

ONTECO: Veri Entegrasyon-Temelli Modüler Ontoloji Geliştirme Aracı

Tuğba Özacar¹Övünç Öztürk²Murat Osman Ünalır³^{1,2,3}Bilgisayar Mühendisliği Bölümü, Ege Üniversitesi, İzmir¹e-posta: tugba.ozacar@ege.edu.tr²e-posta: ovunc.ozturk@ege.edu.tr³e-posta: murat.osman.unalir@ege.edu.tr

Özetçe

Ontolojiler, bir alana ait açık ve formal bir kavramsallaştırma sağladıkları için, veri entegrasyon sistemlerinde yaygın olarak kullanılmaktadırlar [1]. Ontoloji temelli veri entegrasyon sistemleri iki grupta incelenmektedir; (i) merkezi veri entegrasyon sistemleri (ii) eşler arası veri entegrasyon sistemleri. Merkezi veri entegrasyon sistemleri, farklı kaynak ontolojilerini karşılaştırabilir kılmak için, global bir ontolojiyi ortak bir sözlük olarak kullanırlar. Fakat tek parçalı, büyük bir global ontoloji kullanmak, ontoloji yaşam döngüsünün değişik safhalarında problemlere yol açmaktadır. Buna karşın, eşler arası veri entegrasyon sistemleri global bir ontolojiye ihtiyaç duymazlar. Bu sistemler farklı eşlere ait ontolojileri karşılaştırabilir kılmak için ontolojiler-arası eşleme yaparlar. Fakat pratikte ontolojiler arası eşlemelerin tanımlanması çok zor olmaktadır.

Bu çalışmada, merkezi veri entegrasyon sistemlerinin birçok özelleşmiş ontoloji modülünün birleşiminden oluşan bir global ontoloji kullanması önerilmektedir. Bu şekilde tanımlanmış bir global ontolojinin geliştirilmesine yönelik bir metodoloji ve metodolojinin uygulanmasını kolaylaştıracak bir ontoloji geliştirme aracı sunulmaktadır.

1. Giriş

Merkezi veri entegrasyon sistemlerinde, tek parçalı, büyük bir global ontoloji kullanmak, ontoloji yaşam döngüsünün değişik safhalarında problemlere yol açmaktadır. Bu problemlerin birçoğu global ontolojinin modülerleştirilmesi ile çözülmektedir [2]. Bu çalışmada modüler bir global ontoloji kullanılması önerilmektedir. Öngörülen modüller ve aralarındaki ilişkiler Şekil 1'de gösterilmektedir. Kullanılan modüller aşağıda tanımlanmaktadır;

Taban ontoloji modülü: Alan-bağımsız, birçok alana özgü bilgi bu modülde tanımlanır.

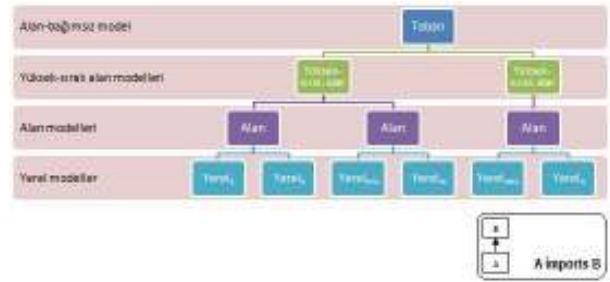
Yüksek-sıralı alan ontoloji modülü: Bazı alanlar arasında ortak kavramlar olabilir. Bu alan modellerinin entegrasyonu için, ortak kavramlar yüksek-sıralı alan ontoloji modüllerinde tanımlanırlar ve domain modüllerinde özelleştirilerek veya referans vererek kullanılırlar.

Alan ontoloji modülü: Bir alan ontoloji modülü, belirli bir alana ait terminolojik bilgiyi modeller.

Yerel ontoloji modülü: Yerel modüller iki tür bilgiyi saklarlar;

-lokalleştirilmiş alan kavramları: bir alan modeli, bir uygulama alanındaki tüm bilgi kaynaklarınca paylaşılan terminolojik bilgiyi gösterir. Fakat, bazı bilgi kaynakları alan kavramlarının daha da özelleştirilmesini isteyebilirler. Bir yerel ontoloji modülü l , l ile ilişkili bilgi kaynaklarınca paylaşılan yerel modül kavramlarını saklar. Bir yerel modül kavramı, bir alan kavramının özelleştirilmesi veya yeniden adlandırılması yolu ile tanımlanır.

-kavramlara ait örnek verileri: alan modülleri belirli bir alana ait terminolojik bilgiyi modellerler. Bunun yanısıra, yerel modüllerde bilgiye ait örnek verileri de tutulur. Bilgi kaynakları kendi veritabanlarındaki örnekleri yerel modüllerdeki örneklerle eşleyerek, ontolojilerle entegre olurlar.



Şekil 1: Modüller ve aralarındaki hiyerarşik ilişkiler.

Ontoloji tabanlı entegrasyon sistemleri için ontolojilerin geliştirilmesi çok önemlidir. Bu nedenle, bu sürecin kolaylaştırılması için önerilmiş metodolojiler ve araçlara ihtiyaç duyulmaktadır. Ne yazık ki, şimdiye kadar önerilmiş entegrasyon sistemlerinin birçoğu metodoloji ve araç desteği sunmamaktadır [1].

Bu çalışmada, modüler bir global ontoloji geliştirmeye yönelik bir metodoloji ve bu metodolojiyi temel alan bir araç sunulmaktadır. Bildirinin geri kalanı şu şekilde düzenlenmiştir; takip eden bölümde entegrasyon-tabanlı modül geliştirme metodolojisi anlatılmaktadır. Bölüm 3 bu metodolojiyi temel alarak gerçekleştirilmiş ontoloji geliştirme aracını sunmaktadır. Son bölüm sonuçları ve bir kısmı üzerinde çalışılmaya başlanmış gelecek çalışmaları özetlemektedir.

2. Modül-tabanlı Ontoloji Geliştirme Metodolojisi

Global ontolojinin modülerleşmesi ve bu modüllerin farklı kişilerce modellenmesi, özelleşmiş bir metodoloji ihtiyacı oluşturmuştur. Modül-tabanlı ontoloji geliştirme metodolojisi bu ihtiyacı karşılamak üzere sunulmuştur ve aşağıda özetlenen adımları içermektedir;

2.1. Modüllerin ve Kapsamlarının Belirlenmesi

Ontoloji geliştirici, teknik ve finansal alan uzmanlarına, kullanıcılara, varolan veritabanlarına ve dökümanlara başvurarak modülleri ve modüllerin kapsamını belirler.

2.2. Modüllerin Oluşturulması

Modüllerin oluşturulması, modüllerin kavramsallaştırılmasını ve formalizasyonunu içerir. Kavramsallaştırma safhasında, modüle ait terimler tiplerine göre gruplandırılır, temel ve yardımcı kavram hiyerarşileri oluşturulur.

Formalizasyon safhası, ontolojinin formal bir dilde gösterilmesidir. Bu çalışmada, özel bir formalizasyon dilinin kullanılması savunulmamaktadır. Bunun yerine, herhangi bir dil ile birlikte kullanılabilir genel bir dil uzantısı sunulmaktadır. Bu yapıtaş seti veri entegrasyonu açısından önem taşımaktadır. Varolan ontoloji dilleri, ontolojilerin bir kişi tarafından geliştirileceğini varsayar ve ontoloji geliştiriciler dışındaki ontoloji kullanıcıları açısından önemli olabilecek yapıtaşlarından yoksundurlar. Formalizasyon safhasında, sunulan yapıtaşlarının kullanılması ontoloji kullanıcılarının önceden-tanımlı modül kavramlarını anlamalarını kolaylaştırmaktadır. Bu dil uzantısının elemanları aşağıda listelenmektedir;

Soyut nitelikler: Soyut nitelikler, örneklendirilemeyen fakat alt nitelikleri yaratılabilen niteliklerdir. Örneğin, *boyut* niteliği örneklendirilemez fakat hiçbir soyut olmayan üç alt niteliğe (*en*, *boy*, *yükseklik*) sahiptir. Bir soyut niteliğe ait değer alanı tanımlanamaz. Bu niteliğin veri entegrasyonu açısından önemi, yerel ontoloji geliştiricileri, örneklendirilmesi istenmeyen alan ontoloji nitelikleri hakkında bilgilendirmesidir.

Temel nitelikler: Felsefe alanında, temel nitelik bir nesnenin mutlaka sahip olması gereken nitelik olarak tanımlanır. Buna karşın, yardımcı nitelik bir nesnenin sahip olabileceği fakat sahip olması zorunlu olmayan niteliğidir [3]. Bu çalışma kapsamında, temel nitelik spesifik bir uygulama için bir nesnenin sahip olması gerekli özellikleri olarak tanımlanmıştır. Bir nesnenin temel niteliklerine ait değerlerin (temel nitelik soyut bir nitelik olarak tanımlanmadığı sürece) belirtmeleri zorunludur. Yerel ontoloji geliştiriciler, ontolojilerindeki örneklere ait temel nitelik değerlerinin belirtilmesinden sorumludurlar.

Ölçü birimleri: Birçok niceliğin ölçülmesi, ifade edilmesi ve aynı tür niceliklerin birbiriyle karşılaştırılması için birimlerin kullanılması zorunludur. Önerilen metodolojiye göre, bir nitelik için en fazla bir birim belirtilebilir. Böylelikle,

modüller arası ölçeklendirme uyumsuzlukları [1] önlenmiş olur.

Somutlaştırıcı nitelikler: Bir ontoloji örneğine *i* ait somutlaştırıcı bir nitelik *p*, bir dizi değere sahiptir. Bu değerlerin herbiri *v_i* için, *p* niteliğinin değeri *v_i* olan en az bir tane gerçek dünyada tanımlı somut örnek bulunmaktadır. Somutlaştırıcı nitelikler, soyut ontoloji örneklerinin birden fazla somut örnekle eşleşmesini sağlarlar. Somut her bir örnek için, soyut bir ontoloji örneği yaratılmadığı için, ontolojiye ait örneksel verinin boyutu belirli sınırlarda tutulabilir. Ontoloji araçlarının ölçeklenebilirlik problemleri gözönünde tutulduğunda, somutlaştırıcı niteliklerin kullanımı büyük ölçekli uygulamalar için kritik olmaktadır.

Somutlaştırıcı niteliklerin kullanılmasının etkileri, birden fazla somutlaştırıcı niteliğe sahip örneklerde daha iyi gözlemlenebilmektedir. Bir soyut örneğin, herbiri *m* adet farklı değer alabilen, *n* tane somutlaştırıcı niteliği olduğu varsayıldığında, karşılık gelen somut örneklerin sayısı *mⁿ* değerine ulaşmaktadır.

2.3. Modüllerin Bakımı

Modüllerin bakımı yerel ve alan ontoloji geliştiriciler arasında yoğun bir iletişim gerektirmektedir. Yerel ontoloji geliştirici, ilgili alan ontolojilerindeki değişiklikleri takip etmeli ve ontolojisini güncel tutmalıdır. Bir yerel ontoloji geliştirici, herhangi bir alan ontolojisinde veya farklı bir yerel ontolojide değişiklik yapamaz. Bu nedenle, alan ontolojisindeki eksiklikler hakkında alan ontoloji geliştiriciyi uyarmalıdır. Buna karşılık, alan ontoloji geliştirici yerel geliştiricilerin isteklerine yanıt vermelidir. Farklı modüller arasında entegrasyonu korumak için metodoloji kapsamında aşağıdaki kurallar tanımlanmıştır;

- Bir ontoloji modülü, yalnızca o modülün geliştiricisi tarafından değiştirilebilir.
- Yerel ve alan ontolojilerinin artımlı olarak modifikasyonunda bir kısıtlama yoktur.
- Örneklendirilmiş veya diğer ontolojilerce referans gösterilmiş alan ontolojisi kavramlarının modifikasyonu engellenmiştir.

3. ONTECO: Veri-Entegrasyonu Temelli Modüler Ontoloji Geliştirme Aracı

Bu çalışmada, modül-tabanlı metodolojinin uygulanmasını kolaylaştırmak için bir ontoloji geliştirme aracı sunulmaktadır. Araç, <http://aegont.bilmuh.ege.edu.tr/onteco/> adresinden indirilebilir. Aracın dört temel bileşeni bulunmaktadır;

Ontoloji Sunucusu ontoloji deposundaki tüm ontolojileri yükler, bu ontolojiler üzerinde çıkarsama yapar ve ontoloji istemcisinden gelen sorguları yanıtlar.

Ontoloji İstemcisi herhangi bir modülü yaratmak veya düzenlemek amacıyla kullanılır. Modül kaydedildikten sonra, değişiklikler sunucuya gönderilir. Aracın ilk versiyonu uzak veya birden fazla ontoloji istemcisini desteklememektedir.

Fakat istemci/sunucu mimarisi, gelişmeye açık bir yapı sunmaktadır.

Ontoloji Deposu ontoloji dosyalarını ve ontolojilere ait metaveriyi saklamaktadır. Ontolojik metaveri OMV [4] modeli kullanılarak yaratılmıştır. Ontolojiye ait metaveriler isim, uri, tip (taban, meta-alan, alan, yerel), tanım, fiziksel patika, import edilen ontoloji listesi, doğal dil, ontoloji dili, geliştirici, geliştirilme ve güncellenme tarihlerini kapsamaktadır. Bu metaveri aynı zamanda ontoloji hiyerarşisinin oluşturulması için kullanılmaktadır.

Benzerlik Tespit Bileşeni ontoloji geliştiricilerin farklı ontolojilere ait kaynaklar arasındaki olası eşlemeleri belirlemelerine yardımcı olurlar. Bu bileşen, farklı isim-tabanlı teknikleri (normalizasyon, altdizgi testleri, edit-distance, Yahoo veritabanının Türkçe Hearst örüntüleri ile aranması) paralel kompozisyon [5] ile bir araya getirerek kullanır.

3.1. Araç Özelliklerine Genel Bakış

Aracın desteklediği özellikler aşağıdaki gibi listelenebilir;

metodoloji-temelli ontoloji geliştirme desteği, modül-tabanlı metodolojinin uygulanmasını kolaylaştırır, metodolojiye aykırı operasyonları engelleyerek *Tablo 1*'deki kısıtlara uygun ontoloji geliştirilmesini sağlar. Ayrıca, araçta metodoloji kapsamında önerilen dil uzantısına ait yapıtaşları da desteklenmektedir (*Şekil 1*).

ontoloji modülleri arasında hiyerarşik ilişkiler, modüller arası "imports" ilişkisi üzerine kuruludur. Bu ağaç topolosindeki ağda, alt seviyeli modüller, yukarıdaki modüllere ait kavramları yeniden kullanırlar. *Şekil 2*, örnek bir ontoloji ağacını göstermektedir.

daha küçük ontoloji modülleri ile hızlı ve etkin geliştirme, farklı spesifikasyonlara sahip istemcilerle çalışabilme olanağı sunar.

farklı modüllere ait kaynaklar arasında metodoloji-uyumlu ilişkiler kurulmasını sağlar. Ontoloji geliştiricilerin şema (kavramlar arası ilişkiler) ve örnek seviyesindeki (kavram örnekleri arasındaki ilişkiler) kaynakları ilişkilendirmesini kolaylaştırır. *Şekil 3*'te, araç yeni oluşturulan bir kavramla eşleşmesi olası kavramları listelemekte ve seçilen eşlemenin metodoloji uyumlu olmadığı kullanıcıya bildirmektedir.

modüller arası aksiyom taşıma desteği sağlar. Bu işlem farklı opsiyonlar ve onların kombinasyonları ile gerçekleştirilebilir. Bu opsiyonlar *sınıf tanımını taşıma*, *sınıfın niteliklerini taşıma*, *sınıfın örneklerini taşıma* ve *sınıfın kaynak ontolojideki sınıflarla ilişkilerini silme* olarak sıralanabilir. *Şekil 4*'te modüller arası aksiyom taşıma düğmesi ve işlem penceresi görülmektedir.

hızlı ve kolay örnek girişi ile yapılandırılmış, satırlar ve sütünlardan oluşan çizelge formundaki örnek verilerinin ontolojiye aktarılmasını sağlar (*Şekil 4*'te gösterilen *DS2Onto* düğmesi ile). Yaygın elektronik çizelge formatları ve CSV formatı desteklenmektedir.



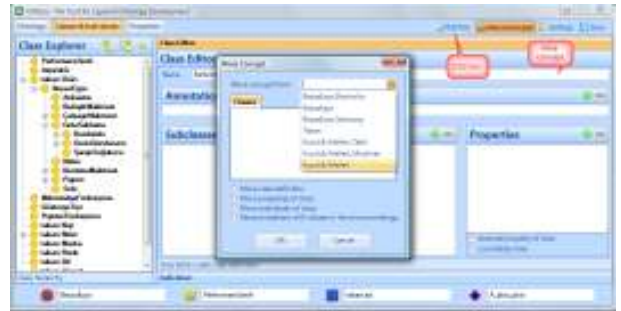
Şekil 1: Metodoloji kapsamında önerilen dil uzantısına ait yapıtaşları.



Şekil 2: Örnek bir ontoloji ağacı.



Şekil 3: Farklı modüllere ait kaynaklar arasında metodoloji-uyumlu ilişkiler kurulması.



Şekil 4: Modüller arası kavram taşıma ve hızlı örnek girişi işlemleri.

Operasyon	Alan Ontolojisi O_d	Yerel Ontoloji O_l
C sınıfına C_{sub} alt sınıfını ekle	eğer C_{sub} sınıfı O_d içerisinde tanımlanmışsa ve C sınıfı O_d veya O_{imp} içerisinde tanımlanmışsa	eğer C_{sub} sınıfı O_l içerisinde tanımlanmışsa ve C sınıfı O_l veya O_{imp} içerisinde tanımlanmışsa
C sınıfına C_{super} üst sınıfını ekle	eğer C_{super} sınıfı O_d veya O_{imp} içerisinde tanımlanmışsa ve C sınıfı O_d içerisinde tanımlanmışsa	eğer C_{super} sınıfı O_l veya O_{imp} içerisinde tanımlanmışsa ve C sınıfı O_l içerisinde tanımlanmışsa
P niteliğine P_{sub} alt niteliğini ekle	eğer P_{sub} niteliği O_d içerisinde tanımlanmışsa ve P niteliği O_d veya O_{imp} içerisinde tanımlanmışsa	eğer P_{sub} niteliği O_l içerisinde tanımlanmışsa ve P niteliği O_l veya O_{imp} içerisinde tanımlanmışsa
P niteliğine P_{super} üst niteliğini ekle	eğer P_{super} niteliği O_d veya O_{imp} içerisinde tanımlanmışsa ve P niteliği O_d içerisinde tanımlanmışsa	eğer P_{super} niteliği O_l veya O_{imp} içerisinde tanımlanmışsa ve P niteliği O_l içerisinde tanımlanmışsa
C sınıfına örnek ekle	eğer C sınıfı O_d veya O_{imp} içerisinde tanımlanmışsa	eğer C sınıfı O_l veya O_{imp} içerisinde tanımlanmışsa
P niteliğine önalın (veya arđalan) ekle	eğer P niteliği O_d içerisinde tanımlanmışsa ve P için #domains (#ranges) ≤ 1 ise	eğer P niteliği O_l içerisinde tanımlanmışsa ve P için #domains (#ranges) ≤ 1 ise
C sınıfına P temel niteliđi ekle	eğer C sınıfı O_d içerisinde tanımlanmışsa	eğer C sınıfı O_l içerisinde tanımlanmışsa
C sınıfına P somutlařtırıcı niteliđi ekle	eğer C sınıfı ve P niteliđi O_d içerisinde tanımlanmışsa	eğer C sınıfı ve P niteliđi O_l içerisinde tanımlanmışsa
P niteliđine u birimini ekle	eğer P niteliđi O_d içerisinde tanımlanmışsa ve P için #units ≤ 1 ise	eğer P niteliđi O_l içerisinde tanımlanmışsa ve P için #units ≤ 1 ise
I örneđine ait P niteliđine deđer ata	eđer I örneđi O_d içerisinde tanımlanmışsa ve P bir soyut nitelik deđerilse	eđer I örneđi O_l içerisinde tanımlanmışsa ve P bir soyut nitelik deđerilse
üçlü sil	eđer üçlü O_d içerisinde tanımlı ise ve çıkarsanmış bir üçlü deđerilse	eđer üçlü O_l içerisinde tanımlı ise ve çıkarsanmış bir üçlü deđerilse

Tablo 1: Metodoloji kısıtları.

4. Sonular ve Gelecek alıřma

Bu alıřmada, veri entegrasyonuna ynelik olarak kullanılan global ontolojilerin modler olarak geliřtirilmesi iin bir metodoloji nerilmektedir. Bu metodolojinin uygulanmasını kolaylařtırmak iin metodoloji kısıtlarını ve gereksinimlerini karřılayacak bir ara geliřtirilmiřtir. Ara desteđi, metodolojiyi diđerlerinden ayıran en nemli farklardan birisidir. İleriye dnk olarak, aracın varolan bir ontoloji ile metodoloji arasındaki uyumsuzlukları bulacak bir bileřenle geniřletilmesi planlanmaktadır. Bylelikle, varolan bir ontolojinin, metodoloji uyumlu bir ontoloji ailesine entegrasyonu kolaylařacaktır.

5. Kaynaka

- [1] Wache, H., Vgele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hbner, S.: Ontology-based integration of information - a survey of existing approaches. *In: IJCAI-01 Workshop: Ontologies and Information Sharing*. (2001), 108-117.
- [2] Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. *In: International Semantic Web Conference*. (2004), 289-303.
- [3] Robertson, T.: Essential vs. accidental properties. *Technical report, Stanford Encyclopedia of Philosophy* (2008)
- [4] Hartmann, J., Sure, Y., Haase, P., Palma, R., del Carmen Surez-Figueroa, M.: Omv - ontology metadata vocabulary. *In: Ontology Patterns for the Semantic Web Workshop*. (2005)
- [5] Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE). (2007)