

İNTERNET ORTAMINDA ORTAKLAŞA İŞ YÜRÜTMİYİ DESTEKLEYEN YENİ BİR NESNE MODELİ VE PROGRAMLAMA ARAYÜZÜ

Güray YILMAZ
Hava Harp Okulu
Bilgisayar Mühendisliği Bölümü
Yeşilyurt, 34807, İstanbul
g.yilmaz@hho.edu.tr

Nadia ERDOĞAN
İstanbul Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü
Ayazağa, 80626, İstanbul
erdogan@cs.itu.edu.tr

ÖZET

Dağıtılmış sistemler bilgisayar ağları üzerinden sistem kaynaklarının ve bilginin paylaşımını sağlarlar. Bu sistemleri çekici yapan önemli tasarım konusu, dağıtım ile ilgili olan tüm ayrıntıların kullanıcılara saydam olarak gerçekleşiyor olmasıdır. Fakat ne yazık ki, hali hazırdaki geniş alana dağıtılmış sistemler, bilginin dağıtımı ile ilgili konuların gerçekleştirilmesini gizleme konusunda başarılı değildir. Yerel sistemler için geliştirilen çözümler geniş alanlı dağıtılmış sistemlere ölçeklenememektedirler. Bu çalışmada Dağıtılmış Bileşik Nesne (Distributed Composite Object-DCO) olarak adlandırılan yeni bir nesne modeli tanıtılmaktadır. Aynı zamanda, Java programlama dili ile dağıtılmış uygulamalar geliştiren yazılımcıların kullanımı için uniform bir arayüz sağlayan DCOBE adını verdiğimiz bir yazılım katmanını tasarım ve gerçekleştirme ayrıntıları da sunulmaktadır. Bu katman kullanıcılara bir çok uygulama için gerekli olan haberleşme, nesnelerin dağıtılması, bölünme, tutarlılık, uygulamaların dinamik olarak yüklenmesi gibi temel mekanizmaları sağlar.

1. GİRİŞ

Dağıtılmış sistemler bilgisayar ağları üzerinde sistem kaynaklarının ve bilginin paylaşımını sağlarlar. Bu sistemleri çekici kılan en önemli tasarım konusu, dağıtım ile ilgili tüm ayrıntıların kullanıcılara saydam olarak gerçekleşiyor olmasıdır. Ne yazık ki, sistem kaynaklarının kullanıcılar arasında saydam

olarak paylaşımına ve yönetimine izin veren genel amaçlı geniş alana yayılmış dağıtılmış sistemler mevcut değildir. Var olan bazı sistemler de kısıtlı bazı kolaylıklar sağlamaktadırlar. İnternet üzerinden veri paylaşımı gibi geniş alanlı uygulamaların geliştirilmesi için genellikle çok fazla emek sarf etmek gerekmektedir. Bu durum, üzerinde çalışılan işletim sistemi ve diğer sistem çözümlerinden gerekli haberleşme desteğinin sağlanamamış olmasından kaynaklanmaktadır [5].

Bu gün için, iyi bilinen bir çok dağıtılmış sistem *uzak-nesne* modelini benimsemişlerdir. Bu modelde, nesne tek bir alana yerleştirilir ve bu nesnenin kullanıcılarına bir vekil arayüzü üzerinden nesneye saydam erişim olanağı sağlanır. En iyi durumda, nesne, kullanıcılarını haberdar etmeksizin bir başka alana göç edebilir. Fakat uzak-nesne modelinin genelde ölçeklenebilme eksikliğinden kaynaklanan bir çok dezavantajları bulunmaktadır. Uzak-nesne modeli kullanıcılara farklı nesne çağrı yöntemlerini gerçekleyecek bir destek sunmamaktadır. Halbuki, sisteme kopyalama ve dağıtım gibi bazı teknikler eklenmek istendiğinde bu destek oldukça gerekli olmaktadır [1].

Bu çalışmada, uzak-nesne modeline alternatif olarak, yeni bir nesne modeli, **dağıtılmış bileşik nesne** (DCO) önerilmektedir. Düşüncemize göre, internet ortamında dağıtılmış, işbirliğiyle yürütülen uygulamalar genellikle *iri ya da orta parçalı nesnelere* üzerinde çalışmaktadırlar. Fakat, bu tip uygulamaların kullanıcıları genelde bu nesnelere küçük bir parçasıyla ilgilenmektedirler. Bundan dolayı, eğer biz büyük bir nesneyi *daha küçük alt nesnelere (finer-grained subobjects-SO)* parçalayabilirsek, erişilecek ve farklı alanlar arasında aktarılacak olan verinin miktarı azalacağından dolayı, çalışan uygulamanın başarımı yükselecektir.

DCO'nun tasarımının arkasında yatan temel fikir, nesnenin çok sayıda farklı alan üzerinde fiziksel olarak dağıtılmış olmasıdır. Hali hazırda kullanılmakta olan DCOM [3], DCE [2] ve CORBA [6] gibi bilinen sistemler, dağıtılmış nesneyi tek bir makine üzerinde çalışan ve belki diğer makineler üzerinde kopyaları da bulunan bir nesne olarak görmektedirler. Bu nesne diğer alanlarda bulunan kullanıcılara *vekil* aracılığı ile yerel bir nesne olarak sunulmaktadır

2. SİSTEMİN YAPISI

Bu çalışmada, *Dağıtılmış Bileşik Nesne Tabanlı Ortam (DCOBE)* sisteminin genel tasarımını ve Java ortamında dağıtılmış bileşik nesne gerçekleştirilmesine ilişkin olarak yaptığımız ilk çalışmaları anlatılmaktadır. DCOBE sistemi dağıtılmış bileşik nesnelere desteklemek için tasarlanmış Java paketlerinden oluşmaktadır. DCOBE'ü özellikle internet ortamında yayılmış, birlikte yürütülen uygulamalar geliştirecek olan kullanıcılar için Java programlama dili [4] üzerinden bir uniform arayüz sağlamak amacıyla tasarlanmıştır. Bu yazılım bir çok uygulama için gerekli olan nesne bileşimi,

kopyalama, tutarlılığın sağlanması, uygulamanın dinamik olarak istenilen alana yüklemesi ve benzeri temel mekanizmalar sunmaktadır.

DCOBE'ü kullanarak, programcılar uygulamalarını merkezi bir ortamda çalıştırmış gibi geliştirebilmektedirler. Daha sonra, kod üzerinde herhangi bir değişiklik yapılmaksızın, uygulama dağıtılmış ortam için konfigüre edilebilmektedir. Bu, dinamik bağlama, senkronizasyon fonksiyonları ve tutarlılık protokolleri gibi kolaylıklar sağlayan kod parçalarının otomatik olarak nesnelere eklenmesi suretiyle yapılmaktadır.

DCOBE sisteminin temel fonksiyonları şunlardır:

- DCOBE özellikle, internet üzerine yayılmış dağıtılmış nesne-tabanlı uygulamaların geliştirilmesi için tasarlanmış olan bir yazılımdır.
- DCOBE, DCO nesne modelini esas almaktadır. DCO'nun tasarımındaki temel fikir, bir DCO'nun alt nesnelere vasıtasıyla fiziksel olarak dağıtılmış olmasıdır. Diğer bir deyişle, nesnenin durumu, aynı anda farklı alanlar üzerine yerleştirilmiş bir çok alt nesnenin durumlarından oluşmaktadır.
- Dinamik yayılma. Uygulamalar istekli düğümlere o uygulamaya sahiplik eden düğümden dinamik olarak yayılır. Böylece kullanıcılar uygulamayı çalıştırmaya başlamadan önce makineye yüklemek zorunda kalmazlar.
- Dinamik uyarılma. Büyük ölçekli geniş alana yayılmış olan bir uygulama dinamik olarak yeni bileşenlerin eklenmesi ve var olanların çıkarılabilmesine olanak tanımalıdır. DCOBE, mevcut durumda nesneye bağlantı sağlamış olan istekçileri etkilemeksizin, DCO'ya yeni bir alt nesne

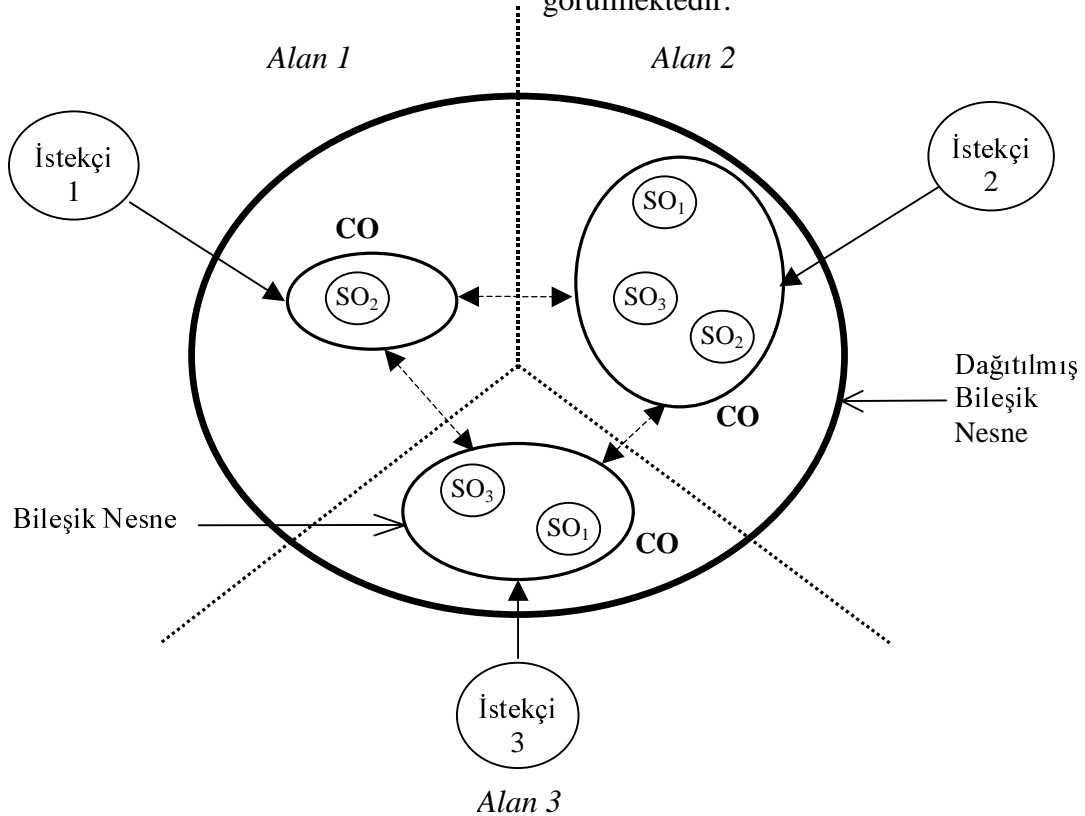
eklenmesine, var olanların çıkarılmasına ya da değiştirilmesine imkan tanır.

- Kopyalama. Dağıtılmış ortamda birlikte yürütülen uygulamalar için en önemli gereksinimlerden biri de kopyalamadır. DCOBE, DCO'nun paylaşılan alt nesnelere birlikte iş yürütmekte olan farklı alanlar üzerinde kopyalanmasına izin verir. Böylece, dağıtılmış nesnelere yerel olarak yürütülür ve iletim gecikmesi azaltılır.
- Saydamlık. DCOBE sisteminde, dağıtılmış ortamdaki çok sayıda kullanıcı tarafından ortaklaşa yürütülen bir uygulama merkezi bir yapıda çalışıyormuş gibi geliştirilebilir. Dağıtım, senkronizasyon ve tutarlılık uygulama kodundan bağımsız olarak programlanır. Bu aynı zamanda mevcut merkezi uygulamaların dağıtılmış ortamda çalışabilmelerini de mümkün kılar.

3. DAĞITILMIŞ BİLEŞİK NESNE

Dağıtılmış bileşik nesne (DCO) modeli, nesnenin gerçekleyicisinin nesne durumunu çoklu alt nesnelere (SO) arasında dağıtabilme düşüncesine dayanmaktadır. Bu, ana nesnenin durumunun parçalı olmasını gerektirir. Bu alt nesnelere farklı alanlara yerleştirilebilirler, kopyalanabilirler ve hatta kendileri de birer bileşik nesne olabilirler.

DCO modelinde, bir çok alt nesne bir araya gelir ve bileşik nesneyi oluşturur. Bir bileşik nesne tekil bir alan üzerinde kendi alt nesnelere ile birlikte yaratılır. Sonra, bu bileşik nesne istek üzerine alt nesnelere bazıları ile birlikte bir çok diğer alan üzerinde kopyalanabilir. Çok sayıda alan üzerine yayılmış bulunan bu bileşik nesnelere dağıtılmış bileşik nesneyi oluşturur. Şekil-1'de, üç farklı alan üzerine yayılmış bulunan bir dağıtılmış bileşik nesne görülmektedir.



Şekil-1: Üç alt nesnesi ile birlikte üç adres alanı üzerine yayılmış bir dağıtılmış bileşik nesne.

Şekil-1'deki yapı incelendiğinde, *CO* bileşik nesnesinin öncelikle *Alan 2* üzerinde *SO₁*, *SO₂* ve *SO₃* alt nesneleri ile birlikte yaratılmış olduğu görülür. *CO* bileşik nesnesi daha sonra *Alan 1*'de yalnızca *SO₁* alt nesnesi ve *Alan 3*'de *SO₁* ve *SO₃* alt nesneleri ile birlikte kopyalanmıştır.

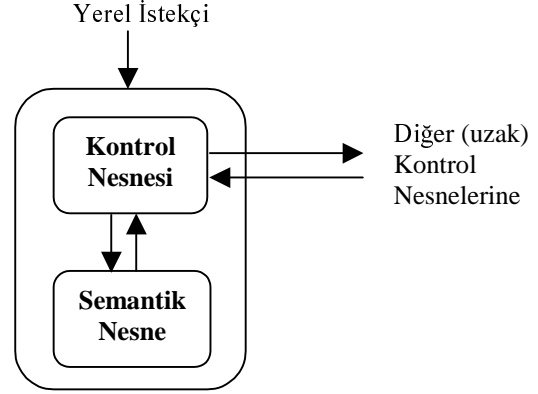
CO bileşik nesnesini oluşturan alt nesneleri ve bunların yapılarını yalnızca bu nesnenin tasarımcısı bilmektedir ve tüm diğer istekçiler, nesne hakkında kendilerine sunulan istekçi arayüzünün haricinde bir başka bilgiye sahip değildirler. Bu istekçiler aynı zamanda hangi alt nesnenin kendi alanlarına yüklendiği, yükleme işleminin nasıl yapıldığı, dinamik bağlama, senkronizasyon ve tutarlılığın sağlanması gibi daha bir çok konunun nasıl çözümlendiğini bilmemektedirler. Tüm bu konular istekçilere saydam olarak alt tabakadaki DCOBE sistemi tarafından gerçekleştirilir.

Bir dağıtılmış bileşik nesnenin iki görünüşü vardır: İstekçilerin bakış açısına göre; tekil paylaşılan bir nesnedir. Farklı adres alanlarında bulunan çok sayıdaki istekçi nesneleri tarafından paylaşılmaktadır. Bu nesneye programcı-tanımlı bir arayüz üzerinden erişilir. Onun bileşenleri ve özellikle de nasıl dağıtıldıkları görünmez. Tasarımcısının bakış açısına göre; bir dağıtılmış bileşik nesne, birlikte çalışan bir alt nesnelere kümesinden oluşmaktadır. Bu yapıdaki her bir alt nesne, merkezi yapıdaki gibi, temel bir nesnedir. Diğer bir deyişle, her bir alt nesne merkezi bir gösterilime sahiptir.

3.1. Bir alt nesnenin yapısı

Yukarıda da sözünü ettiğimiz gibi, bir dağıtılmış bileşik nesne bir kaç alt nesnenin bir araya gelmesinden oluşmuştur. Bir alt nesne tekil bir adres alanına yerleştirilir ve dağıtılmış bileşik nesnenin tutarlı bir görüntüsünü

oluşturmak için diğer adres alanındaki alt nesnelere haberleşir. Bu amaçla her bir alt nesne Şekil-2'de gösterildiği gibi, iki nesnenin birleşiminden oluşmaktadır. Bunlar; *semantik nesne* ve *kontrol nesnesi*'dir. Bir alt nesne semantik nesnesinin sunmuş olduğu arayüzün aynısını gerçekleştirir.



Şekil-2: Alt nesnenin yapısı.

Şekil-2'deki semantik nesne, dağıtılmış bileşik nesnenin gerçek semantiğinin bir bölümünü gerçekleyen bir yerel parçadır. Nesne tasarımcısı, dağıtılmış bileşik nesneyi oluşturan farklı semantik nesnelere her biri için sınıf nesnelere gerçekleştirmekle yükümlüdür. Diğer bir deyişle, semantik nesne, programcı tarafından gerçekleştirilecek olan tek nesnedir. Diğer iki nesne (kontrol nesnesi ve alt nesne) ise, semantik nesnesinin arayüz tanımlayıcısını kullanarak *nesne üretici* tarafından otomatik olarak üretilecektir.

Kontrol nesnesi, ilgili semantik nesnesinin arayüz tanımlaması temel alınarak üretilen bir nesnedir. İki tip çağrı isteğini kontrol eder; yerel istekçiden gelen istek çağrılar ve ağ üzerinden diğer kontrol nesnelere üzerinden gelen istek çağrılar. Kontrol nesnesinin yerel istekçiye sunmuş olduğu arayüz de, semantik nesnenin arayüzü ile aynıdır. Ek olarak, diğer alanlara yerleşmiş bulunan kontrol nesnelere için, tutarlılık ve eşzamanlılık kontrolü ile ilgili

metotları içeren bir arayüz daha sunar. Bu arayüz istekçilerden gizlidir.

Kontrol nesnesi gelen çağrı isteklerini karşılamak ile sorumludur. Dağıtılmış bileşik nesnenin genel durumu, onun çok sayıdaki semantik nesnelere durumlarından meydana geldiğinden ve bu semantik nesnelere de kullanışlılık, hata toleransı ve başarımlar gibi nedenlerden dolayı çok sayıdaki alan üzerinde kopyalanmış durumdadır. Bu yüzden, kontrol nesnesi eşzamanlılık ve tutarlılık gibi konuları da çözümlenmiş durumundadır.

4. NESNE ÜRETME

Programcı uygulamalarını, ne ek bir dil desteği, ne de bir sistem desteği olmaksızın, Java programlama dilinde merkezi bir uygulama yazıyormuş gibi geliştirir. Öncelikle, her bir farklı semantik sınıfa ilişkin sınıf tanımlamaları yazar, ikinci adımda ise, uygulama dağıtılmış ortam için konfigüre edilir. Bu adımda, semantik nesnenin arayüz tanımlaması kullanılarak öncelikle alt nesneye ait sınıf dosyası üretilmektedir. Uygulama merkezi yapıya uygun olarak geliştirildiği için, senkronizasyon ve tutarlılık ile ilgili, herhangi bir kod içermemektedir. Bu yüzden, konfigürasyon aşamasının İkinci adımında, kontrol nesnesine ilişkin sınıf dosyası üretilir. Bu sınıf yapısı dağıtım, kopyalama ve tutarlılığın sağlanması gibi ayrıntıları içeren metotlardan oluşmaktadır.

5. SONUÇ

Bu çalışmada, **dağıtılmış bileşik nesne** olarak adlandırdığımız yeni bir nesne modeli önerilmiş ve bu modeli destekleyen sistem yazılımını DCOBE ait tasarım ve gerçekleştirme ayrıntıları anlatılmıştır. DCOBE yazılımı, internet ortamına yayılmış dağıtılmış uygulamalar için sistem desteği sağlamaktadır. Dağıtılmış bileşik nesne modeli, nesne

durumunun, farklı alanlara yayılmış alt nesnelere üzerine kopyalanma suretiyle dağıtıldığı, hatta alt nesnelere kendileri de birer bileşik nesne olabildiği bir modeldir.

DCOBE ile uygulamalar istekte bulunulan alanlar üzerine, uygulamaya sahiplik eden alandan, dinamik olarak yayılır. Böylece, çalışmaya başlamadan önce uygulamanın o alanda yüklü olması gerekmez. Ayrıca, geniş alana yayılmış dağıtılmış uygulamalar üzerinde, o uygulamayı kullanmakta olan istekçileri etkilemeksizin, yeni bileşenlerin eklenmesi, var olanların çıkarılması ya da değişiklik yapılması gibi durumları dinamik olarak gerçekleştirmesi gerektiği öngörüldüğünde; DCOBE, dağıtılmış bileşik nesneye, alt nesnelere eklenmesi, çıkarılması ve değiştirilmesi durumları, o nesneyi hali hazırda kullanmakta olan istekçilere saydam olarak gerçekleştirmektedir.

DCOBE dağıtılmış bileşik nesneyi oluşturan alt nesnelere çok sayıdaki alan üzerinde kopyalanmasına izin vererek, çağrılarını yerel olarak yürütülmesini sağlamakta ve böylece iletim gecikmelerini azaltmaktadır.

DCOBE’da, dağıtılmış uygulamalar merkezi bir uygulama geliştiriliyormuş gibi, Java dili ve DCOBE’un sunduğu birkaç ek komut kümesi kullanılarak yazılabilmektedir. Dağıtım, senkronizasyon ve tutarlılık konuları, uygulamayı oluşturan koddan ayrı olarak DCOBE tarafından gerçekleştirilmektedir. Bu özellik, aynı zamanda, mevcut merkezi uygulamaların dağıtılmış ortama kolayca uyarlanabilmesini de sağlamaktadır.

6. REFERANSLAR

- [1] B.C. Neuman, Scale in Distributed Systems, In *Readings in Distributed Computing Systems*, IEEE Computer Society Press, 1994.
- [2] D. Rosenberry, D. Kennedy and G. Fisher, *Understanding DCE*, O'Reilly and Associates, Sebastopol , Calif., 1992.
- [3] G. Eddon and H. Eddon, *Inside Distributed COM*, Microsoft Press, Redmond, WA, 1998
- [4] J. Gosling, B. Joy and G.Steele, *The Java language Specification*, Addison-Wesley Developers Press, Sunsoft Java Series, 1996.
- [5] M. Van Steen, P. Homburg, and A. Tanenbaum, Globe: A Wide-Area Distributed System, *IEEE Concurrency*, 7(1), pp. 70-78, Jan.-Mar., 1999.
- [6] Object Management Group, *The Common Object Request Broker: Architecture and Specification. Revision 2.3.1*. OMG Document formal/99-10-07, Object Management Group, Framingham, MA, Oct., 1999.