

Gömülü Yazılımlarda Yeniden Kullanım Zorlukları ve ASELSAN Çözümleri

Barış İYİDİR¹, Adnan KALAY²

^{1,2} ASELSAN A.Ş. Savunma Sistem Teknolojileri Grubu

¹ biyidir@aselsan.com.tr, ² akalay@aselsan.com.tr

Özetçe

Günümüzde yazılım projelerin teslim sürelerinin kısa olması yazılım geliştirme aşamasında üretilen ürünlerin yeniden kullanımını zorunlu kılmaktadır. Bu ürünlerin yeniden kullanımı konusunda birçok yöntem önerilse de bu yöntemler genel olarak masaüstü uygulamalar temel alınarak oluşturulmuştur. Masaüstü uygulamalar ile gömülü yazılımların kısıtları arasında farklılıklar bulunması, masaüstü uygulamalarda kullanılan yöntemlerin hiç değiştirilmeden gömülü yazılım geliştirme süreçlerinde kullanılmasını engellemektedir[1][2].

1. Giriş

Yazılım projelerinde yeniden kullanım terimi ilk olarak kaynak kodun yeniden kullanımını akla getirirse de yeniden kullanım olarak yazılım geliştirme sürecinde üretilen bütün yazılım ürünlerinin yeniden kullanımının ele alınmasının daha faydalı olacağı düşünülmektedir. Bu ürünler içerisinde kaynak kodun yanı sıra, yazılım gerekleri, yazılım tasarımı ve yazılım testleri de sayılabilir. Ancak bu ürünlerin yazılım projelerinde yeniden kullanımın sağlanabilmesi yeniden kullanım odaklı bir sürecin uygulanması ile mümkün olmaktadır. Bu anlamda yeniden kullanım odaklı projeler ile diğer projeler arasında süreç anlamında ciddi farklılıklar vardır[3]. Yeniden kullanım odaklı süreçlerde, geliştirilen projelerde ortak kullanılacak yazılım parçalarının bulunması ve bu parçaların geliştirilmesi önemli bir yer tutmaktadır[4].

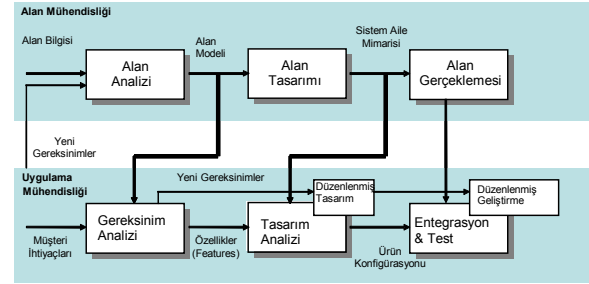
Bu bildiri kapsamında ilk önce yeniden kullanım süreci konusunda kısa bir bilgi verildikten sonra gömülü yazılımlarda yeniden kullanımın zorlukları belirtilecek ve ASELSAN'da gerçekleştirilen gömülü yazılım projelerinde karşılaşılan problemler ile bu problemlerin çözümleri kapsamında yapılan çalışmalardan bahsedilecektir. Sonuç olarak mevcut durumun tespiti yapılacak ve yeniden kullanımı arttırmak için yapılabilecek çalışmalar özetlenecektir.

2. Yeniden Kullanım Süreci

Yazılım geliştirme sürecinde sistematik yeniden kullanımı sağlamak için, geliştirilecek projeler arasındaki ortaklıkların belirlenmesi gerekmektedir. Bu sayede farklı projeler arasında ortak kullanılacak bileşenler belirlenerek, bu bileşenlerin bütün projelere hizmet verecek şekilde oluşturulması sağlanmaktadır.

Yeniden kullanım odaklı projelerde önerilen "Yazılım Ürün Hattı" süreci Şekil 1'de verilmiştir. Projeler arasındaki ortak bileşenlerin bulunması ve gerçekleşmesi işlemi için yeniden kullanım odaklı süreçlerde alan mühendisliği adı verilen bir faz tanımlanmıştır[4]. Alan mühendisliği fazının çıktıları arasında ortak kullanılacak bileşenler, bu bileşenlere ait

gereksinim, tasarım, test dokümanları ve bu bileşenlerin entegrasyonunun nasıl yapılacağını tanımlayan referans mimari sayılabilir. Alan mühendisliği fazında oluşturulan yazılım ürünleri, uygulama mühendisliği fazında kullanılarak proje ihtiyaçlarını karşılayacak yazılımlar oluşturulmaktadır.



Şekil 1 Yazılım Ürün Hattı Süreci[4]

Yeniden kullanım, yazılım geliştirme sürecinde en çok yeniden kullanımı sağlanan ürün olan kaynak kodun yeniden kullanım biçimine göre, kara kutu yeniden kullanım, gri kutu yeniden kullanım ve beyaz kutu yeniden kullanım olarak sınıflandırılmaktadır. Bu sınıflandırma yeniden kullanım sonucunda elde edilebilecek faydaları da göstermektedir. Kara kutu yeniden kullanımda, kullanıcı kaynak kodun içeriğini görememekte ve değiştirememektedir. Bu yeniden kullanım biçimi en çok tavsiye edilen yöntem olsa da kaynak kodun görülebilmesi kullanılan kaynak kodun işlevlerinin anlaşılması konusunda problem yaratmaktadır. Beyaz kutu yeniden kullanımda ise kullanıcı kaynak kodu görebilmekte ve değiştirebilmektedir. Kaynak kodun değiştirilmesi nedeniyle kaynak kodun dışındaki yazılım ürünlerinin de uygun şekilde güncellenmesini gerekmektedir. Bu durum yeniden kullanım esnasında ek iş gücü getirmektedir. Bu nedenle beyaz kutu yeniden kullanım zorunlu durumlar dışında tavsiye edilmemektedir. Gri kutu yeniden kullanımda ise kaynak kod görülebilmekte ancak değiştirilememektedir. Gri kutu yeniden kullanım birçok projede beyaz ve kara kutu yeniden kullanımın getirdiği problemlere çözüm bulması nedeniyle tercih edilmektedir.

3. Gömülü Yazılımlarda Yeniden Kullanım Kısıtları

Gömülü sistemler cep telefonu, mp3 çalar gibi küçük cihazların yanı sıra uçuş kontrol sistemleri, silah kontrol sistemleri gibi karmaşık birçok sistemi içine almaktadır. Bu tür sistemlerde çoğunlukla gerçek zamanlılık istenir de bulunmaktadır. Ayrıca birçok gömülü sistem görev-kritik (mission-critical) sistem sınıfına girmektedir. Bu tür sistemlerin güvenilirlik, tahmin edilebilir davranış ve düşük

tepkisi süresi gibi kalite gereklileri bulunmaktadır. Ayrıca bazı gömülü sistemlerde kullanılan donanımlar gereği güç ve hafıza kullanımı bakımından önemli kısıtlar bulunabilmektedir. Bu nedenlerle masaüstü yazılımlarda kullanılan EJB, .NET gibi teknolojiler aynı şekilde gömülü yazılımlarda kullanılamamaktadır. Bildirinin bu bölümünde gömülü yazılımlara özel kısıtlar alt başlıklar şeklinde incelenecek ve ASELSAN SST grubu yazılım mühendisliği bölümünde gömülü yazılım geliştirme sürecinde karşılaşılan problemlerden bahsedilecektir.

3.1. Bileşen Entegrasyon Teknolojileri

Bileşen tanımı konusunda literatürde bir çok tanım bulunsa da genel kabul görmüş tanıma göre bileşen, tek başına çalışabilen ve çalışma zamanında diğer bileşenler ile entegre olabilen yazılım parçaları olarak tanımlanmaktadır[5]. Ancak gömülü sistemlerde kullanılan gerçek zamanlı işletim sistemleri genel olarak çalışma zamanı entegrasyonuna izin vermemekte, bu tür entegrasyona izin veren ortamlarda ise performans ve güvenilirlik problemleri ortaya çıkmaktadır. Bu nedenle genel kabul görmüş bileşen tanımı gömülü yazılımlar için tam olarak kullanılamamaktadır.

Gömülü yazılımlarda masaüstü uygulamalardaki bileşen tanımına tam olarak uyulmasa da, bileşenlerin yeniden kullanımını konusundaki temel prensipler gömülü alanda da uygulanabilmektedir. Bu anlamda gömülü yazılımlarda çalışma zamanı entegrasyon yerine derleme zamanı veya model seviyesi entegrasyon tercih edilmektedir[1]. Çalışma zamanı entegrasyon teknolojilerinin gömülü yazılımlarda çok fazla kullanılamaması nedeniyle gömülü yazılımlarda kaynak kodun yeniden kullanımı masaüstü uygulamalara göre daha yaygın görülmektedir. Ancak kaynak kodun yeniden derlenmesi yeni bir çalıştırılabilir yazılım anlamına geldiğinden bileşen testlerinin tekrar yapılmasını gerektirmektedir. Bunun yanı sıra kaynak kodun "beyaz kutu" kullanımını kaynak kod dışındaki yazılım ürünlerinin sistematik yeniden kullanımını zorlaştırmaktadır. Bu nedenle bileşenlerin yeniden kullanımı için bağlanma(link) zamanı entegrasyonu sağlayan kütüphane şeklinde yeniden kullanım önerilmekte ve bu sayede yeni sürüm çıkarılmaması sağlanarak bileşenin gerek, tasarım ve test dokümanlarının da yeniden kullanımının sağlanabileceği düşünülmektedir.

ASELSAN SST grubu yazılım mühendisliğince geliştirilen gömülü sistemlerde kaynak kodun beyaz kutu yöntemi ile yeniden kullanılması durumunda sistematik bir yeniden kullanımın sağlanamadığı gözlenmiştir. Bileşenlerde proje ihtiyaçları doğrultusunda kontrolsüz değişiklik yapılması sonucunda sürüm takibinin yapılamamasının bu duruma neden olduğu düşünülmektedir. ASELSAN SST grubu yazılım mühendisliği bölümünce uygulanan çözüm yöntemleri bildirinin 4.2 alt başlığında verilmiştir.

3.2. Gömülü Yazılım Geliştirme Süreci

Yeniden kullanımın sağlanabilmesi, geliştirilecek projeler arasındaki ortaklıkların belirlenmesine bağlıdır. Ancak genel olarak gömülü sistemler özel bir işi yapmak üzere geliştirildiklerinden bu projelerde kullanılan yazılımlar birbirinden bağımsız olarak ele alınmakta ve ortaklıkların yakalanması mümkün olmamaktadır. Ayrıca klasik (traditional) yazılım geliştirme süreçleri yazılım gereklilerinin belirlenmesi, yazılım tasarımı, yazılım gerçekleştirilmesi ve yazılım test adımlarını içerdiklerinden yazılımlar arası

ortaklıkların yakalanamamasına neden olmaktadır. Yeniden kullanım odaklı süreçlerde ise yazılımlar arasındaki ortaklıkların bulunması sonucunda bu adımların çıktılarının yeniden oluşturulması yerine bu ürünlerin hepsinin yeniden kullanımını hedeflendiğinden, klasik yazılım geliştirme süreçlerine göre farklılık göstermektedir. Nokia, Philips gibi firmaların gömülü alanda yeniden kullanım odaklı bir süreç işleterek başarıya ulaşmaları gömülü sistemler alanında da proje aileleri oluşturularak ortaklıkların yakalanabileceğini göstermiştir[7][6]. ASELSAN SST grubunda geliştirilen silah sistemleri ele alındığında bu sistemlerin genel olarak donanım yeniden kullanımı ile gerçekleştirildiği görülmektedir. Silah sistemlerinin farklılıkları ise bu sistemlerin yazılımlarına eklenen özellikler ile sağlanmaktadır. Silah sistemleri yazılımlarında sistematik yeniden kullanımı sağlamak amacıyla bu yazılımların ortaklıklarının ve değişkenliklerinin belirlenmesi gerekmektedir. ASELSAN SST grubu yazılım mühendisliği bölümünce belirtilen problemleri çözmek için uygulanan yöntemler bildirinin 4.1 alt başlığında verilmiştir

3.3. Performans Kısıtları

Gömülü yazılımlar genel olarak gerçek zaman kısıtları olan yazılımlardır. Bu kısıtlar sistemin bütünü değerlendirilerek ortaya çıkmaktadır. Bu nedenle bileşenlerin tek başına sağladığı performanstan çok bileşenlerin entegrasyonundan sonra elde edilen performans önem kazanmaktadır. Bileşen entegrasyonu sonrasında oluşan problemler değerlendirilerek bileşenler içerisinde projelere özel performans artırıcı değişiklikler yapılması gerekebilmektedir. Bu durum da bileşenin değiştirilmeden yeniden kullanımını engellemektedir. Bileşenin kaynak kodunda değişiklik yapılması ve yeniden derlenmesi nedeniyle bileşen için daha önce hazırlanan yazılım gereklileri, yazılım test raporları gibi yazılım ürünlerinin de yeniden oluşturulması gerekmektedir. Performans kısıtlarını ortadan kaldırmak için uygulanan yöntemler bildirinin 4.3 alt başlığında verilmiştir.

3.4. Yazılım Test Kısıtları

Literatürde getirisi en fazla olan yazılım geliştirme yönteminin yeniden kullanım olduğunun vurgulanmasına karşın, en fazla götürüye sahip yöntemin de yeniden kullanım olduğu belirtilmiştir. Buna göre çok yönlü test edilen ve hata içermediği doğrulanmış bileşenlerin yeniden kullanımı büyük getiriler sağlarken, yeterli şekilde doğrulanmayan bileşenlerin yeniden kullanımının faydadan çok zarar getirdiği görülmüştür[5]. Özellikle görev-kritik ve güvenlik-kritik yazılımlar söz konusu olduğunda bu durum daha fazla önem kazanmaktadır. Gömülü yazılımları masaüstü yazılımlardan ayıran en önemli özellik yüksek güvenilirlik gereğidir. Bir masaüstü bilgisayarda uygulamanın hatalı çalışması durumunda bu hatanın maliyeti genel olarak bilgi kaybı ile sonuçlanmasına rağmen, bir silah sistemindeki yazılımın hatalı çalışması veya çalışmaması ciddi mal kayıpları ve yaralanmalara, hatta can kaybına neden olabilir. Güvenlik-kritik yazılımların doğrulanma sürecinin önemi geçmişte yaşanan felaketlerde edinilen kötü tecrübeler sonunda anlaşılmıştır. 1996 yılında Ariane 5 uzay roketinin navigasyon sisteminde Ariane 4'te kullanılmış bir yazılım parçasının gerektiği gibi doğrulanmadan yeniden kullanılması sonucunda tarihin en maliyetli yazılım felaketi meydana gelmiştir [8][9]. Bu tür yazılımlarda ortaya çıkacak hataların maliyetlerinin çok yüksek olması, yeniden kullanılacak bileşenlerin

kullanıma alınmadan önce detaylı olarak test edilmesini ve doğrulanmasını zorunlu kullmaktadır.

Gömülü sistemler için geliştirilen yazılımların genel olarak çalışması hedeflenen donanım dışındaki donanımlar üzerinde çalıştırılmaması nedeniyle gömülü yazılımlar için test işlemleri genellikle örnek sistem üzerinde gerçekleştirilmektedir. Ancak örnek sistemin nihai hale gelmesi zaman aldığından bileşen testlerinin gerçekleştirilmesi ve bulunan hataların düzeltilmesi konusunda problemler yaşanabilmektedir. Bu bölümde verilen kısıtları aşmak için uygulanan yöntemler bildirinin 4.4 alt başlığında verilmiştir.

4. ASELSAN'da Kullanılan Yeniden Kullanım Yöntemi

ASELSAN SST grubu yazılım mühendisliği bölümü tarafından gömülü yazılımlarda sistematik yeniden kullanımın artırılması amacıyla çalışmalar yapılmıştır. Bu bölüm kapsamında 3. bölümde verilen kısıtların ve ASELSAN SST grubu yazılım mühendisliği tarafından geliştirilen projelerde karşılaşılan problemlerin giderilmesine yönelik ASELSAN SST grubu yazılım mühendisliği bölümünce gerçekleştirilen çalışmalar anlatılacaktır.

4.1. ASELSAN'ın Gömülü Yazılım Geliştirme Sürecine Bakışı

ASELSAN SST grubunda geliştirilen gömülü yazılımlar genel olarak silah sistemi atış kontrol yazılımlarıdır. ASELSAN SST grubu yazılım mühendisliği bölümünde silah sistemi atış kontrol yazılımı geliştiren ekiplerin yapılandırılması, sistemde kullanılan mühimmatın özelliklerine göre şekillendirilmiştir. Bu yapılanma ile farklı mühimmat grupları üzerinde uzmanlaşmış özel ekiplerin oluşturulması ve bu uzmanlık ile hızlı ve kaliteli ürün çıkarılması hedeflenmiştir. Bu yaklaşım ile ekipler içerisinde yeniden kullanımın üst düzeye çıkarılmasına rağmen, silah sistemlerindeki bütün ortaklıkların ve değişkenliklerin yakalanamaması nedeniyle bütün silah sistemlerinde değiştirilmeden ortak kullanılacak bileşenlerin oluşturulmasında problemlerle karşılaşmıştır. ASELSAN SST grubunda geliştirilen silah sistemleri projeleri incelendiğinde sistemde kullanılan mühimmatların değişmesine rağmen, sistemi oluşturan birçok donanımın ve sistemin fonksiyonel işlevlerinin değişmediği görülmektedir. Bu anlamda bu donanımları kontrol etmek ve fonksiyonel işlevleri yerine getirmek için kullanılan yazılım parçalarının da yeniden kullanılabilirliği düşünülmüştür. Bu yaklaşım mühimmat özelliklerinin ortaklaşmasının yanı sıra birçok sistem özelliklerine göre de ortaklama yapılabileceğini ve ortak amaç için geliştirilen yazılım bileşenlerinin hiç değiştirilmeden bütün silah sistemleri proje ekipleri tarafından kullanılabilirliğini göstermektedir.

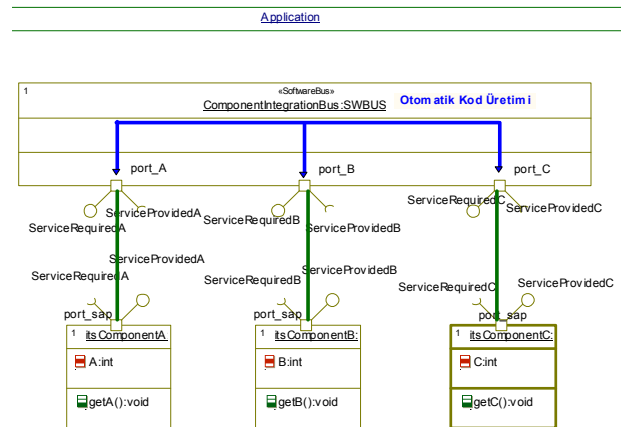
ASELSAN'da geliştirilen gömülü yazılım projelerinde elde edilen tecrübe sonucunda sistematik yeniden kullanımın ancak değişkenlik ve ortaklık analizi yapılması ile gerçekleştirilebileceği düşünülmüştür. Ancak bütün gömülü yazılım alanı için ortaklık ve değişkenlik analizi yapılmasının gömülü yazılımların içinde barındırdığı değişkenlik, güvenilirlik, gerçek zamanlılık gibi kalite faktörleri göz önüne alındığında maliyet açısından uygun olmayacağı, ortak özellikleri fazla olan proje aileleri için bir alan analizi çalışması başlatılmasının maliyet etkin bir çözüm olacağı

değerlendirilmiştir. Bu kapsamda ASELSAN SST grubu yazılım mühendisliğince üniversiteden akademik destek alınarak silah sistemleri atış kontrol yazılımları alanında alan analizi çalışması gerçekleştirilmiştir. Bu çalışma sonucunda silah sistemleri atış kontrol yazılımları için özellik modeli oluşturulmuş, silah sistemi atış kontrol yazılımlarında ortak kullanılacak bileşenlerin ve bileşen entegrasyon yönteminin belirtildiği referans mimari ortaya konulmuştur.

Yeniden kullanım odaklı yazılım geliştirme süreçleri konusunda henüz tam olarak kabul görmüş bir yöntemin olmaması, bu konuda akademiden destek alınarak bir çalışma yapılmasının önümüzdeki dönem çalışmaları açısından faydalı olacağı göstermektedir. ASELSAN SST grubu yazılım mühendisliğinde edinilen tecrübeler doğrultusunda, yeniden kullanım odaklı yazılım süreci belirleme çalışmalarının sistem mühendisliği ve proje yönetimi süreçlerine de etkisinin olacağı düşünülmektedir.

4.2. ASELSAN Bileşen Entegrasyon Çözümü

Farklı silah sistemi projelerinde kullanılan bileşenlerin ortaklanması amacıyla alan mühendisliği fazında oluşturulan bileşenler, konfigürasyon kontrolü altında bir bileşen havuzunda tutulmaktadır. Sistematik yeniden kullanım için bileşen havuzunda bulunan test edilmiş bileşenlerin kaynak kodlarının değiştirilmeden kullanılması gerekmektedir. Aynı şekilde bileşen havuzundan kullanılan bileşenlerin gerçekleştirilen testlerinin tekrarlanmasını önlemek için bileşenlerin yeniden derlenmesinin de engellenmesi gerekmektedir. Ayrıca performans kısıtları değerlendirildiğinde bileşen entegrasyonunun çalışma zamanı öncesinde yapılmasının gerekleri karşılamak için bir zorunluluk olduğu görülmektedir. Bu ihtiyaçları doğrultusunda, ASELSAN'da geliştirilen gömülü yazılımların UML standardına uyumlu model tabanlı araçlar kullanılarak geliştirilmesi ve bileşenlerin entegrasyonun da derleme öncesinde model seviyesinde UML portları kullanılarak otomatik olarak yapılmasına karar verilmiştir[10].



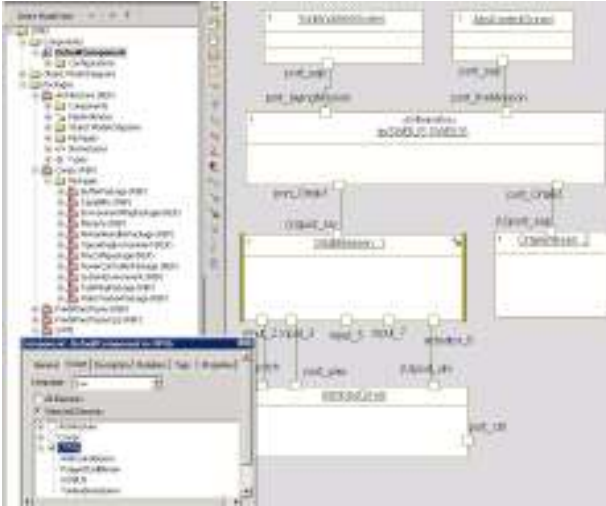
Şekil 2 Bileşen Entegrasyon Altyapısı[10]

Model seviyesinde otomatik bileşen entegrasyonunda düşünülen çözümün yazılım modelinde gösterimi Şekil 2'de verilmektedir. SWBUS[10] sınıfı şekildedeki gibi modellendiğinde, diğer tüm bileşenlerin sağladığı ve ihtiyaç duyduğu arayüz bilgilerine sahip olmaktadır. Bu arayüzlerin

analiz edilmesi ve ihtiyaç duyulan ve sağlanan arayüzlerin eşleştirilmesi ile bileşen entegrasyonu gerçekleştirilmektedir.. Bu işlem yazılan bir "script" ile otomatik olarak yapılabilmekte ve analiz sonrasında bileşenleri entegre eden kod otomatik olarak üretilebilmektedir. Bu sayede gömülü yazılımların önemli bir kısıtı olan derleme öncesi bileşen entegrasyonu ve tek çalıştırılabilir kod üretilmesi sağlanabilmektedir.

ASELSAN SST grubu yazılım mühendisliğinde model tabanlı yazılım geliştirme aracının önemli özelliklerinden bir tanesi de referans ile hazır bileşenlerin modele eklenebilmesidir. Modele referans ile eklenen bileşenlerin içeriği değiştirilememesine rağmen içeriği görüntülenebilmektedir. Bu paketler için proje konfigürasyonu ile kod üretilmesinin de önüne geçilebilmektedir. Bu sayede gri-kutu yeniden kullanım için altyapı hazırlanabilmektedir. Bileşenleri gri-kutu yeniden kullanabilmek için başlık dosyalarının ve kütüphanelerinin konfigürasyon kontrolü altına alınmış olması ve bu dosyaların tutulduğu dizinlerin proje dosyası içerisinde gösterilmesi gerekmektedir. Gri-kutu olarak eklenen bileşenler kullanılarak yeniden kullanılan bileşenler için üretilen kodlar değiştirilmeden, bileşen entegrasyonlarını sağlayan kod parçaları otomatik olarak üretilebilmektedir.

Şekil 3'te örnek bir yeniden kullanım projesi verilmiştir. Projede sadece UYMS paketi altındaki sınıflar yeni geliştirilmiştir. Yeni geliştirilen sınıflar proje ayarlarında seçilerek kullanılan model tabanlı geliştirme aracının sadece bu sınıflar için kod üretmesi sağlanmaktadır. Projedeki diğer paketler referans ile modele eklenmiştir. Yine proje ayarları kullanılarak referans ile eklenen hazır bileşenlerin başlık dosyaları ve kütüphaneleri modele eklenerek çalıştırılabilir kod oluşturma aşamasında problem yaşanmaması sağlanmaktadır.



Şekil 3 Örnek Yeniden Kullanım Projesi

Şekil 3'te görülen bütün bağlantılar tasarımcı tarafından gerçekleştirilmektedir. Ancak SWBus sınıfına bağlı olan portlar için ek olarak bileşenlerin ihtiyaç duydukları arayüzler ve sundukları arayüzler otomatik bir "script" aracılığıyla belirlenerek bileşen entegrasyonu gerçekleştirilmektedir.

4.3. ASELSAN'ın Performans Çözümü

Güvenlik-kritik ve/veya görev-kritik yazılımların çalışma

zamanındaki performansının derleme öncesinde analiz edilerek kritik noktaların tespit edilmesi önem taşımaktadır. ASELSAN SST grubu yazılım mühendisliği bölümünde kullanılan bileşen entegrasyonu metodunun model seviyesinde olması çalışma zamanı öncesinde sistem davranışının analiz edilebilmesini sağlamanın yanı sıra, dinamik entegrasyonun getirdiği çalışma zamanındaki sistem davranışının değişkenliği problemini de ortadan kaldırmaktadır. Çalışma zamanı öncesinde yapılan bileşen entegrasyonu sayesinde model seviyesinde global optimizasyonlara (global değişken kullanımı, işaretçi kullanımı vb) olanak sağlamakta, çalışma zamanında bileşenlerin arasındaki haberleşmede kaybolacak zaman doğrudan fonksiyon çağırma işlemleri ile en alt seviyeye indirilmektedir.

ASELSAN SST grubu yazılım mühendisliği bölümünde kullanılan bileşen entegrasyon teknolojisinde, bileşen entegrasyonunu sağlarken herhangi bir kayıtlanma, veri tabanı veya yeni bir mesaj kuyruğu kullanılmaması minimum işlemci gücü harcanmasında önemli bir etkidir[10]. Uygulamanın ve bileşenlerin gerçek zamanlılık gereklilerinin etkilenmemesi, bileşen entegrasyonunu sağlayan SWBUS'ın çalışmasının tamamen kendisini kullanan bileşenlerin öncelikleri (priority) ile olması sayesinde sağlanabilmektedir. Bu yöntemde fazladan işlemci gücü sadece otomatik üretilen kodun içerdiği anahtarlama kodları için harcanmaktadır.

Tüm projeler tarafından yeniden kullanıma uygun geliştirilen bileşenlerin kod kapsama oranını düşürmesinin önüne geçmek için geliştirme sürecinde kullanılan model tabanlı geliştirme aracının sunduğu değiştirilebilir kod üretim alt yapısının kullanılabilmesi düşünülmektedir.

ASELSAN SST grubu yazılım mühendisliği bölümünde geliştirilen silah sistemleri yazılımlarında genel olarak gerçek zamanlı işletim sistemi kullanılmaktadır. Bu şekilde işletim sisteminden kaynaklanacak gecikmeler en düşük seviyeye indirilmektedir. Silah sistemleri yazılımlarında dinamik nesne yaratımı veya hafıza ayrımı işlemleri zorunlu olmadıkça yapılmayarak yazılımların çalışma zamanı davranışlarının farklılık göstermemesi sağlanmaktadır. Dinamik hafıza ayrımının yapıldığı yazılım parçaları ise hafıza kaçak testlerine tabii tutularak doğrulanmaktadır. Ayrıca silah sistemleri yazılımlarında bileşen entegrasyonu model seviyesinde yapıldığından performans kaybı olmamaktadır. Bunlara ek olarak, kaynak kodların derleme zamanında derleyici optimizasyonlarının devreye alınması ve kullanılmayan derleyici özelliklerinin kapatılması sayesinde üretilen yazılımların performansı en üst düzeye çıkarılmaktadır[11].

4.4. ASELSAN Yazılım Test Yaklaşımı

ASELSAN SST grubu yazılım mühendisliğinde geliştirilen gömülü yazılım projeleri kapsamında bileşen gerekleri testlerinin genel amaçlı bilgisayar kullanılarak yapılmasının örnek sistemin hazırlanma süreci ve maliyeti değerlendirildiğinde maliyet-etkin bir çözüm olduğu görülmüştür. Bu nedenle silah sistemleri atış kontrol yazılımlarını işletim sisteminden soyutlayan bir katman ortaya konulmuştur.. Bu katman sayesinde geliştirilen bileşenlerin donanım ve işletim sisteminden bağımsızlığı sağlanarak basit donanımlar üzerinde fonksiyonel bileşen testleri gerçekleştirilebilmektedir. Ancak gerçek zamanlı sistemlerin zaman kritik gereklilerini doğrulayan performans testleri örnek

sistem üzerinde gerçekleştirilmektedir. Bu sayede fonksiyonel testler örnek sistemin geliştirilme süreci ile paralel olarak gerçekleştirilerek yazılımdaki hataların daha erken safhalarda çözülmesi sağlanmaktadır. Performans testlerinden önce gerekli analiz çalışmaları yapılarak bir problem olup olmadığını kontrol edilmektedir. Son testler örnek sistemin tamamlanması ile birlikte gerçekleştirilmekte ve performans gereklerinin karşılandığı, sistem üzerinde doğrulanmaktadır. ASELSAN SST grubunda yeniden kullanılacak bileşenlerin kullanımından beklenen getirinin elde edilebilmesi için bileşenlerin doğrulama süreci kapsamlı olarak ele alınmakta ve böylece yüksek kaliteye sahip bileşenlerin oluşturulması hedeflenmektedir. Bu amaç doğrultusunda bileşenlerin sadece kaynak kod olarak değil, birim testleri, fonksiyonel test durumları ve dokümantasyonu ile birlikte bir bütün olarak ortaya konulması istenmektedir. Ortak kullanılacak bu bileşenlerin birim test raporları, fonksiyonel test raporları, statik ve dinamik analiz raporları üretilerek doğrulanmasının ardından kullanıma alınması planlanmaktadır. Bu detaylı doğrulama faaliyetleri ilk etapta yüksek maliyete sahip gibi gözükse de, kaliteli bileşenlerin düzenli olarak yeniden kullanılmasıyla uzun vadede çok önemli kazanımlar sağlayacağı düşünülmektedir. Bu nedenle gerek ASELSAN SST grubu bünyesinde geliştirilecek, gerekse alt yüklenici firmalarca geliştirilmesi planlanan yeniden kullanım amaçlı yazılım bileşenlerinin bu doğrulama sürecinden geçmiş ve belirtilen kaliteye erişmiş olma şartı aranacaktır.

5. Sonuç

Bu bildiri ASELSAN SST grubunda geliştirilen gerçek zamanlı gömülü yazılımlar kapsamında yeniden kullanım sürecinde karşılaşılan zorluklar ve bu zorlukların giderilmesi için yapılan çalışmalar ortaya konulmuştur.

Dünyada gömülü yazılım alanında sistematik yeniden kullanımı sağlamış şirketlere bakıldığında alan mühendisliği ve uygulama mühendisliği kavramları ön plana çıkmaktadır. Bu şirketlerin alan mühendisliği ve uygulama mühendisliği kavramları etrafında yapılanarak yeniden kullanımı en üst seviyeye taşıdıkları görülmektedir. ASELSAN SST grubu yazılım mühendisliği bölümünde de yeniden kullanımı daha ileri seviyeye taşımak amacıyla akademi ile beraber çalışılarak alan mühendisliği ve uygulama mühendisliği kavramları çerçevesinde iyileştirmeler yapılabileceği düşünülmektedir.

Yazılım projelerinde sistematik yeniden kullanımı sağlayabilmek için yeniden kullanım odaklı bir süreç oluşturulması gerektiği ASELSAN SST grubu yazılım mühendisliği tarafından geliştirilen projelerde elde edilen deneyimler ışığında görülmüştür. Bu kapsamda silah sistemleri alanında alan mühendisliği çalışması yapılarak silah sistemlerindeki ortaklıklar ve değişkenlikler ortaya konulmuştur. Bu çalışma ile projelerin tamamını içeren bir özellik modeli ortaya konulmuş ve her bir silah sistemi yazılım projesinin ayrı bir proje olarak görülmesinin önüne geçilerek sistematik yeniden kullanımı sağlamak yolunda önemli bir adım atılmıştır.

Silah sistemleri yazılımlarının tasarımı aşamasında gerçek zamanlı gömülü sistemlerde en önemli kısıt olan performans ve güvenilirliği sağlamak amacıyla bileşen entegrasyon teknolojisi olarak çalışma zamanı öncesinde model seviyesi bir entegrasyon teknolojisi tercih edilmiştir. Bu teknolojinin tercih edilmesinin bir sebebi de günümüze kadar ASELSAN SST grubu gömülü yazılım projelerinde model tabanlı yazılım

geliştirme araçlarının kullanılmasıdır. Model seviyesi bileşen entegrasyonunun kullanılması nedeniyle yeniden kullanılan bileşenlerin modele dahil edilmesi gerekmektedir. Modele eklenen bileşenler için her projede yeniden kod üretimi ve derleme işlemi yapılmasının sistematik yeniden kullanımı azaltacağı düşüncesiyle kullanılan model tabanlı yazılım geliştirme aracının referans ile ekleme özelliği kullanılarak bileşenlerin kaynak kodunda değişiklik yapılmasına izin verilmeden modele eklenmesi sağlanmıştır. Bu şekilde bileşenler için her projede yeni sürüm oluşturulmasının önüne geçilerek, bileşenler için daha önce hazırlanan gerek, tasarım ve test dokümanlarının yeniden kullanımının sağlanabileceği düşünülmüştür.

Yazılım-yoğun silah sistemleri projelerinde sistematik yeniden kullanımı sağlamak için sistem geliştirme sürecinde yer alan bütün fonksiyonel birimlerin yeniden kullanım odaklı çalışması çok önemlidir. Sistem tasarımı aşamasında sistemde bulunacak donanımların özelliklerinin yanı sıra, yeniden kullanılacak yazılım bileşenlerinin de belirlenmesinin ve sistem tasarımının daha önce geliştirilen yazılım bileşenleri üzerine dayandırılmasının sistem geliştirme süresini ve maliyetini düşüreceği düşünülmektedir.

6. Kaynakça

- [1] Crnkovic, I., Larsson, M. *Building Reliable Component Based Software Systems*, Artech House Inc., Boston, 2002, ISBN: 1-58053-327-2
- [2] Crnkovic I., "Component-based approach for embedded systems", *Ninth International Workshop on Component-Oriented Software Design, 2004*
- [3] Crnkovic I., "Component-based Software Engineering – New Challenges in Software Development", *Information Technology Interfaces, 2003*
- [4] Clements, P., Northrop, L. *Software Product Lines: Practice and Patterns*, Addison Wesley (2002)
- [5] Szyperki C. *Component Software* 1998.
- [6] Savolainen, J., Oliver, I., Varvana Myllärniemi, V., Männistö, T., "Analyzing and Re-structuring Product Line Dependencies", 31st Annual International Computer Software and Applications Conference, July 2007
- [7] van Ommering, R., "Software Reuse in Product Populations", *IEEE Transactions On Software Engineering* Vol. 31, No. 7, July 2005
- [8] Nuseibeh, B., "Ariane 5: Who Dunit?", *IEEE Software*, 14(3):15-16, 1997
- [9] Almeida, E. S., Alvaro, A., Garcia, V. C., Mascena, J. C. C. P., Buregio, V. A. A., Nascimento, L. M., Lucredio, D., Meira, S. L., *Component Reuse In Software Engineering*, C.E.S.A.R e-book, 2007
- [10] Kahraman, E., İpek, T., "Gerçek Zamanlı Gömülü Yazılım Geliştirmede Bileşen Entegrasyon Deneyimleri", II. Ulusal Yazılım Mimarisi Konferansı UYMK'08, 11-12 Eylül, Ege Üniversitesi, İzmir
- [11] İyidir, B., Kalay, A., İşbitiren, G., Yılmaz, Ö., "Gerçek Zamanlı Gömülü Yazılımlarda Derleyici Bayraklarının Etkin Kullanımı", Yazılım Kalitesi ve Yazılım Geliştirme Araçları 2008 YKGS2008, 9-10 Ekim, TC İstanbul Kültür Üniversitesi, İstanbul
- [12] Ünal, V., Kahraman, E. "Gerçek Zamanlı Gömülü Sistem ve Yazılım Tasarımı'nda ASELSAN Yaklaşımı", *III. Ulusal Yazılım Mühendisliği Sempozyumu UYMS'07*, 27-30 Eylül 2007, Bilkent Üniversitesi, Ankara