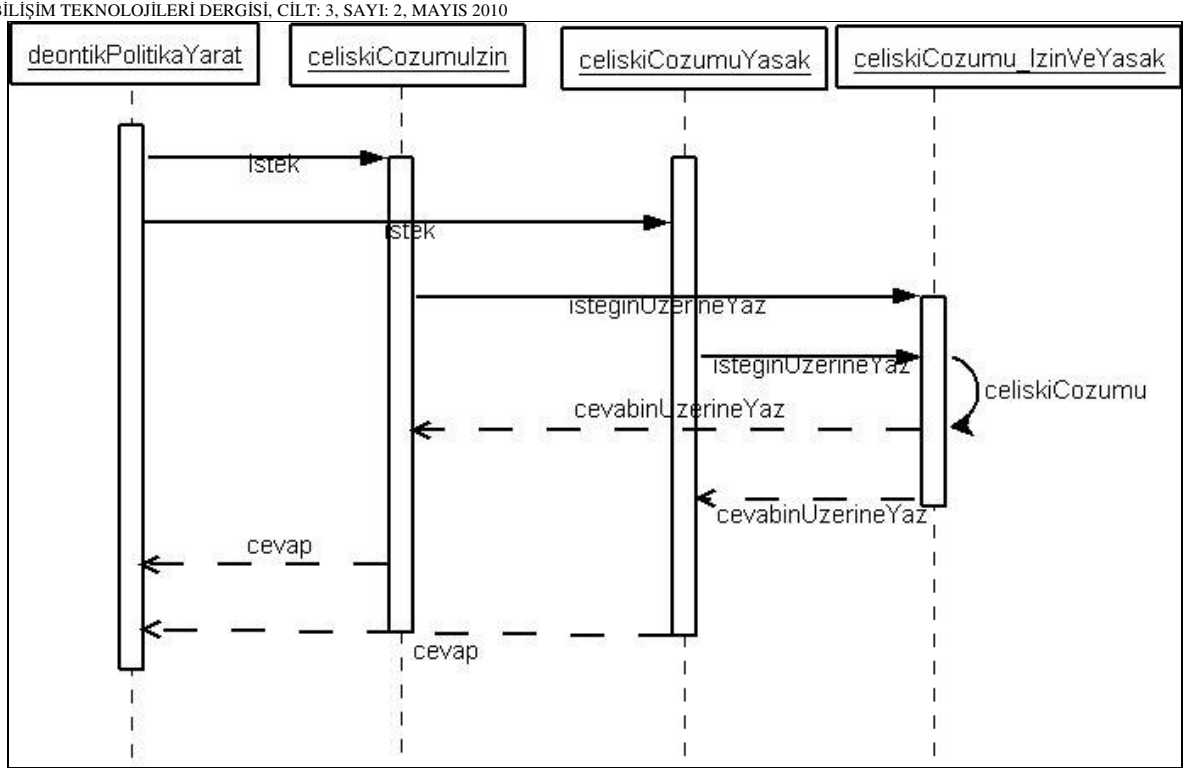
Şekil 9'da politika motorunun yapısal paylaşılan veri görünümü (architectural shared data style) yer almaktadır. Burada her bir birimin politika ontoloji dosyası üzerinde gerçekleştirdiği işlemler gösterilmektedir.



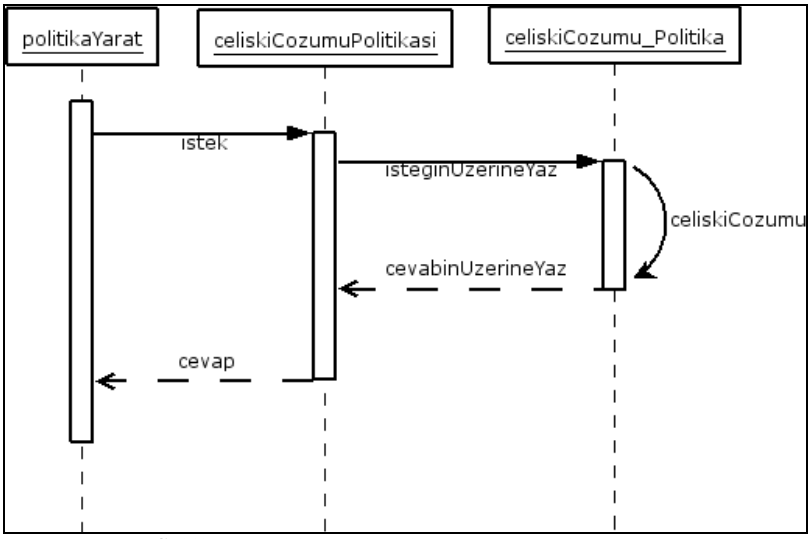
Şekil 9. Politika motoru yapısal paylaşılan veri görünümü

Politika kuralı çelişkisinin çözümü için izinin yasak üzerine ya da yasağın izin üzerine önceliği kullanıcı tarafından belirtildikten sonra çelişki çözümü gerçekleştirilmektedir. Bu akış Şekil 10'da görülmektedir. Deontik politika oluşturulurken izin ve yasak sınıflarında kural çelişkisi olup olmadığı tespit edilmektedir. Eğer herhangi bir kural çelişkisi oluşmuş ise *celiskiCozumu_IzinveYasak()* sınıfında kullanıcıdan alınan üstünlük bilgisine göre çelişki çözülmektedir.

Politikalar arasında oluşabilecek çelişkilerin çözümü için ilgili akış şeması Şekil 11'de yer almaktadır. Politika oluşturulurken, çelişki tespiti yapılıp, çelişki olması durumunda hangi politikanın önceliğe sahip olacağı kullanıcı tarafından belirtildikten sonra politika çelişkisi çözümü gerçekleştirilmektedir.



Şekil 10. Politika kuralı çelişki çözümü akış diyagramı



Şekil 11. Politika çelişkisi çözümü akış diyagramı

4.2. Uygulama Yazılımı

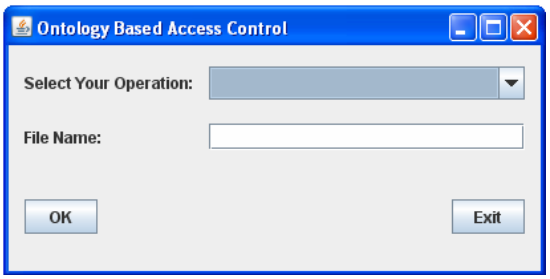
Bir kullanıcı arayüzü aracılığı ile politikaların yaratılması, düzenlenmesi, silinmesi ve politikalar üzerinde sorguların yapılabilmesi için Ontoloji Tabanlı Erişim Denetimi uygulama yazılımı geliştirilmiştir. Bu amaçla, Anlamsal Web uygulamalarının geliştiriminde sıklıkla kullanılan Jena Anlamsal Web Çatısından yararlanılmıştır.

Java programlama dilinin kullanıldığı Ontoloji Tabanlı Erişim Denetimi uygulaması Eclipse [20] ortamı kullanılarak geliştirilmiştir. Arayüz tasarımında Eclipse ortamına uyumlu bir ek olan Jigloo [21] arayüz yapıcısı tercih edilmiştir.

Uygulama çalıştırıldığında, öncelikle, kullanıcı gerçekleştirmek istediği işlemi seçmekte ve ontoloji dosyasının ismini belirtmektedir. Kullanıcının karşılaşacağı ilk ekran görüntüsü Şekil 12'de yer almaktadır. Kullanıcının gerçekleştirebileceği işlemler sırası ile aşağıdaki gibidir:

- Yeni Etki Alanı Politikası (New Domain Policy)
- Yeni Profil Politikası (New Profile Policy)

- Politika Dosyasının Düzenlenmesi (Edit Policy File)
- Politika Dosyasının Silinmesi (Delete Policy File)
- Konuşma Edimleri (Speech Act)
- Ontolojinin Görüntülenmesi (View Ontology)
- Etki Alanı Ontolojisinin Sorgulanması (Query Domain Ontology)

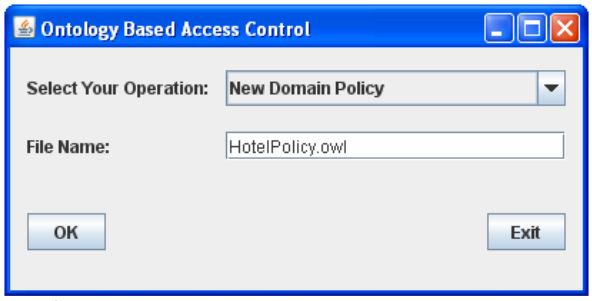


Şekil 12. Ontoloji Tabanlı Erişim Denetimi arayüzü

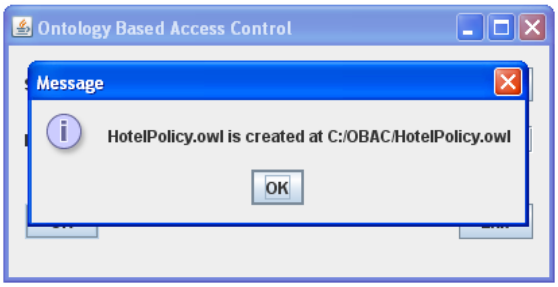
4.2.1 Yeni Politika Oluşturulması

Yeni etki alanı politikası arayüzü aracılığı ile etki alanını temel alan yeni bir politika ontolojisi oluşturulmaktadır. Profil tabanlı politika ontolojisinin yaratılması ise yeni profil politikası arayüzü aracılığı gerçekleştirilmektedir. Yeni etki alanı ve profil politika ontolojilerinin oluşturulmasında aynı işlemler gerçekleştirilmektedir. Bu nedenle, burada sadece yeni etki alanı politikası arayüzü anlatılacaktır.

Kullanıcının dosya adı kısmına girdiği yeni politika ontolojisi Şekil 13 ve Şekil 14'de görüldüğü gibi yaratılmaktadır.



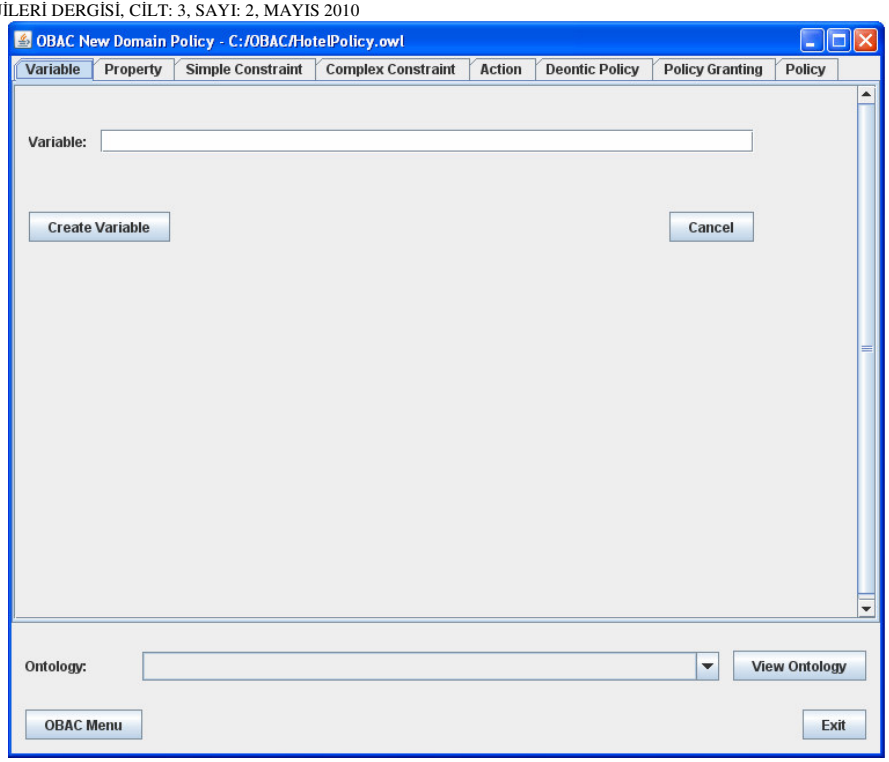
Şekil 13. Yeni etki alanı politikasının yaratılması



Şekil 14. Yeni etki alanı politikasının yaratıldığının onayı

Yeni etki alanı politika ontoloji dosyası oluşturulduktan sonra kullanıcının karşılaşacağı ekran görüntüsü Şekil 15'de yer almaktadır. Bu pencerede yer alan ve politika ontolojisinin yaratılması için gerekli olan bölümler aşağıdaki gibidir:

- *Değişken (Variable):* Değişken tanımları yapılmaktadır.
- *Özellik (Property):* Özellik tanımlarının yapıldığı bu sekmede Nesne ve Veri türü olmak üzere iki özellik tanımı yapılmaktadır.
- *Basit Kısıt (Simple Constraint):* Basit kısıt tanımının yapıldığı bu sekmede 4 alan tanımlanmaktadır. Bunlar; yaratılacak *basit kısıtın adı*, profil ya da etki alanı ontolojilerinden gelen *nesne*, *yüklem* ve *özne*dir.
- *Karmaşık Kısıt (Complex Constraint):* Karmaşık kısıt tanımında; yaratılacak olan *karmaşık kısıdın adı*, *türü* (AND, OR ya da NOT), *ilk kısıt* ve *ikinci kısıt* olmak üzere 4 alan yer almaktadır.
- *Eylem (Action): Eylemin adı*, *aktör*, *konum* ve *önkoşul* eylem tanımında yer alan alanlardır. Aktör, üzerinde çalışılan ontolojiden eklenebildiği gibi profil ontolojisinden de eklenebilmektedir. Konum ve önkoşul bilgileri etki alanı ontolojisinden eklenebilmektedir.
- *Deontik Politika (Deontic Policy):* Deontik politika sekmesinde *politika türü*, *politikanın adı*, *aktör*, *eylem* ve *koşul* alanları yer almaktadır. Politika türü; izin, yasak, zorunluluk ve özel izin olmak üzere 4 şekilde olmaktadır. Aktör bilgisi profil ontolojisinden eklenebilmektedir.
- *Politika Onay (Policy Granting):* Politika onayının yapıldığı bu sekmede 3 alan yer almaktadır. Bu alanlar; *politika onayının adı*, *onaylanacak deontik politikanın adı* ve *onaylamanın kime yapılacağı*dır.
  Onaylamanın kime yapılacağı, üzerinde çalışılan ontolojiden eklenebildiği gibi profil ontolojisindende eklenebilmektedir.
- *Politika (Policy):* Politika tanımlamanın son işlemi olan politika sekmesinde *politikanın adı*, *aktör*, *eylem* ve *onaylanan politikanın adı* olmak üzere 4 alan bulunmaktadır. Politika aktörü bilgisi profil ontolojisinden eklenebilmektedir.



Şekil 15. Yeni etki alanı politikası ara yüzü

Politika ontolojisinin oluşturulmasında kullanılan bu alanların yanı sıra, varolan tanımlanmış ontolojilerin görüntülenmesini sağlayan Ontoloji Görüntüle (View Ontology) kısmı bu pencerede yer almaktadır. Ontoloji görüntüleme kısmında yer alan görüntülenebilecek ontolojiler şunlardır:

- *Etki Alanı Ontolojisi (Domain Ontology): Kullanılan etki alanı ontolojisini göstermektedir.*
- *Üst Profil Ontolojisi (Meta Profile Ontology): Profil tanımlamak için gerekli üst verilerin tanımlandığı ontolojidir.*
- *Profil Ontolojisi (Profile Ontology): Profil, profil adı, yaş ve meslek bilgilerinin tanımlandığı ontolojidir.*
- *Etki Alanı Politika Ontolojisi (Domain Policy Ontology): Üzerinde çalışılan politika ontolojisini göstermektedir.*

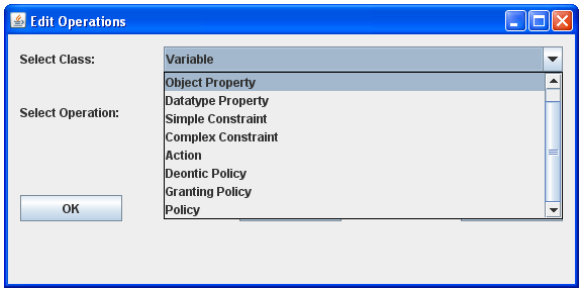
4.2.2 Politika Dosyasının Düzenlenmesi

Politika dosyasının düzenlenmesi kısmında sistemde yer alan politika ontolojilerinin üzerinde değişiklik yapılabilmesi sağlanmaktadır. Düzenleme işlemi için kullanıcı öncelikle üzerinde çalışacağı politika ontoloji dosyasının adını belirtmektedir. Dosya açıldıktan sonra politika ontolojisinde yer alan ve üzerinde değişiklik yapılabilecek olan Şekil 16'da yer alan ontoloji sınıfları şunlardır:

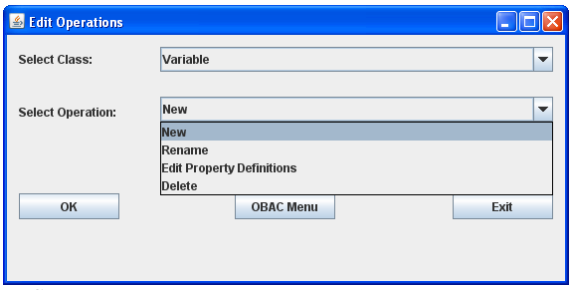
- *Değişken (Variable)*
- *Nesne Özelliği (Object Property)*
- *Veri türü Özelliği (Datatype Property)*
- *Basit Kısıt (Simple Constraint)*
- *Karmaşık Kısıt (Complex Constraint)*
- *Eylem (Action)*
- *Deontik Politika (Deontic Policy)*
- *Politikanın Onaylanması (Granting Policy)*
- *Politika (Policy)*

Bu sınıfların her biri için gerçekleştirilebilecek işlemler ise Şekil 17'de gösterilmiş olan aşağıdaki işlemlerdir:

- *Yeni (New):* Yeni işlemi, ilgili sınıf için yeni bir tanımlama yapılmasını göstermektedir.
- *Yeniden adlandırma (Rename):* Yeniden adlandırma işleminde önce ilgili sınıf örneğinin değiştirilmek istenen varolan adı daha sonra yeni adı girilmektedir.
- *Özellik Tanımlarının Düzenlenmesi (Edit Property Definitions):* Her bir sınıfın örneği ile ilgili düzenlemelerin yapılması Özellik Tanımlarının Düzenlenmesi adı altında gerçekleştirilmektedir. Burada, herhangi bir sınıftaki özelliklere politika içindeki tanımlarını belirten bir tanımlama özelliği eklenmekte ya da varolan tanımlamaları güncellenmektedir.
- *Sil (Delete):* Herhangi bir sınıfta yer alan bir örneğin silinmesi bu işlem adımında gerçekleştirilmektedir.



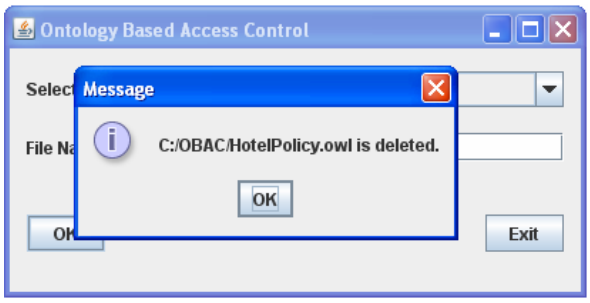
Şekil 16. Düzenleme işleminin yapılabileceği politika ontolojisi sınıfları



Şekil 17. Sınıfların düzenlenmesinde yapılabilecek işlemler

4.2.3 Politika Dosyasının Silinmesi

Sistemde yer alan politika ontolojisinin silinmesi için kullanıcı öncelikle dosya adını belirtmektedir. Belirtilen politika ontolojisi sistemde yer alıyor ve başarılı bir şekilde silme işlemi gerçekleştiyse Şekil 18'de belirtilen dosya silindi iletisi, belirtilen dosyanın sistemde bulunamadığı ve silme işleminin gerçekleştirilemediği durumda ise dosya bulunamadı iletisi görüntülenecektir.

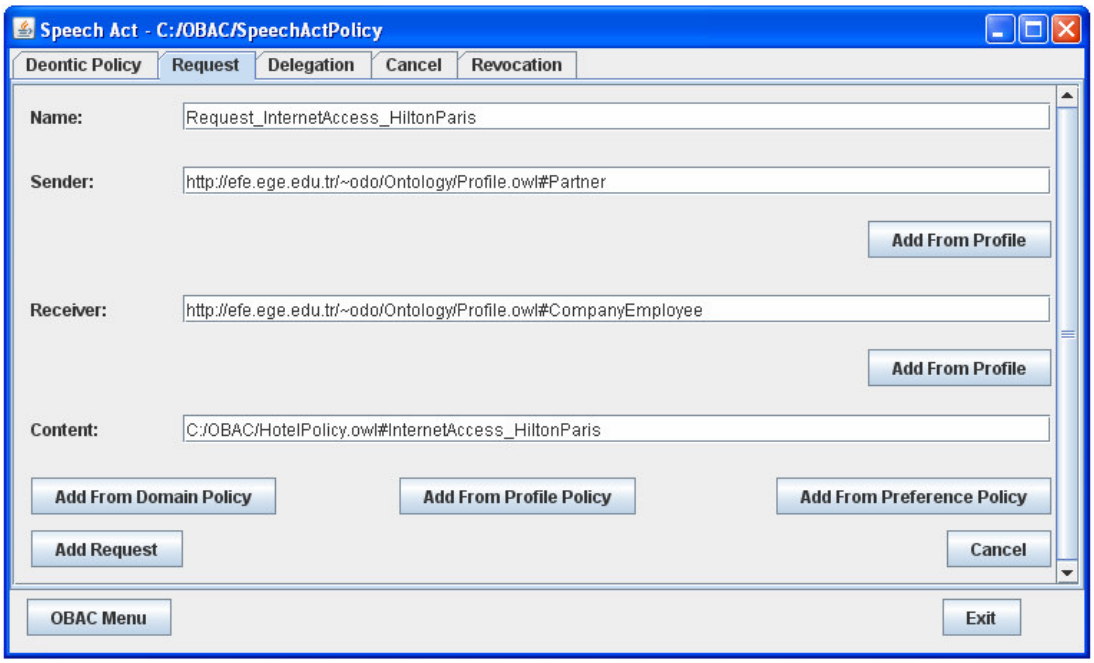


Şekil 18. Politika ontolojisinin başarılı bir şekilde silinmesi

4.2.4 Konuşma Edimleri

Güvenlik denetimini sağlamak amacı ile kullanılan konuşma edimlerinin oluşturulması için kullanıcı öncelikle oluşturulacak dosya adını belirtmektedir. Şekil 19'da yer alan konuşma edimi penceresi 5 sekmeden oluşmaktadır:

- *İstek (Request)*
- *Yetki aktarımı (Delegation)*
- *İptal (Cancel)*
- *Yetkinin geri alımı (Revocation)*
- *Deontik politika (Deontic policy)*



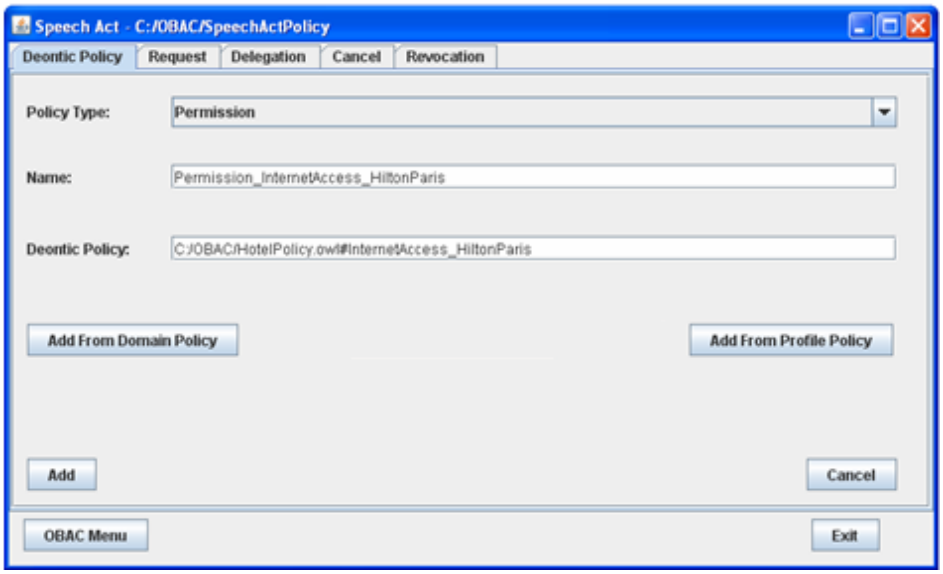
Şekil 19. Konuşma edimi penceresi

İstek, Yetki aktarımı, İptal ve Yetkinin geri alımı sekmeleri sırası ile aşağıdaki dört alandan oluşmaktadır:

- *Ad (Name):* Oluşturulan konuşma ediminin adıdır.

- *Gönderici (Sender):* Konuşma edimi ile ilgili istekte bulununan varlıktır.
- *Alıcı (Receiver):* Konuşma edimi ile ilgi isteği alan varlıktır.
- *İçerik (Content):* Konuşma ediminin ilgili olduğu içeriği belirtmektedir.

Deontik politika sekmesi, Şekil 20'de görülen, oluşturulacak *deontik politikanın türü*, *adı* ve etki alanı, profil ya da başka bir politika ontolojisinden alınan *deontik politika alanlarından* oluşmaktadır.



Şekil 20. Deontik politika sekmesi

4.2.5 Ontolojilerin Görüntülenmesi

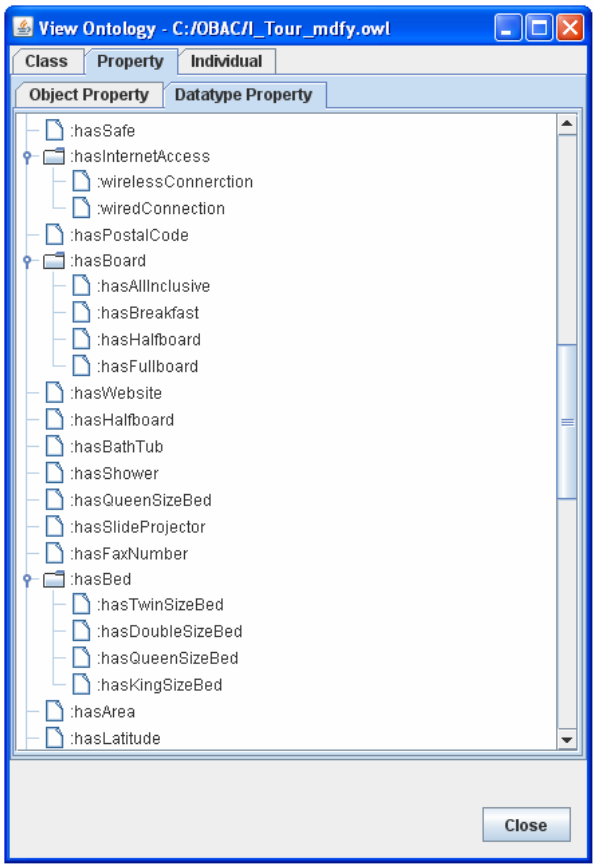
Sistemde yer alan bir ontolojinin sınıf, özellik ve örnek sıradüzensellerinin görüntülenmesi bu bölümde gerçekleşmektedir. Kullanıcı, Ontoloji Görüntüle işlemini seçtikten sonra görüntülemek istediği ontoloji dosyasının adını belirtmektedir. Ontolojinin özellik sıradüzenseli; nesne (object) ve veri türü (datatype) olmak üzere iki şekilde görüntülenmektedir. Örnek bir ontolojinin veri türü özellik sıradüzenseli ise Şekil 21'de gösterilmiştir.

4.2.6 Etki Alanı Ontolojisinin Sorgulanması

Kullanıcı, etki alanı ontolojisinin sorgulanması işlemini seçtikten sonra, etki alanı ontolojisinin sırasıyla sınıf, özellik, örnek ve seçilmiş özelliğin örnek sıradüzenselleri görüntülenmektedir.

Etki alanı ontolojisinin sınıf, özellik ve örnek sıradüzenselleri ontoloji görüntüleme kısmında anlatılanlar ile aynıdır. Ancak, etki alanı ontolojisinin sorgulanmasında, ontoloji görüntüleme kısmından farklı olarak kullanıcı tarafından seçilmiş sınıfın, örneğin ya da seçilmiş bir özellik ile ilgili politika ontolojileri içerisinde sorgulamalar gerçekleştirilmektedir. Ayrıca, Seçilmiş Özellik Örneği (Selected Property Individual) sekmesinde seçilmiş bir özelliği kullanan örnek sıradüzensellerinin görüntülenmesi yer almaktadır. Bu sekmede, bütün örneklerin listelenmesi yerine sadece özellik sıradüzenselinden seçilmiş olan herhangi bir özelliğin kullanıldığı örnekler listelenmektedir.

Etki alanı ontolojisinin sorgulanması penceresinin alt kısmında yer alan Sorgu Sonuçları (Query Results) kısmında ise seçilen sınıf, özellik ve örnek ile ilgili sorgu



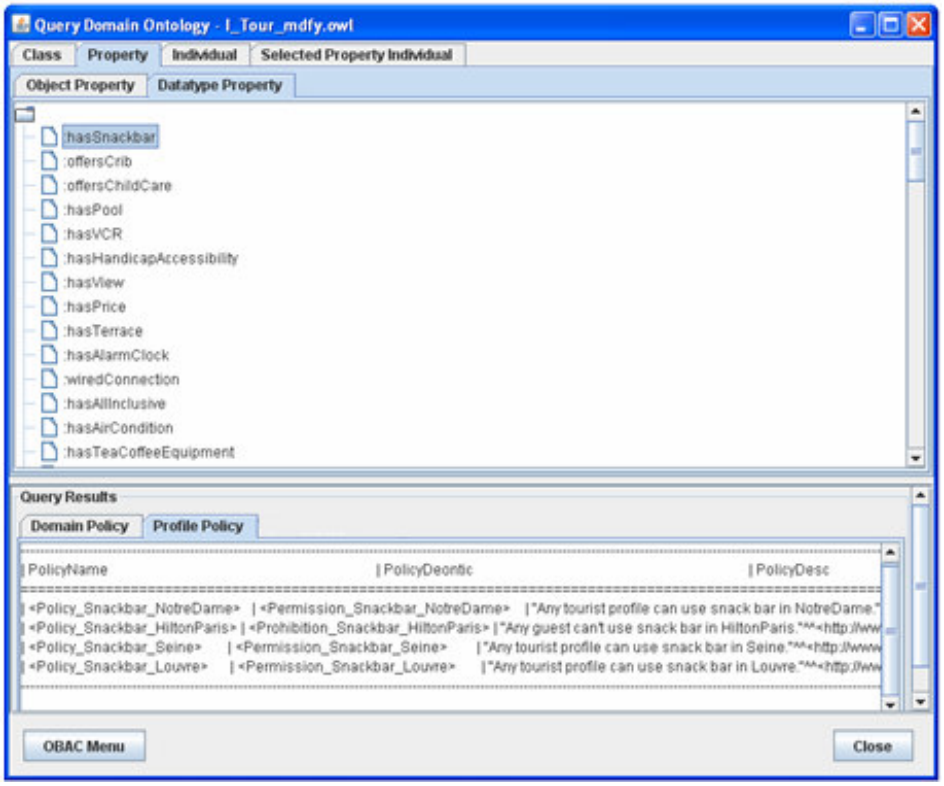
Şekil 21. Ontolojinin veri türü özellik sıradüzenseli

sonuçları listelenmektedir. Daha iyi bir kişiselleştirmenin sağlanabilmesi için sorgular politikalar üzerinden gerçekleştirilmektedir. Böylelikle kullanıcı için sorgu sonuçlarıda daraltılmış olmaktadır. Sorgu sonuçları iki bölümden oluşmaktadır:

- Etki Alanı Politikası (Domain Policy)
- Profil Politikası (Profile Policy)

Seçilen sınıf, özellik ya da örnek ile ilgili sorgu işlemleri her iki politika ontolojisi içerisinde çalıştırıldığından her bir ontolojiden gelen sonuçlar ilgili ontolojinin sekmesinde listelenmektedir.

Örneğin, turizm etki alanında yer alan "*hasSnackBar=true*" veri türü özelliği seçildiğinde, bu özellik ile ilgili profil politikası ontolojisinden gelecek olan sorgu sonuçları Şekil 22'de görülmektedir.



Şekil 22. *hasSnackBar* özelliği ile ilgili profil politikasından gelen sorgu sonuçları

Profil politika ontolojisinden gelen sorgu sonuçlarını gösteren Şekil 22'de 2 yasak 2 izin olmak üzere 4 sonuç yer almaktadır. Bu sonuçların elde edildiği profil politika ontolojisinde çalıştırılan sorgu Şekil 23'de yer almaktadır. Burada politika adı, politika deontik nesnesinin türü ve politika açıklaması listelenmektedir.

```
final Query queryString = QueryFactory.create(
"PREFIX dc:
<http://efe.ege.edu.tr/~ozgucan/PhD/Ontology/I_Tour_mdfy.ow
l#> " +
"PREFIX tourism:
<http://efe.ege.edu.tr/~odo/Ontology/I_Tour_mdfy.owl#>"+
"PREFIX policy:
<http://www.cs.umbc.edu/~lkagal1/rei/ontologies/ReiPolicy.o
wl#>"+
"PREFIX action:
<http://www.csee.umbc.edu/~lkagal1/rei/ontologies/ReiAction
.owl#>"+
"PREFIX constraint:
<http://www.cs.umbc.edu/~lkagal1/rei/ontologies/ReiConstrai
nt.owl#>"+
"SELECT ?PolicyName ?PolicyDeontic ?PolicyDesc WHERE {
?PolicyName policy:action ?xy."+
"?xy action:precondition ?z ." +
"?z constraint:object tourism:"+selectedNode+" ."+
"?PolicyName policy:grants ?w . "+
"?w policy:deontic ?PolicyDeontic ."+
"?PolicyDeontic policy:desc ?PolicyDesc }");
```

Şekil 23. Profil politika ontolojisinde çalıştırılan sorgu

5. SONUÇLAR

Ontoloji tabanlı uygulamalar Anlamsal Web tabanlı sistemlerde önemli bir role sahiptir. Kaynakçada, ontoloji mühendisliği ve tekniklerini destekleyen birçok araç yer almasına rağmen ontoloji tabanlı uygulamaların nasıl gerçekleştirileceğini belirten mimarilerin yönergelerine çok sık rastlanılmamaktadır. Bu çalışmada, Anlamsal Web için önemli konulardan biri olan güvenlik kavramının sağlanmasında kullanılan politikaların yönetimi için Ontoloji Tabanlı Erişim Denetim Modeli ve bu modelin Jena Anlamsal Web çatısı kullanılarak bir uygulaması geliştirilmiştir. Bu modelde, politikaları oluşturan nesne, eylem ve koşullar anlamsallığın sağlanabilmesi için ontolojik olarak tanımlanmaktadır. Ayrıca, kişiselleştirmenin sağlanabilmesi için kullanıcı profilleri oluşturulmakta, politika ontolojileri de etki alanı ve profil tabanlı olmak üzere iki şekilde yaratılmaktadır. Geliştirilen uygulama ile politika ontolojilerinin yaratılması, düzenlenmesi, silinmesi ve sorgulanması işlemleri gerçekleştirilmektedir. Geliştirilen bu uygulama ile ontoloji tabanlı bilgi sistemi geliştirilirken erişim denetiminin sağlanabilmesi için politika ontolojileri yaratılıp düzenlenebilir ve politika motoru olarak sistem ile bütünleştirilebilir. Gelecek çalışmalarda, Anlamsal Web'de yer alan bir diğer önemli kavram olan gizliliğin sağlanmasına yönelik olarak eklentiler yapılarak model genişletilecektir. Bu amaçla gizlilik ontolojisi geliştirilecek ve uygulamaya eklenerek sorgulama işlemleri genişletilecektir.

KAYNAKLAR

[1] N. Suri, A. Uszok, "Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KaoS, Rei, and Ponder", **2nd International Semantic Web Conference (ISWC 2003)**, Sanibel Island, Florida, USA, 419-437, 2003.

[2] T. Finin, "ROWLBAC - Representing Role Based Access Control in OWL", **Proceedings of the 13th Symposium on Access Control Models and Technologies**, Colorado, USA, 2008.

[3] T. R. Gruber, "Toward principles for the Design of Ontologies Used for Knowledge Sharing", *International Journal of Human-Computer Studies*, 43(5-6), 907-928, 1993.

[4] Internet: KAoS Policy and Domain Services, http://www.ihmc.us/research/projects/KAoS, 23.06.2010.

[5] Internet: Rei: A Policy Specification Language, http://rei.umbc.edu, 23.06.2010.

[6] Internet: Ponder: A Policy Language for Distributed Systems Management, http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml, 23.06.2010.

[7] G. Tonti, J. M. Bradshaw, R. Jeffers, R. Monranari, N Suri, A. Uszok, "Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KaoS, Rei, and Ponder", **2nd International Semantic Web Conference (ISWC 2003)**, Sanibel Island, Florida, USA, 419-437, 2003.

[8] L. Kagal, T. Finin, A. Joshi, "A Policy Based Approach to Security for the Semantic Web", **2nd International Semantic Web Conference (ISWC 2003)**, Sanibel Island, Florida, USA, 402-418, 2003.

[9] G. Antoniou, G. and F. van Harmelen, **A Semantic Web Primer**, The MIT Press, 2004.

[10] Ö. Can, **Anlamsal Web için Kişiselleştirilebilir Ontoloji Tabanlı Erişim Denetimi ve Politika Yönetimi**, Doktora Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, 2009.

[11] L. Kagal, T. Finin, M. Paolucci, N. Srinivasen, K. Sycara, G. Denker, "Authorization and Privacy for Semantic Web Services", *IEEE Intelligent Systems*, 19(4), 50-56, July/Aug, 2004.

[12] B. Thuraisingham, **Building Trustworthy Semantic Webs**, Auerbach Publications, 2007.

[13] Internet: Jena - A Semantic Web Framework for Java, http://jena.sourceforge.net, 23.06.2010.

[14] Internet: E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query, 23.06.2010.

[15] S. Gauch, M. Speretta, A. Chandramouli, A. Micarelli, "User Profiles for Personalized Information Access", **The Adaptive Web 2007**, 54-89, 2007.

[16] E. Rich, "Users are individuals: Individualizing User Models", *Int. J. Hum. Comput. Stud.*, 51(2), 323-338, 1999.

[17] A. Katifori, M. Golemati, C. Vassilakis, G. Lepouras, C. Halatsis, "Creating an Ontology for the User Profile: Method and Applications", **In the proceedings of the First IEEE International Conference on Research Challenges in Information Science (RCIS)**, Morocco, 407-412, 2007.

[18] O. Bursa, M. O. Ünalır, "Anlamsal Web Portallarda Profil Yönetimi", **Yazılım Kalitesi ve Yazılım Geliştirme Araçları Sempozyumu 2008 (YKGS 2008)**, İstanbul, 9-10 Ekim, 2008.

[19] Ö. Can, M. O. Ünalır, "Personalizable Ontology-Based Access Control", *Gazi Üniversitesi Fen Bilimleri Dergisi*, 23(4), 2010.

[20] Internet: Eclipse, http://www.eclipse.org, 23.06.2010.

[21] Internet: Jigloo SWT/Swing GUI Builder for Eclipse and WebSphere, http://www.cloudgarden.com/jigloo, 23.06.2010.

EK

Anlamsal Web uygulamalarının geliştiriminde kullanılan temel Jena işlemleri aşağıda örneklendirilmektedir.

İsim uzaylarının yaratılması:
String ReiAction="http://www.cs.umbc.
edu/~lkagal1/rei/ontologies/ReiAction.owl#";
PrefixMapping ReiActionNS=tourismModel
Policy.setNsPrefix("action",ReiAction);

Bir ontoloji modelinin yaratılması:
OntModel tourismModelPolicy=Model Factory.createOntology
Model();

Boş bir OWL modelinin yaratılması:
OntModel tourismModelPolicy=Model
Factory.createOntologyModel(OntModel Spec.OWL_MEM, null);
Ontology tourismModelPolicyOntology=
tourismModelPolicy.createOntology(OntologyURI);

Bir nesne özelliğinin yaratılması:
ObjectProperty actionActor=tourism
ModelPolicy.createObjectProperty(Rei Action + "actor");

Bir veri türü özelliğinin yaratılması:
DatatypeProperty policyDesc=tourism
ModelPolicy.createDatatypeProperty(Rei Policy + "desc");

Bir sorgunun çalıştırılması:
QueryExecution qexec=QueryExecution
Factory.create(queryStringDomain, tourismPolicyModel) ;
ResultSet results=qexec.execSelect();
ResultSetFormatter fmt=new ResultSetFormatter();