

ED Model Yapıtaşı Haberleşme Altyapısı

Aysun Sancar Yılmaz, Betül Baydemir Çankaya, Hande Doğan Köseoğlu

REHİS-EHGYM, Aselsan A.Ş. , Ankara
{asancar,baydemir,hdogan}@aselsan.com.tr

Özet. Elektronik Destek (ED) Model Yapıtaşı, Radar Elektronik Harp projelerinde ihtiyaç olan veri yapısı ortaklığı, kontrol işlemci haberleşme ve eğitim simülasyonu ihtiyaçlarını karşılamak için geliştirilen bir yapıtaşıdır. Yapıtaşı, sistemde bulunan diğer yazılımlarla konfigüre edilebilir olarak OSGI, ATA ve CORBA teknolojileri kullanarak haberleşebilmektedir. Bu çalışmada ED Model Yapıtaşı haberleşme mimarisi aktarılacaktır.

Anahtar Kelimeler: Radar Elektronik Destek, OSGI, Yazılım Mimarisi, CORBA, ATA, Tasarım Kalıpları, Kullanıcı Arayüzü

1 Giriş

ED Model Yapıtaşı, radar elektronik destek kullanıcı arayüzü yazılımlarının temel işlevleri olan yayınların tespiti, yayın parametrelerinin bulunması için yapılan detaylı tespit, tespit edilen parametrelerin kaydı, yayın parametrelerine göre yayının kimlik bilgisini belirlemek için kullanılacak veri dosyalarının yönetimi, yayınlar üzerinde parametre bazlı filtreleme, yayınlara alarm üretme, sistem kalibrasyon işlemleri ve eğitim simülasyon ihtiyaçlarını kontrol işlemci yazılımı ile haberleşerek karşılayan bir yapıtaşıdır.

Benzer ED projelerinde Kİ birimiyle haberleşme ve senaryo yönetimlerinin ortaklanmasını sağlama, ortak bileşenlerle daha hızlı ürün çıkarma çalışmaları kapsamında alan analizi çalışmaları yapılmış, bu kapsamda ED Model Yapıtaşı ortak bir katman olarak ED projelerinde kullanılmak üzere geliştirilmeye başlanmıştır.

Yapıtaşı servis tabanlı mimariye (Service Oriented Architecture) uygun olarak geliştirilmiştir. Yapıtaşı birbirleri ile haberleşen servislerden oluşmaktadır. Bu servislerin aranmasını, bulunmasını ve kullanılmasını sağlayan servis platformu olarak OSGİ (Open Services Gateway Initiative)[1] kullanılmıştır. OSGİ platformu çeşitli hazır yönetim servisleri sunmaktadır. Her bir servis bu yönetim servislerini kullanarak servis kütüğüne kendi sunduğu servisleri kaydeder. Bu servisleri kullanacak diğer servisler yine yönetim servislerini kullanarak istedikleri servisleri servis kütüğünden sorgular, bağlanır ve kullanır. Servis kullanımı nesnelerin metotlarını çağırarak şeklinde olur.

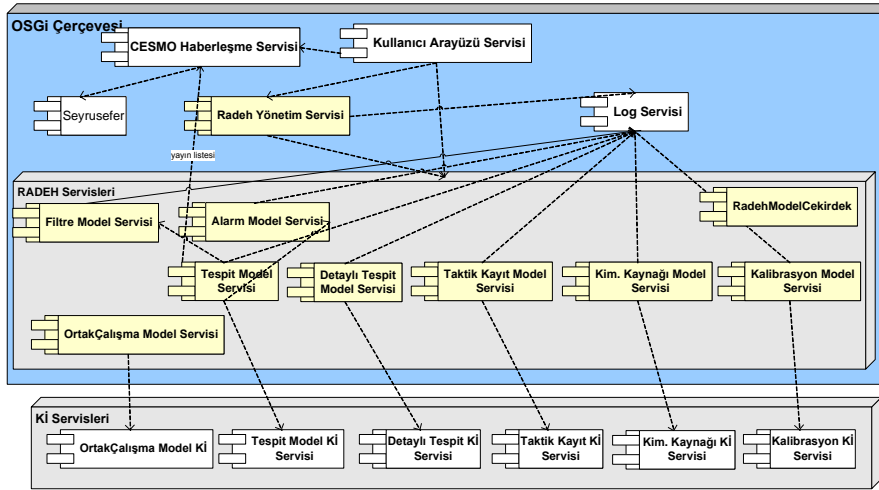
Yapıtaşı içinde geliştirilen bileşenlerin birbiri ile OSGİ altyapısı üzerinden haberleşmesi yanı sıra veri akışını gerçekleştirdiği kontrol işlemci yazılımı ile haberleşme ihtiyacı bulunmaktadır. ED Model Yapıtaşı bu haberleşmeyi OSGİ, ATA (Arayüz Tanımlama Aracı) [3] ya da CORBA (Common Object Request Broker

Architecture) [2] aracılığı ile gerçekleştirir. Bu makalede ED Model Yapıtışının haberleşme mimarisi aktarılacaktır.

2 ED Model Yapıtışının Kapsamı

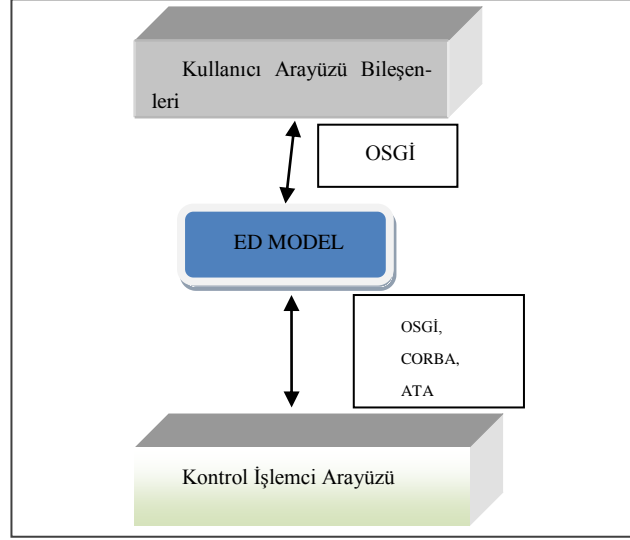
ED Model Yapıtışı, OSGİ ortamında kendini ortama kaydeden java ‘plugin’ lerinden oluşmaktadır. Bu yapıtışını kullanacak uygulamalar OSGİ ortamından servisleri çekerek kullanıma alır. Yapıtışının temel görevleri radar elektronik harp destek projelerinde yer alan tespit, teşhis, kalibrasyon, kayıt, kimliklendirme kaynağı, alarmlama, filtreleme ve ortak çalışma mesajlaşması (ED sisteminin diğer sistemlerle haberleşmesi) gibi genel ed işlemlerini gerçeklemek (görev modu), bu yeteneklere ek olarak da ortam simülasyonu yapmaktır. (eğitim modu)

Şekil 1’de ED Model servislerinin yer aldığı projenin çalışma ortamı gösterilmektedir. OSGİ çerçevesinde bulunan kullanıcı arayüzü servisleri ED model servislerini ortamdan alarak kullanırlar. Radeh Yönetim Servisi mod geçişlerini yöneten servis olarak bir üst katmanda konuşlandırılmıştır. Bütün servis dış arayüzü “RadehModelCekirdek” servisi üstünden sunulmaktadır. Bu sayede bağımlılıklar tek bir elden yönetilir.



Şekil 1. ED Model Yapıtışı Çalışma Ortamı Genel Görünümü

Kİ servisleri diye gösterilen kısım ise ED Model Yapıtışının haberleştiği kontrol işlemci arayüzünü belirtmektedir. Her bir kontrol işlemci arayüzü için bir bağlantı kurulmaktadır. Şekil 2’de ED Model Yapıtışının hem kendini kullanan Kullanıcı Arayüzü hem de kontrol işlemci yazılımı ile kurduğu genel haberleşme yapısı gösterilmektedir.



Şekil 2. ED Model Yapıtışı Genel Haberleşme Yapısı

3 ED Model Yapıtışında Kullanılan İlk Haberleşme Altyapısı (FAZ 1)

ED Model Yapıtışı elektronik harp projelerinde yer alan ortak kontrol işlemci (Kİ) haberleşme ve senaryo ihtiyaçlarının ortaklanması amacıyla gerçekleştirilmeye başlamıştır. Proje ilk defa hayata geçirildiğinde Kİ haberleşmesi CORBA altyapısı ile yapılacağı için ilk sürüm olarak CORBA altyapısı kullanılmıştır. Makalede Faz 1 olarak bahsedilen altyapı ED Modelin CORBA kullanılarak yapılan tasarımıdır.

3.1 CORBA

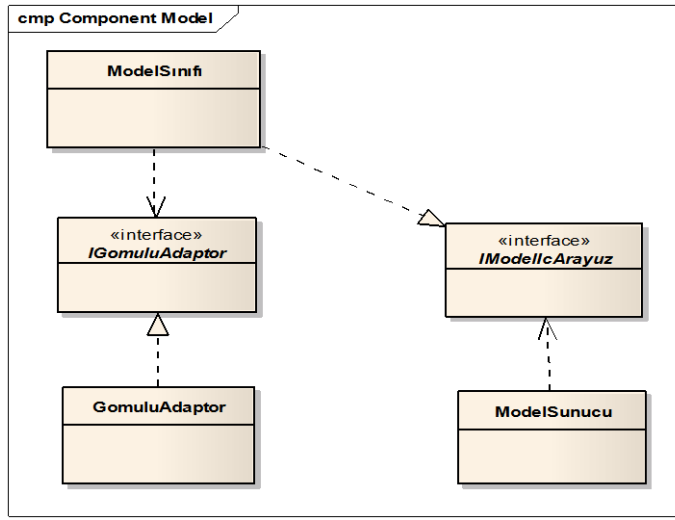
CORBA , Object Management Group (OMG) tarafından oluşturulan farklı platformlardaki sistemlerin haberleşmesi için geliştirilmiş ortak bir dildir. İşletim sisteminden, dilden ve donanımdan bağımsız bir şekilde iletişimi sağlar. Bu özellikleri sayesinde farklı ortamlarda koşabilen ve farklı dillerde geliştirilmiş olabilen projeye özel kontrol işlemci yazılımlarının dış arayüz ile haberleşme senaryolarında CORBA kullanılmıştır. CORBA, 'IDL'[4] (interface definition language) adı verilen mesajların yazılı olduğu dosyaları kullanır. Bu dosyalar 'idl' derleyicileri tarafından derlenip kodlamada kullanılacak kaynak kodlar oluşturulur. ED Model Yapıtışı java ortamında geliştirildiği için 'idl' derlemesi sonucu oluşan java dosyaları ilk aşamada kullanıma alınmıştır.

Bu ‘idl’ dosyalarının diğeri bir özelliđi de Faz 2’de gerçeklenen ATA ve OSGİ altyapısı için ortak kullanım dili sađlayacak olmasıdır.

3.2 İlk Tasarım

Bu mimaride, her bir Kİ bađlantısı CORBA üzerinden bađlantı kurup yönetir. Yapıtasının haberleşme seçeneđi mevcut kontrol işlemci yazılımlarının CORBA bađlantısını destekleyecek şekilde tasarlanmıştır.

Bir ED Model servisinin Kİ yazılımı ile haberleşmek için kullandıđı tasarım kalıbı Şekil 3’te gösterilmektedir.



Şekil 3. ED Model Yapıtası Faz 1 Haberleşme Tasarım Kalıbı

Bu tasarım kalıbına göre Kİ yazılımından mesaj alan sınıf “ModelSunucu” sınıfıdır ve servis sınıfı olan “ModelSınıfı” sınıfını “IModelIcArayuz” arayüz sınıfı üzerinden bilmektedir, doğrudan bir bađlantı kurulmamıştır. Benzer şekilde “ModelSınıfı” sınıfı Kİ yazılımına mesaj göndermek amacıyla “IGomuluAdaptor” arayüzünü kullanmakta, bu arayüzü gerçekleyen “GomuluAdaptor” sınıfına doğrudan bir erişimi bulunmaktadır.

4 ED Model Haberleşme Altyapısı (FAZ 2)

ED Model Yapıtası bileşeninin farklı elektronik harp projelerinde farklı Kİ bileşenleri ile haberleşme ihtiyacı doğmuştur. Makalede Faz 2 olarak bahsedilen ED Model Yapıtası , OSGİ ve ATA arayüzleri ile Kİ haberleşmesini de gerçekleyecek şekilde genişletilmiştir.

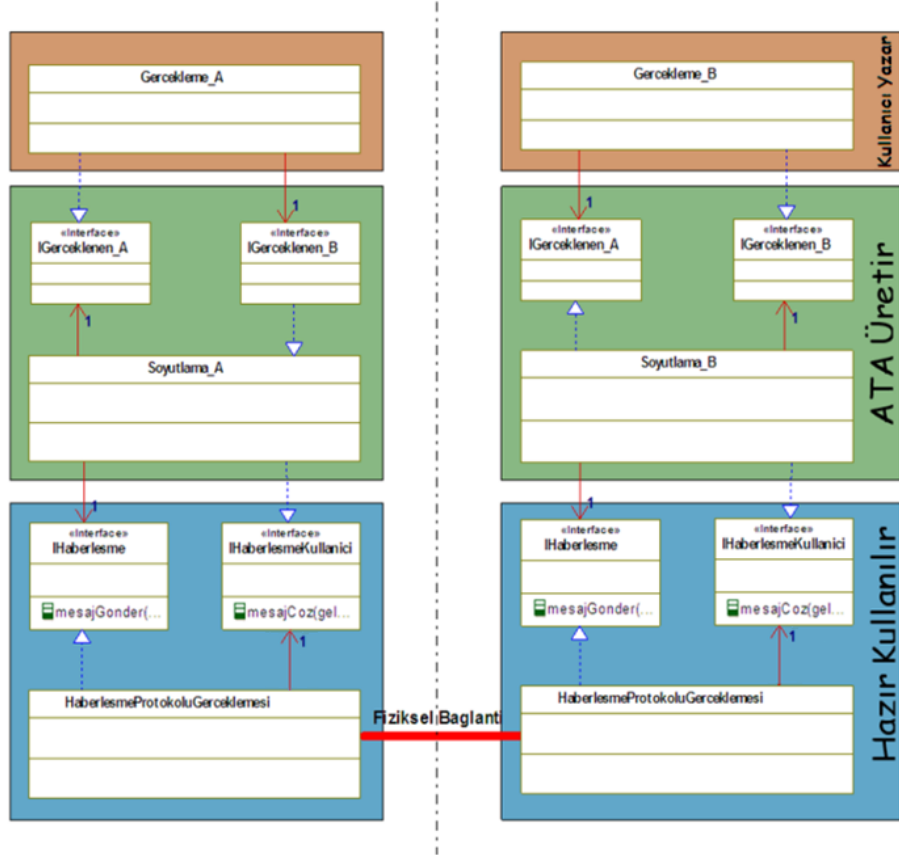
4.1 OSGİ

OSGİ, yeniden kullanılabilirliđi destekleyen modüler yapıda bir mimari ile java uygulamaları geliřtirmeyi sađlayan bir çerçevedir.

Ed Model Yapıtařı, OSGİ çerçevesi kullanılarak geliřtirilmiřtir. ED projelerinde, zamanla deđiřen istekler ve ortamlar sonucunda kontrol iřlemci yazılımları gml kartlardan Windows ortamına alınmıř, yazılım geliřtirici ekibin de tecrbelerine dayanarak OSGİ kullanılarak geliřtirilmeye bařlanmıřtır. Kİ yazılımının OSGİ'de geliřtirilmesi sonucu, bu yazılımla haberleřecek ED Model Yapıtařı yazılımının da OSGİ arayz olması istenmiřtir.

4.2 ATA

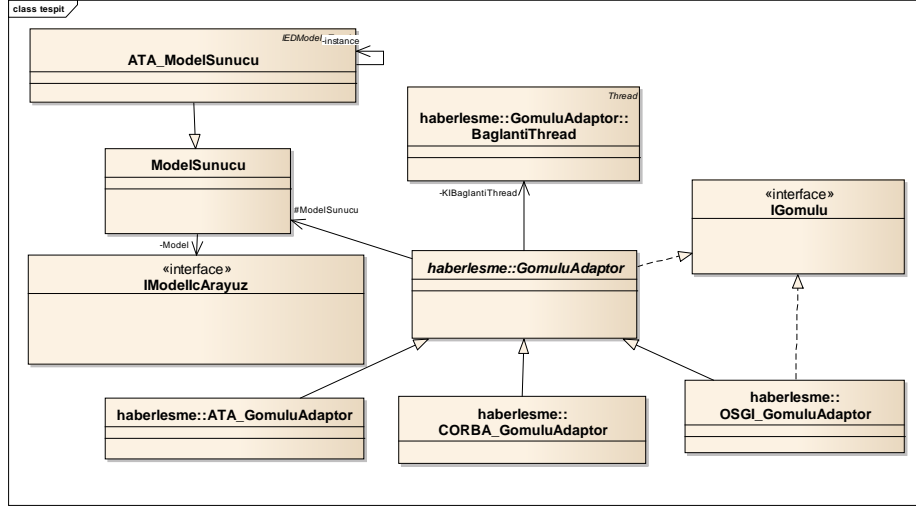
Arayz Tanımlama Aracı (ATA), dađıtık birimler zerinde alıřan yazılım bileřenlerinin arayzlerini bir arayz modeli ile tanımlayan, tanımlanan arayz modeli zerinden, ara katman kaynak kodu, arayz dokmanı ve arayz test etmek iin alıřtırılabilir bir test aracı reten Eclipse tabanlı bir aratır. ATA, tanımlanmıř bir ATA arayz modeli zerinden ara katman kodu, alıřtırılabilir bir test aracı ve yazılım arayz tasarım tanımlı dokmanı retebilmektedir. Oluřturulan ara katman kodu RPC[5] (Remote Procedure Call) mantıđı ile alıřan bir katmandır. Oluřturulan kod haberleřme protokolnden bađımsızdır. retilen ara katmanın kullanım yntemi Őekil 4'te verilmiřtir.



Şekil 4. ATA Kullanım Yöntemi

Üretilen ara katmanın kullanılması için kullanıcının tek yapması gereken dış arayüzden gelen mesajları ve senaryoları gerçeklemesidir.

4.3 Haberleşme Tasarımı



Şekil 5. ED Model Yapıtaşı Faz 2 Haberleşme Tasarım Kalıbı

Kİ yazılımı ile haberleşen ED Model servislerinde kullanılan ortak bir haberleşme tasarım kalıbı bulunmaktadır. Bu tasarım kalıbı Şekil 5’te verilmektedir. Bu tasarım kalıbında “GomuluAdaptor” sınıfı sunucu nesnesine mesaj göndermek amacıyla kullanılacak olan sınıftır. “GomuluAdaptor” sınıfı, gönderilecek mesajları içeren “IGomuluAdaptor” arayüz sınıfı üzerinden model sınıfı tarafından kullanılacaktır. Sunucu nesnesinden gelen mesajları alacak sınıf ise “ModelSunucu” sınıfıdır. “ModelSunucu” sınıfı “IModelArayuz” arayüz sınıfı üzerinden model sınıfına mesajları aktarır.

GomuluAdaptor sınıfından türeyen 3 adet sınıf bulunmaktadır. Bu sınıflarda ATA, CORBA ve OSGI haberleşmesi için özelleşmiş kodlar bulunmaktadır. ED Model servisleri konfigürasyondan ayarlanan haberleşme tipine göre ilgili “GomuluAdaptor” nesnesi yaratılır. OSGI ve CORBA haberleşmesi için aynı nesne türleri kullanılmıştır. CORBA tarafından kullanılan sınıflar ortaklık sağlaması açısından OSGI haberleşmesinde de tercih edilmiştir. ATA haberleşmesi için gelen ve giden mesajlar ayrı bir sınıf kümesi için tanımlanmış, bu sebeple sunucu tarafı ATA için yeniden yazılmıştır. “ATA_ModelSunucu” sınıfına gelen mesajlar CORBA sınıf tiplerine çevrilerek bundan sonra olan akış aynı olacak şekilde tasarım yapılmıştır.

ATA haberleşmesi için yapılan bu değişikliğin ana sebebi, haberleşme sağlanan Kİ bileşenin ATA sınıf tiplerini kullanmasıdır. OSGI haberleşmesi için OSGI’de geliştirilen Kİ yazılımı da ‘idl’den üretilen CORBA sınıflarını kullanmaktadır.

5 Sonuç

ED Model Yapıtışı, 2008 yılından beri deęişen ekiplerle beraber farklı ED projelerinde kullanıma alınmıştır. Projelerdeki ihtiyaçlar ve talepler doğrultusunda kendini son teknolojilere uygun olarak güncellemiş, şu an aktif olarak 5 projede kullanıma alınmıştır. Bu projelerden biri ATA arayüzünü, 3 tanesi de OSGİ arayüzünü kullanmaktadır. Bir projede de CORBA ile idamesi devam etmektedir. Kİ ile haberleşmeler ‘idl’ arayüzü kullanılarak ortaklanmaktadır. Kullandığımız tasarım kalıbı sayesinde yeni bir mesaj eklendiğinde ‘idl’ deęişikliği yapılır, bu ‘idl’ deęişikliği ATA için ayrıca derlenir, ‘idl’ için üretilen kodlar ve ATA için üretilmiş yine ‘idl’den derlenen ‘jar’ projeye eklenerek haberleşmede deęişen kısım hızlıca eklenebilmektedir. Bütün türetilmeler ‘idl’ üzerinden olduđu için yapılacak hatalar minimum düzeye indirgenmiştir.

Kullandığımız mimari ED Model Yapıtışı için yeni bir altyapı ile Kİ haberleşmesini destekleyecek durumdadır. “GömülüAdaptor” sınıfından türeyen yeni bir sınıf yazmak hem mevcut yapıyı bozmayacak, hem de yeni bir haberleşme entegrasyonunu sağlayacaktır.

Kaynaklar

1. www.osgi.org, OSGI Web Sayfası
2. www.corba.org, CORBA Web Sayfası
3. B. Kekeç,O. Karadeniz, “Arayüz Tanımlama Aracı”, UYMS,2013.
4. www.omg.org, Object Management Group Web Sayfası
5. A.D. Birrell, B.J. Nelson, ”Implementing remote procedure calls”, ACM Transactions on Computer Systems 2, 1984.