

## Yazılım Ürün Hattı Yaklaşımında Model GÜdümlü Uygulama Mühendisliği

Elif Kamer KARATAŞ<sup>1</sup>, Barış İYİDİR<sup>2</sup>

<sup>1,2</sup> ASELSAN A.Ş. Savunma Sistem Teknolojileri Grubu

<sup>1</sup> [ekkaratas@aselsan.com.tr](mailto:ekkaratas@aselsan.com.tr) , <sup>2</sup> [biyidir@aselsan.com.tr](mailto:biyidir@aselsan.com.tr)

### Özetçe

Yazılım ürün hattı yaklaşımının uygulama mühendisliği aşamasında, alan mühendisliği sürecinde oluşturulan temel varlıkların kullanılması ile ürün maliyetinin düşürülmesi hedeflenmektedir. Aynı zamanda uygulama mühendisliği sürecinde gerçekleştirilen adımların otomatize edilmesi ile de maliyetin düşürülmesi için çalışmalar yapılmaktadır[1][2]. Bu bildiri kapsamında ASELSAN SST grubu yazılım mühendisliği bölümü tarafından yapılan yazılım ürün hattı yaklaşımında uygulama mühendisliği adımlarının otomatize edilmesi ile ilgili çalışmalar aktarılacaktır.

### 1. Giriş

Büyük ve yazılım odaklı projeler, günümüzde bir ürün yerine bir ürün ailesi için geliştirilmektedir. Bu sayede ürünlerin farklı müşteri istekleri doğrultusunda değiştirilmesi kolaylaşmakta ve yeni ürünlerin çıkarılması için gereken süre azalmaktadır. Ürün ailesi için yazılım geliştirmek için kullanılan yöntemlerden bir tanesi de yazılım ürün hattı yaklaşımıdır[3]. Yazılım ürün hattı yaklaşımında ortak kullanılacak temel varlıkların oluşturulması ile bu varlıklar kullanılarak ürün oluşturma süreçleri birbirinden ayrılmıştır. Ürün ailesi için geliştirme yapılırken alan mühendisliği sürecinde ortaklıkların ve değişkenliklerin modellenmesi ile ürün ailesi için oluşturulacak yazılım mimarisinin belirlenmesi önem kazanmaktadır. Değişkenlikler ile farklı ürünler ifade edilirken, yazılım mimarisi ile bu farklılıkların yazılım içerisine hızlı ve kolay bir şekilde uyumlandırılması hedeflenmektedir. Alan mühendisliği sürecinde ürün ailesi içerisinde ortak kullanılacak yazılım bileşenleri, özellik modeli ve yazılım mimarisi göz önüne alınarak belirlenmekte ve gerçekleştirilmektedir[4].

Ürün ailesi içerisindeki ortaklıklar ve değişkenlikler genel olarak özellik modelleri ile ifade edilirken, ürün ailesi için oluşturulan yazılım mimarileri UML (Unified Modelling Language) modelleme dilini temel alan model tabanlı yazılım geliştirme araçları kullanılarak oluşturulmaktadır. Uygulama mühendisliği sürecinde yeni bir ürün tanımlamak için ürün ailesi özellik modeli kullanılmakta, yeni ürünün barındırdığı özellikler ürün ailesi için oluşturulan özellik modeli kullanılarak belirlenmektedir. Yeni tanımlanan ürün için seçilen özellikleri gerçekleyen yazılım bileşenleri bulunmakta ve ürün ailesi yazılım mimarisi kullanılarak yeni ürün için çalışabilir yazılım oluşturulmaktadır.

Model güdümlü yazılım geliştirme yaklaşımı, platformdan bağımsız modeller aracılığıyla farklı platformlar için çalıştırılabilir kod üretmeyi sağlaması nedeniyle günümüzde tercih edilmektedir[5]. Özellikle karmaşık sistemlerin geliştirilmesi aşamasında soyutlama seviyesinin yüksek olması sayesinde yazılım geliştirme sürecini kolaylaştırmaktadır. Model güdümlü yaklaşım kullanılarak, yazılım ürün hattı uygulama mühendisliği sürecinde ürün

özelliklerini karşılayan uygulama yazılımının otomatik olarak üretilmesi mümkün görünmektedir[1][2]. Bu işlem için ise model tabanlı yazılım geliştirme aracına girdi olarak verilecek, uygulama yazılımını ifade eden bir modele ihtiyaç duyulmaktadır. Uygulama yazılımlarını ifade eden modellerin ise alan mühendisliği sürecinde üretilen özellik modeli, hazır bileşenler ve ürün ailesi yazılım mimarisi kullanılarak oluşturulabileceği düşünülmektedir. Uygulama yazılımını ifade eden modelin otomatik olarak oluşturulması durumunda yeni bir ürün oluşturmak için harcanacak iş gücünün çok düşük bir seviyeye indirilebileceği düşünülmektedir.

### 2. Özellik modeli, Referans Mimari ve Bileşenler arasındaki ilişkiler

Yazılım ürün hattı sürecinde, ilk önce alan analizi gerçekleştirilmektedir. Bu analiz sonucunda alan içerisindeki ortaklıklar ve değişkenlikler belirlenmekte ve özellik modeli ile ifade edilmektedir [3][6][7]. Özellik modeli içerisinde yer alan özellikler için birçok tanım bulunmakla beraber bu bildiri kapsamında yapılan çalışmada FORM (Feature Oriented Reuse Method) yönteminde kullanılan özellik tanımı temel alınmıştır[7].

FORM yaklaşımında özellik modeli “Capability”, “Operating Environment”, “Domain Technology” ve “Implementation Technique” olmak üzere 4 katmandan oluşmaktadır. Bu katmanlardan “Capability” ve “Operating Environment” katmanından bulunan özellikler temel olarak sistem yeteneklerini ve sistemde bu yetenekleri gerçeklemek için kullanılan cihazları ifade etmektedir. Yazılım mimarisinin oluşturulmasında ve bileşenlerin belirlenmesinde bu iki katman önemli rol oynarken, “Domain Technology” ve “Implementation Technique” katmanında ifade edilen değişkenlikler bileşenlere parametre olarak verilmektedir[7].

Özellik modeli, yazılım ürün ailesi referans mimarinin oluşturulması ve bileşenlerin belirlenmesi süreçlerinde girdi olarak kullanılmakla[8] beraber her yeni ürünün geçerliliğini doğrulamak için başvurulacak bir referans olarak da kullanılmaktadır.

Yazılım ürün hattı yaklaşımında referans mimari problem uzayında tanımlanan özellik modeli ile çözüm uzayında bulunan bileşenler arasındaki bağlantıyı kurmak için kullanılmaktadır. Bu işlem analiz aşamasında belirlenen ürün ailesi yeteneklerinin çözüm uzayında bulunan bileşenler ile eşleşmesi olarak da düşünülebilir. Referans mimaride bu eşleşmenin yanı sıra bileşenlerin entegrasyon yöntemi, alan için önemli tasarım kararları da yer almaktadır. Referans mimarinin problem uzayındaki değişkenlikler göz önüne alınarak oluşturulmasının ve farklı ürün konfigürasyonlarının kolay bir şekilde gerçekleştirilmesini sağlayacak bir yapıda olmasının yazılım ürün hattının başarıya ulaşmasında en önemli etken olduğu düşünülmektedir[4][8].

Özellik modeli üzerinde özellik ilişkilendirme analizi (Feature Binding Analysis) gerçekleştirilerek çözüm uzayında

gerçeklenmesi gereken bileşenler tanımlanmaktadır[4]. Bu anlamda bileşenler, çözüm uzayında, özellik modelinde tanımlanan problem uzayı yeteneklerini gerçekleyen yazılım parçaları olarak değerlendirilebilir. Bir bileşen problem uzayında tanımlanan birden fazla özelliği gerçekleyebildiği gibi sadece bir özelliği de gerçekleyebilir. Ancak problem uzayında tariflenen bir özelliğin birden fazla bileşen tarafından gerçekleştirilmesi tavsiye edilmemektedir[8]. Bu ilişkilendirme analizi sırasında “optional” olarak tanımlanan özelliklerin ayrı bileşen olarak tanımlanması önerilirken, birbirine sürekli ihtiyaç duyan özelliklerin aynı bileşende gerçekleştirilmesi önerilmektedir. Bütün yazılım bileşenlerini etkileyen özelliklerin ise bileşenlere parametre olarak geçirilmesi ve değişkenliğin bileşen içerisinde ele alınmasının uygun olacağı belirtilmektedir.

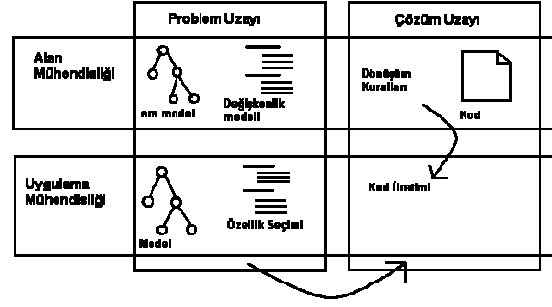
### 3. Model Güdümlü Uygulama Mühendisliği

Model güdümlü yazılım geliştirme yaklaşımı, yazılım sisteminin bütün önemli özelliklerinin platformdan bağımsız modeller ile tanımlanmasını hedeflemektedir. Modeller, yüksek soyutlama seviyesi nedeniyle kaynak kod ile karşılaştırıldığında paydaşların üzerinde daha kolay tartışılabileceği ortam sağlamaktadır. Platformdan bağımsız modeller üzerinde uzlaşa sağlandıktan sonra belirtilen modeller kullanılarak farklı platformlar için otomatik kod üretimi sayesinde kodlama sürecinde oluşabilecek hatalar da en düşük seviyeye indirilebilmektedir.

UML, analiz ve tasarım aşamalarında oluşturulan modellerin standart bir şekilde oluşturulmasını sağlayan bir modelleme dili olduğundan yazılım geliştiricileri tarafından tercih edilmektedir[6]. UML modelleri ile yazılım yoğun sistemlerin kodlama işlemleri gerçekleştirilmeden analizinin yapılması mümkün olmakta ve hataların erken bulunması sayesinde yazılım geliştirme maliyeti azalmaktadır. UML’de tanımlanan “XML Metadata Interchange (XMI)” yeteneği ile UML modelleri XML dili ile ifade edilebilmekte ve modellerin farklı yazılım geliştirme araçlarına taşınması mümkün olmaktadır.

Yazılım ürün hattı yaklaşımında alan mühendisliği ve uygulama mühendisliği iki ayrı süreç olarak tanımlanmış olsa da model güdümlü geliştirme yaklaşımı kullanılarak bu iki sürecin aynı çatı altında birleştirilebileceği düşünülmektedir. Ürün ailesini tarifleyen bir modelin farklı ürünler için müşteri istekleri doğrultusunda düzenlenmesi ve ortaya çıkan modelden ürün oluşturulması mümkün görünmektedir. Bu şekilde model güdümlü yazılım geliştirme yönteminin soyutlama yeteneği ile yazılım ürün hattı yaklaşımının değişkenlik yönetiminin birleştirilebileceği düşünülmektedir.

Şekil 1’de alan mühendisliği ve uygulama mühendisliği süreçlerinde oluşturulan modeller verilmiştir. Alan mühendisliği sürecinde problem uzayında bulunan ortaklıkların ve değişkenliklerin modellenmesi (özellik modeli) ile farklı ürünler tanımlanabilmekte, yine alan mühendisliği sürecinde referans mimari tanımlanarak problem uzayından çözüm uzayına geçiş yöntemi belirtilebilmektedir. Uygulama mühendisliği aşamasında alan mühendisliği sürecinde oluşturulan özellik modeli kullanılarak yeni ürünün sahip olacağı yetenekler belirlenebilmekte ve son olarak alan mühendisliği sürecinde tanımlanan referans mimari ve uygulama mühendisliğinin ilk aşamasında oluşturulan ürüne özel özellik modeli ile ürün oluşturulabilmektedir.



Şekil 1 Alan Mühendisliği ve Uygulama Mühendisliği Süreç Çıktıları[6]

Uygulama mühendisliği sürecinde temel olarak alan mühendisliği sürecinde oluşturulan ürünler kullanıldığından uygulama mühendisliği sürecinin model güdümlü bir geliştirme yöntemi kullanılarak otomatize edilebileceği ve hızlı bir şekilde ürün oluşturulabileceği düşünülmektedir. Bu otomasyon süreci içerisinde alan mühendisliği sürecinde oluşturulması gereken özellik modeli, referans mimari ve bileşenlerin hazırlanmış olması durumunda, ürün geliştiricinin sadece ürün özelliklerini seçmesinin yeterli olacağı düşünülmektedir. Ürüne özel model oluşturulduktan sonra model tabanlı yazılım geliştirme araçlarının otomatik kod üretme özelliği kullanarak çalışabilir yazılımın oluşturulabileceği düşünülmektedir.

### 4. Uygulama Mühendisliği Otomasyonu

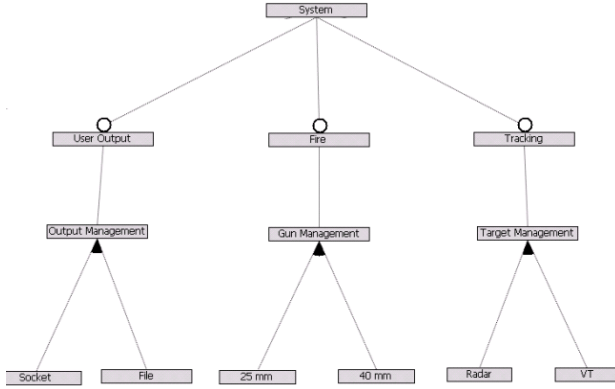
Yazılım ürün hattı yaklaşımında temel odaklanma ortak kullanılabilir ürünlerin yaratıldığı alan mühendisliğine olsa da uygulama mühendisliği sürecinde tanımlanan adımlar için de işgücü ayrılmaktadır. Uygulama mühendisliği sürecinde, genel olarak ürüne özel değişiklikler yapıldığından harcanan işçiliğin yeniden kullanılabilirlik açısından bir geri dönüşü olmamaktadır. Bu durum uygulama mühendisi olarak görev alan kişilerde iş tatmini konusunda sorun oluşturabilmekte, ilgili personelin potansiyelinin tam olarak kullanılamamasına neden olmaktadır[1][2]

Uygulama mühendisliği sürecinde yapılan işlemlerin otomatik olarak gerçekleştirilmesi ile ürüne özel harcanan işgücünün alan mühendisliği sürecine kaydırılmasının mümkün olacağı düşünülmektedir. Bu şekilde uygulama mühendisliği sürecinde görev alan personelin potansiyelinin tamamının kullanılmasının yanı sıra alan mühendisliğinde üretilecek yeniden kullanılabilir ürünler için ek işgücü yaratılması hedeflenmektedir.

ASELSAN SST grubu yazılım mühendisliğince uygulama mühendisliğinin otomasyonu için örnek bir proje gerçekleştirilmiştir. Bu proje ile özellik modelinden uygulamada yer alması istenen özelliklerin seçilmesi sonucunda otomatik olarak çalışabilir kod üretimi hedeflenmiştir. Proje kapsamında alan mühendisliği ürünleri olan özellik modeli, yazılım referans mimarisi ve ortak kullanılacak bileşenlerin hazır olduğu varsayılarak, yazılım ürün hattı sürecinin uygulama mühendisliği aşamasına odaklanılmıştır.

Örnek proje kapsamında silah kontrol sistemleri alanı için alan analizi aşamasında oluşturulan özellik modeli basitleştirilerek kullanılmıştır. Bu özellik modelinin oluşturulması aşamasında FORM[7] (Feature Oriented Reuse

Method) metodu temel alınmış ve FORM metodunda tanımlanan özellikler arası ilişkilerden “mandatory”, “optional” ve “alternative” tipi ilişkiler kullanılmıştır. FORM metodunda tariflenen özellik modeli katmanlarından “Capability” ve “Operating Environment” katmanındaki özelliklerin temel olarak yazılım bileşenlerini belirlemesi nedeniyle örnek çalışma kapsamında sadece bu katmanda yer alan özellikler modellenmiştir[4][7][8].



Şekil 2 Özellik Modeli

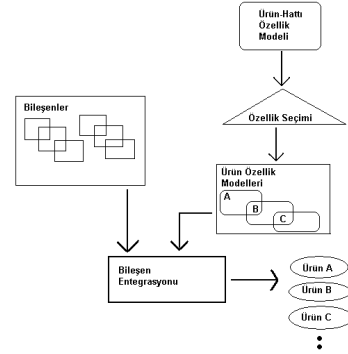
Silah kontrol sistemleri incelendiğinde temel olarak bir hedefin izlenmesi, bu hedefin ateş altına alınması ve ateş sırasında oluşan hataların raporlanması işlevlerinin gerçekleştirildiği görülmektedir. Bu işlevler gerçekleştirilirken farklı silah türleri kullanılabilir gibi farklı hedef tespit cihazları da kullanılabilir. Silah kontrol sistemlerindeki temel değişkenlikleri ifade eden özellik modeli Şekil 2’de verilmiştir. Özellik modelinde yer alan “user output” özelliği hataların raporlanmasını, “Fire” özelliği hedefin ateş altına alınmasını ve “Tracking” özelliği ise hedefin izlenmesini ifade etmektedir. Bu işlevler sistemin yeteneklerine göre ürün ailesinde yer alan bütün ürünlerde olmayabilir. Bu nedenle “Optional” olarak modellenmiştir. Sistemde seçilen “optional” özellikler, sağladığı işlevi yerine getirmek için ikinci seviyede yer alan bazı özelliklere ihtiyaç duymaktadır. Bu ilişkiler özellik modelinde “Mandatory” tipi ilişkiler ile verilmiştir. İkinci seviyede yer alan özellikler ise çıkış hattının kontrolünü, silahların yönetimi ve hedef bilgilerinin yönetimini ifade etmektedir. Özellik modelinde en alt seviyede verilen özellikler ise sistem işlevlerinin yerine getirilebilmesi için kullanılacak birimleri belirtmektedir. Bu birimler sistemlerde beraber kullanılacak olmasına rağmen örnek proje kapsamında özellik ilişkisi çeşitliliğini arttırmak amacıyla “Alternative” olarak modellenmiştir.

#### 4.1. Yazılım Ürün Hattı Ürün Oluşturma Yazılımı

ASELSAN SST grubu yazılım mühendisliği bölümünde yazılım geliştirme aracı olarak model tabanlı yazılım geliştirme araçları kullanılmaktadır. Bu araçlar genel olarak oluşturulan modellerin incelenmesi, değiştirilmesi veya yeni model oluşturulmasını sağlamak üzere çeşitli eklentiler sunmaktadır.

Yazılım ürün hattı ürün oluşturma yazılımı model tabanlı geliştirme aracının sunduğu VBA (Visual Basic For Applications) eklenti arayüzü ile gerçekleştirilmiştir. Bu yazılım Şekil 3’de verilen uygulama mühendisliği adımlarını

otomatize etmek amacıyla geliştirilmiştir. Yazılım, yeni ürünün özelliklerini seçmek için bir arayüz sunmakta ve seçilen özelliklere uygun çalışabilir yazılımın otomatik olarak oluşturulmasını sağlamaktadır.



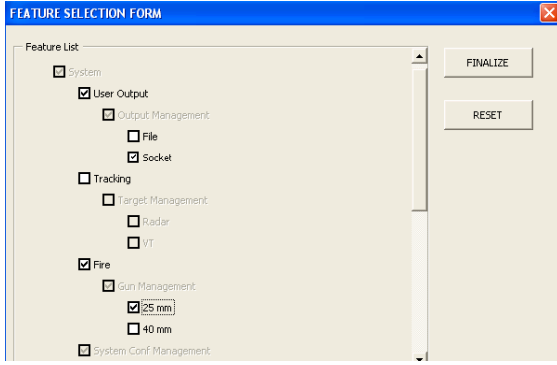
Şekil 3 Uygulama Mühendisliği Süreci

#### 4.1.1. Ürün Modelinin Oluşturulması

Yazılım ürün hattı içerisinde bulunan ürünlerin sahip olabileceği özellikler ve bu özelliklerin aralarındaki ilişkiler alan mühendisliği aşamasında oluşturulan özellik modeli ile belirlenmektedir. Özellik modelini oluşturmak için sadece özellik modellemek için kullanılan araçlar bulunduğu gibi UML tabanlı geliştirme araçları da kullanılabilir. ASELSAN SST grubu yazılım mühendisliği bölümünde özellik modelleme işlemi, sadece özellik modeli oluşturmak için kullanılan ticari bir araç kullanılarak gerçekleştirilmektedir. Bu aracın oluşturulan özellik modelini XML olarak ifade etme yeteneği ile özellik modeli, model tabanlı yazılım geliştirme ortamına aktarılabilir.

Özellik modeli oluşturma aracı kullanılarak oluşturulan XML dosyası, yazılım ürün hattı ürün oluşturma yazılımına girdi olarak verilmiştir. Bu XML dosyası kullanılarak kullanıcının ürün özelliklerini seçmesi için oluşturulan kullanıcı arayüzü Şekil 4’te verilmiştir. Kullanıcı arayüzünde gösterilen özellik listesinin en üst bölümünde gösterilen özellik hedef uygulamayı temsil etmekte ve özellik modelinde bulunan bir özelliğe karşılık gelmektedir. Kullanıcı arayüzünde gösterilen diğer özellikler ise özellik modelinde belirtilen özelliklere karşılık gelmektedir. Kullanıcıyı doğru şekilde yönlendirebilmek amacıyla arayüz, özellik modelinde bulunan hiyerarşiye uygun şekilde oluşturulmuştur.

Kullanıcı arayüzünde geçersiz yapılandırmaların oluşturulmasını önlemek amacıyla özellik modelinde tanımlanan ilişkiler doğrultusunda gerekli önlemlerin alınması sağlanmıştır. Kullanıcıya yardımcı olabilmek amacıyla özellik modelinde zorunlu olarak belirtilen özellikler (mandatory features) kullanıcının tercihine bırakılmadan otomatik olarak seçilmekte ve seçimin kullanıcı tarafından kaldırılması engellenmektedir. Ayrıca özellik modelinde belirtilen ilişkiler doğrultusunda kullanıcı arayüzünde seçilen her özellik için bu özelliğin gerektirdiği özellikler otomatik olarak seçilmekte, seçilen özelliğin dışladığı özelliklerin ise kullanıcı tarafından seçilmesi engellenmektedir. Kullanıcının özellik seçim işlemini tamamlaması ile birlikte seçilen özelliklerin geçerli bir ürün tanımlayıp tanımlamadığı özellik modeli kullanılarak kontrol edilmekte, kullanıcı hata durumunda hatalı veya eksik seçim konusunda uyarılmaktadır.



Şekil 4 Özellik Modelinden Oluşturulan Kullanıcı Arayüzü

Uygulama geliştirme mühendisi, kullanıcı arayüzünde verilen listeyi kullanarak hedef uygulamasında yer almasını istediği özellikleri seçebilmektedir. Şekil 4'te verilen arayüzde uygulama mühendisinin raporlama ve ateş altına alma işlevlerinin yeni sistemde yer almasını istediğini görmekteyiz. Ayrıca raporlama arayüzü için soket, silah olarak da 25 mm Top seçilmiştir. Bu seçimlere uygun olarak bir yazılımın "Yazılım Ürün Hattı Ürün Oluşturma Yazılımı" tarafından oluşturulması hedeflenmiştir.

#### 4.1.2. Özellik-Bileşen Eşleşmesi

Kullanıcının ürün özelliklerini seçmesinin ardından çalışabilir yazılımının oluşturulması için gerekli işlemler başlanmaktadır. İlk olarak seçilen özellikler ile bu özelliklerin gerçekleştirildiği bileşenlerin eşleşmesi gerekmektedir. Alan mühendisliği sürecinde belirlenen bileşenlerin gerçekleştirilmesi UML tabanlı yazılım geliştirme aracı kullanılarak yapılmıştır. Hazırlanan bileşenler ile özellik modeli arasındaki izlenebilirliğin sağlanması için ise UML'de tanımlı "stereotype" kavramı kullanılmıştır. Bu anlamda hazırlanan her bileşen için özellik ilişkilendirme analizi sonucunda oluşan özellik grubunu belirten bir "stereotype" atanmıştır. Dolayısıyla bileşen eşleme işlemi bileşen kütüphanesi içerisinde arama yapılarak gerçekleştirilebilir. Etiketleme işlemi UML dilinde tanımlanan "stereotype" kavramı ile yapıldığından arama işlemi de model elemanlarının "stereotype" alanları üzerinde yapılmaktadır.

Bu bildiri kapsamında gerçekleştirilen çalışmada anlaşılabilirlik göz önüne alınarak özellik modelindeki özellikler ile bileşenler arasında birebir eşleme yapılmıştır. Birden fazla özelliğin bir bileşen tarafından gerçekleştirilmesi durumunda da aynı yaklaşım kullanılarak uygulama mühendisliği süreci otomatizasyonu sağlanabilir. Ancak bir özelliğin birden fazla bileşeni etkilediği durumlarda bu özelliklerin bileşenlere parametre olarak verilmesi önerildiğinden[4][8] model seviyesinde bileşen değişkenlerinin, özellik modelindeki seçime uygun şekilde ayarlanması gerekmektedir.

Şekil 5'de model tabanlı yazılım geliştirme ortamında bulunan bileşenler ve "stereotype" bilgileri verilmiştir. Yazılım ürün hattı ürün oluşturma yazılımı her seçilen özellik için bileşen kümesinin içinde "stereotype" alanı seçilen özellik adıyla aynı olan bileşeni aramakta ve yeni ürün için hazırlanan nesne model diyagramına (Object Model Diagram) eklemektedir. Bu aşamadan sonra Nesne Model Diyagramı

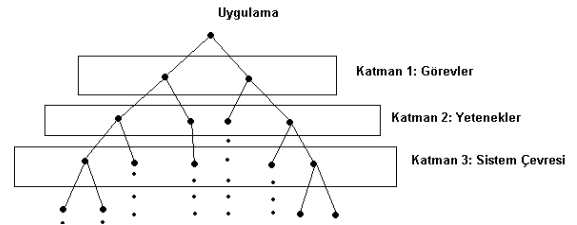
üzerinde bileşen entegrasyonu gerçekleştirilecektir.



Şekil 5 Model Tabanlı Araçta Oluşturulan Bileşen Havuzu

#### 4.1.3. Bileşen Entegrasyonu

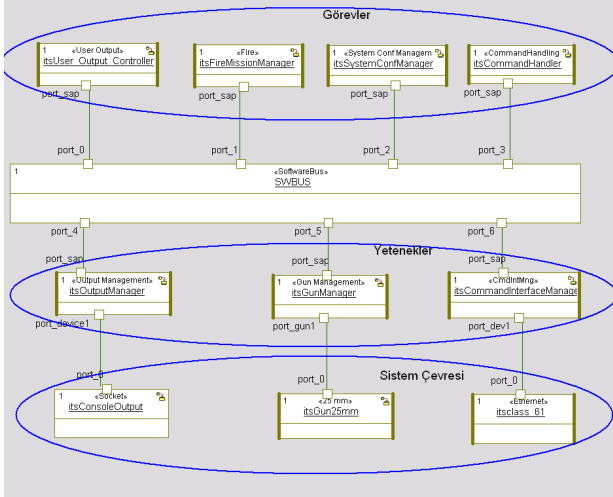
Çalışabilir yazılımının oluşturulabilmesi için bileşenlerin entegrasyon işlemlerinin gerçekleştirilmesi gerekmektedir. Bileşen entegrasyon kuralları ve alt yapıları alan mühendisliği çalışmaları kapsamında tariflenmekte ve hazırlanmaktadır. Bu kapsamda ASELSAN SST grubu yazılım mühendisliğinin gerçekleştirdiği alan mühendisliği çalışmaları sonucunda bileşen entegrasyon teknolojisi olarak SWBUS (Software Bus) yapısının kullanılmasına karar verilmiştir[10]. Ayrıca ürün ailesi için bir yazılım referans mimarisi oluşturulmuş ve bileşenlerin bağlantı noktaları belirlenmiştir. Yazılım referans mimarisinin katmanları Şekil 6'da verilmiştir.



Şekil 6 Yazılım Referans Mimari Katmanları

Bu yapı doğrultusunda birinci seviyede yer alan özellikler (Şekil 2: User Output, Fire, Tracking) görevlere, ikinci seviyede yer alan özellikler (Şekil 2: Output Management, Gun Management, Target Management) yeteneklere karşılık gelmektedir. İkinci seviyenin altındaki özellikler ise "Sistem çevresi" ile ilgili daha alt seviye özelliklerdir. Belirtilen katmanlar göz önünde bulundurularak entegrasyon aşamasında birinci seviyedeki özelliklere karşılık gelen bileşenler SWBUS'a direk bağlanmakta ve daha üst seviye bileşenler olduklarını ifade etmek amacıyla nesne model diyagramında SWBUS'ın üzerinde yer almaktadırlar. İkinci seviye bileşenler de direk SWBUS'a bağlanmakta ve hiyerarşide bir alt seviyede olduklarını ifade etmek amacıyla nesne model diyagramında SWBUS'ın altında yer almaktadırlar. Üçüncü seviyeden itibaren daha alt seviyedeki bileşenler SWBUS'a doğrudan bağlanamamaktadır. Bu bileşenler referans mimaride tanımlandığı şekilde, özellik modelinde bağlı oldukları özelliklere karşılık gelen

bileşenlere bağlanmaktadır.



Şekil 7 Örnek Uygulama Nesne Model Diyagramı

Yazılım ürün hattı ürün oluşturma yazılımı özellikler ile eşleştirilen bileşenleri belirledikten sonra referans mimaride verilen kurallar ışığında bileşen bağlantılarını gerçekleştirmektedir. Bu işlem sonucunda otomatik olarak oluşturulan nesne model diyagramı Şekil 7’de verilmiştir. Bileşen entegrasyonun tamamlanması ile ürün için uygulama geliştirme işlemi tamamlanmış olmaktadır. Model tabanlı geliştirme aracının otomatik kod üretme özelliğinin aktive edilmesi ile oluşturulan ürün modelinden çalışabilir yazılımın oluşturulması sağlanmaktadır.

## 5. Sonuç

Yazılım ürün hattı yaklaşımında odaklanma bütün ürünlerde ortak kullanılacak yapıtaşlarının oluşturulduğu alan mühendisliği sürecine olmaktadır. Uygulama mühendisliği sürecinde ürüne özel yazılım parçaları geliştirilse de temel olarak alan mühendisliği sürecinde oluşturulan yapıtaşları kullanılarak ürünler hazırlanmaktadır. Bu anlamda uygulama mühendisliği süreci yazılım tasarım sürecinden çok sistem üretim sürecine benzemektedir. Bu süreçte sistem bileşenlerinin entegrasyonu sağlanarak sistem üretimi gerçekleştirilmektedir.

Sistem üretiminin gerçekleştirilebilmesi için üretim aşamasında kullanılacak kuralların tamamı tasarım sürecinde belirlenmektedir. Benzer şekilde uygulama mühendisliği sürecinde gerçekleştirilecek bileşen entegrasyonunun nasıl yapılacağı da alan mühendisliği aşamasında tariflenmektedir. Dolayısıyla uygulama mühendisliği sürecinde ürüne özel bileşen geliştirme dışında kalan bütün adımların otomatize edilebileceği düşünülmüştür. Ürüne özel bileşenlerin de ürün oluşturma süreci öncesinde hazır olması gerekliliği düşünüldüğünde bir çeşit sistem üretim süreci olan ürün oluşturma sürecinde bulunan adımların alan mühendisliğinde belirlenen kurallar (Referans Mimari, Bileşen entegrasyon teknolojisi, Bileşen arayüzleri vb) doğrultusunda otomatik olarak gerçekleştirilebileceği görülmektedir.

ASELSAN SST grubu yazılım mühendisliği bölümünde gerçekleştirilen çalışmalar sonucunda uygulama mühendisliği sürecinin otomasyonunda kullanılacak bir yazılım oluşturulmuştur. Model güdümlü bir yaklaşım kullanılarak

ürün ailesi özellik modelinden bir ürüne ait özellik modelinin çıkarılması kullanıcı arayüzü ile sağlandıktan sonra ürüne özel ürün ağacı kullanılarak yazılım nesne model diyagramı oluşturulmuştur. Model tabanlı aracın otomatik kod üretme özelliğinden yararlanılarak nesne model diyagramında otomatik olarak çalışabilir kod üretimi sağlanmıştır.

Yapılan çalışma sonucunda uygulama mühendisliği süreç adımlarının otomatize edilebilmesi için alan mühendisliği sürecinde, olası ürünlerin özelliklerinin modellendiği özellik modelinin oluşturulması gerekliliği görülmüştür. Ayrıca özellik modeli ile bileşenler arasındaki geçişi sağlayan ve bileşenlerin entegrasyon kurallarını tarifleyen referans mimari olmaması durumunda uygulama mühendisliği sürecindeki adımların otomatize edilmesinin mümkün olmayacağı düşünülmektedir.

ASELSAN SST grubu yazılım mühendisliği bölümünde gerçekleştirilen çalışma sonucunda uygulama mühendisliği süreci adımlarının model güdümlü bir yaklaşım ile otomatize edilebileceği görülmüştür. Bu sayede ürüne özel harcanan işgücünün alan mühendisliği sürecine kaydırılması ile uygulama mühendisliği sürecinde görev alan personelin potansiyelinin kullanımının kullanılmasının yanı sıra alan mühendisliğinde üretilecek yeniden kullanılabilir ürünler için ek işgücünün yaratılabileceği düşünülmektedir.

## 6. Kaynakça

- [1] Krueger, W. C., “New Methods in Software Product Line Development”, 10th International Software Product Line Conference, 21-24 August 2006, Baltimore, Maryland, USA
- [2] Groher, I., Papajewski, H., Voelter, M., “Integrating Model-Driven Development and Software Product Line Engineering”, Eclipse Summit Europe 2007, Ludwigsburg, Germany
- [3] Clements, P., Northrop, L. *Software Product Lines: Practice and Patterns*, Addison Wesley (2002)
- [4] Jaejoon Lee, Kyo C. Kang, and Sajoong Kim, “A Feature-Based Approach to Product Line Production Planning”, LNCS, Vol. 3154/2004, pp. 183-196, 2004
- [5] Czarnecki, K., Antkiewicz, M., Hwan, C., Kim, P., Lau, S., Pietroszek, K., “Model-Driven Software Product Lines”, OOPSLA’05, October 16–20, 2005, San Diego, California, USA
- [6] Beuche, D., Dalgarno, M., “Software Product Line Engineering with Feature Models” [www.pure-systems.com/fileadmin/downloads/pure-variants/tutorials](http://www.pure-systems.com/fileadmin/downloads/pure-variants/tutorials)
- [7] Kang, K.C., Kim, S., Lee, J., Kim, K., Kim, G.J., Shin, E., FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures, 1998
- [8] M. Kim, J. Lee, K.C. Kang, I. Y. Hong, S. Bang, “Re-engineering software architecture of home service robots: a case study” in *Proc. of the 27th International Conference on Software Engineering (ICSE 2005)*, 505-513, 2005
- [9] Rushton, G., Baillargeon, R., “Model-Driven Product Line Software Development Process”, General Motors Corporation, SAE International, 2005 SAE
- [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