

Yazılım Mimari Tasarımından Yazılım Geliştirme Çatısının Üretilmesinde Model GÜdümlü Bir Yaklaşım

İbrahim Onuralp Yiğit¹, Nafiye Kübra Turhan², Ahmet Erdiñç Yılmaz³, Bülent Durak⁴

^{1,2,3,4}ASELSAN A.Ş. SST Sk. Bşk.lığı-Komuta Kontrol Yaz. Tsr. Mdl. P.K.1 06172,
Yenimahalle/Ankara, Türkiye
¹ioyigit@aselsan.com.tr, ²nkturhan@aselsan.com.tr,
³aeyilmaz@aselsan.com.tr, ⁴durak@aselsan.com.tr

Özet. Bu bildiri, yazılım mimari tasarımından, yazılım geliştirme çatısının otomatik olarak oluşturulması için uygulanan model güdümlü yaklaşım, bu yaklaşımın sağladığı kazanımlar ve edinilen tecrübeler paylaşılmaktadır. Belirli bir alanda faaliyet gösteren yazılım ekipleri, genellikle ürünlerini tanımladıkları referans yazılım mimarisine uygun bir şekilde geliştirmektedirler. ASELSAN SST Sektör Başkanlığı bünyesinde deniz sistemleri alanında çalışan bir yazılım ekibi tarafından geliştirilen tüm yazılımlar, tanımlanan referans yazılım mimarisine uygun olarak tasarlanmaktadır. Referans yazılım mimarisine uygun olarak tasarlanmış yazılım mimarisinden, kodlamaya geçiş sürecini otomatikleştirmek için model güdümlü bir yaklaşım benimsenmiştir. Bu sayede yazılım ekibi tarafından yazılımların geliştirilmesine referans yazılım mimarisine uygun olarak başlanmaktadır. Ayrıca yazılım mimari tasarımından kaynak koda geçiş sürecinde harcanan iş gücünden de tasarruf sağlanmaktadır.

Anahtar Kelimeler: Model Güdümlü Yazılım Geliştirme, Yazılım Mimarisi, Referans Yazılım Mimarisi, Alana Özgü Dil Tasarımı, Otomatik Kod Üretme, Deniz Sistemleri

1 Giriş

Günümüzde hızlı ve etkin şekilde, kaliteli yazılım ürünlerinin geliştirilmesi için önceden geliştirilmiş yazılım ürünleri yeniden kullanılmaktadır [1]. Yazılımda yeniden kullanımın sağlanması, yazılımı oluşturan parçaların birbirleriyle uyum içerisinde çalışacağı bir yapının tasarlanmasına bağlıdır.

Referans yazılım mimarisi, yazılımın genel çerçevesini tanımlamaktadır. Yazılımı oluşturan parçaları, bu parçaların birbiriyle ve dış dünya ile etkileşimini tarif etmektedir [2]. Yazılım geliştirme sürecinin merkezinde yer almakta ve yazılım geliştirme yaşam döngüsündeki her türlü faaliyete kılavuzluk etmektedir. Yazılım sistemindeki karmaşıklığın kontrol edilebilmesi için daha basit ve yönetilebilir bir yapı sunmaktadır [3]. Ayrıca referans yazılım mimarisi, yeniden kullanımın artırılmasını da sağlamaktadır. Tekrar kullanılacak varlıklar ve uygulamalarda

ortaya çıkabilecek değişkenlikler de referans yazılım mimarisinde belirlenebilmektedir.

Belirli bir alanda faaliyet gösteren yazılım ekipleri, genellikle yazılım ürünlerini tanımladıkları referans yazılım mimarisine uygun bir şekilde geliştirmektedirler [4]. Referans yazılım mimarisine tekrar kullanılabilir varlıklar ve değişkenlikler ortaya koyulmaktadır [5]. Böylece hem mevcut yazılım ürünlerinde hem de gelecekte geliştirilecek yazılım ürünlerinde yeniden kullanımın sağlanması için uygun ortam hazırlanmış olur.

Bu bildiriye, yazılım mimari tasarımından, yazılım geliştirme çatisının otomatik olarak oluşturulması için uygulanan model güdümlü yaklaşım, bu yaklaşımın sağladığı kazanımlar ve edinilen tecrübeler paylaşılmaktadır. ASELSAN SST Sektör Başkanlığı bünyesinde deniz sistemleri alanında çalışan bir yazılım ekibi tarafından geliştirilen tüm yazılımlar tanımlanan referans yazılım mimarisine uygun olarak tasarlanmaktadır. Referans yazılım mimarisine uygun olarak tasarlanmış yazılım mimarisinden, kodlamaya geçiş sürecini otomatikleştirmek için model güdümlü bir yaklaşım benimsenmiştir. Bu yaklaşımın uygulanmasıyla referans yazılım mimarisine uygun bir şekilde yazılımların geliştirilmesine başlanmaktadır. Bu sayede yazılım mimari tasarımından kaynak koda geçiş sürecinde kod-mimari uyumu sağlanmaktadır. Ayrıca şablon kodlar otomatik üretildiği için geçiş sürecinde harcanan iş gücünden de tasarruf sağlanmaktadır.

Bildirinin bundan sonraki bölümleri şu şekilde düzenlenmiştir: ikinci bölümde sistematik yeniden kullanımın sağlanması için benimsenen yaklaşım hakkında bilgi verilmiştir. Üçüncü bölümde uygulanan model güdümlü yaklaşım ve bu yaklaşımın sağladığı kazanımlar paylaşılmaktadır. Son bölümde yapılan çalışma sonucunda gelinen noktanın değerlendirmesi yapılmış ve gelecek dönemde yapılacak çalışmalardan bahsedilmiştir.

2 Yeniden Kullanım Yaklaşımı

Yazılım geliştirme çalışmalarında sistematik yazılım yeniden kullanımının sağlanması için tanımlanmış olan politika, yöntem, araç ve uygulamaların bütününe, Deniz Savunma Sistemleri Yeniden Kullanım Yaklaşımı (DSS-YKY) adı verilmektedir [6]. DSS-YKY ile sistematik bir yeniden kullanımın sağlanmasına ilişkin tanımlamaların yapılması, belirlenen altyapıların oluşturulması, etkin kullanımın sağlanması, tanım ve altyapıların idame ettirilmesi amaçlanmıştır. Bu kapsamda ortak mimari altyapı ve ortak sistem bileşenleri kullanılarak ürünlerin geliştirilmesi hedeflenmiştir.

DSS-YKY kapsamında yazılım ekibi tarafından geliştirilen yazılımların ortak bir mimari yapı kullanılarak geliştirilmesi için Referans Yazılım Mimari (DSS-RYM) tanımlanmıştır. DSS-RYM, geliştirilen yazılımlar için katmanlı bir mimari yapı ve bu katmanlı yapıya uygun olarak yazılımların geliştirilmesi için yönlendirmeler sunmaktadır. DSS-RYM için tanımlanan kavramsal mimari Şekil 1’de verilmiştir.

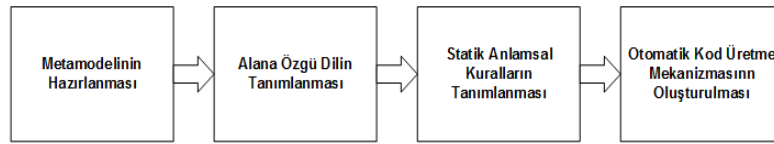


Şekil 1. Referans Yazılım Mimarisi

3 Model Güdümlü Yazılım Geliştirme Yaklaşımı

Model güdümlü yazılım geliştirme yaklaşımı, platformdan bağımsız modeller aracılığıyla çalıştırılabilir kod üretmeyi sağlaması sebebiyle günümüzde tercih edilir [7]. Modeller, üst seviye soyutlama imkânı sunduğu için karmaşık sistemlerin geliştirilmesini kolaylaştırır [8]. Platformdan bağımsız modeller sayesinde aynı model kullanılarak farklı platformlar için otomatik kod üretilebilir. Böylece tasarımdan kodlama sürecine geçişte oluşabilecek hatalar da en düşük seviyeye indirilebilir. Ayrıca, modeller kullanılarak otomatik kod üretilebildiği için kodlama sürecinde harcanan iş gücünden tasarruf sağlanabilir.

DSS-RYM'ye uygun olarak tasarlanmış yazılım mimarisinden, kodlamaya geçiş sürecini otomatikleştirmek için model güdümlü bir yaklaşım benimsenmiştir. DSS-MGY adı verilen bu çalışma kapsamında izlenen yol haritası aşağıdaki şekilde gösterilmektedir. DSS-MGY yol haritasında bulunan adımlara ilişkin açıklamalara bu bölümün alt başlıklarında yer verilmiştir

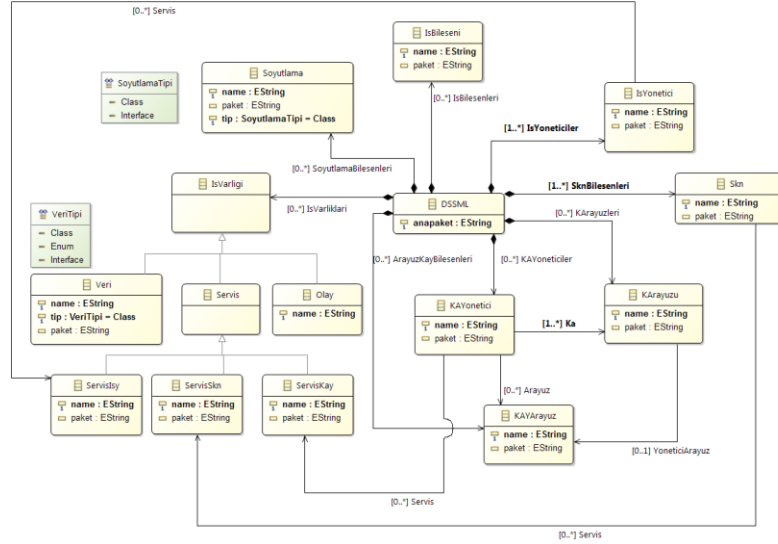


Şekil 2. Model Güdümlü Yazılım Geliştirme Yaklaşımı Yol Haritası

3.1 DSS-RYM Metamodeli

DSS-MGY kapsamında, alana özgü dil tanımlamasında kullanılacak DSS-RYM Metamodeli oluşturulmuştur. Metamodel oluşturmak için EMF (Eclipse Modeling Framework) kullanılmıştır. EMF, ECore adı verilen, modelleme anlamında yazılım alanında çok sık kullanılan, bir meta metamodel sunmaktadır. Kullanıcılar, ECore meta metamodelini kullanarak kendi metamodellerini tanımlayabilirler [3].

DSS-RYM Metamodeli oluşturulurken DSS-RYM’de yer alan kavramlar ve bu kavramlar arasındaki ilişkilerden yola çıkmıştır. Oluşturulan DSS-RYM Metamodeli aşağıdaki şekilde gösterilmiştir.



Şekil 3. DSS-RYM Metamodeli

Şekil 5’de DSS-RYM Metamodelinin elemanları kutularla, elemanlar arasındaki ilişkiler de çizgilerle gösterilmiştir.

DSS-RYM Metamodeli oluşturulduktan sonra somut dilbilgisi üretmek için bir Eclipse eklentisi olan Xtext¹ aracı kullanılmış, Xtext’in var olan metamodelden otomatik dilbilgisi üretme tekniğinden yararlanılmıştır. Otomatik üretilen dilbilgisi üzerinde ufak değişiklikler yapılmış ve oluşturulan bu alana özgü dile DSS-ML adı verilmiştir.

Dilbilgisine göre DSS-ML uygulama modeli, yazılımın ana paket ve bileşen tanımlarından oluşacaktır. Bu bileşenler DSS-RYM’de yer alan kavramlara karşılık gelmektedir.

3.2 DSS-ML Statik Anlamsal Kurallar

Metamodelde ve somut dilbilgisi modelinde elemanlar arası ilişkiler sadece sayısal bağlamda kısıtlanmaktadır. DSS-ML için metamodelde ve somut dilbilgisi modelinde getirilmeyen kısıtlara yönelik bazı statik anlamsal kurallar tanımlanmıştır. Kural tanımlamaları için Xtext’in geçerlilik kontrol mekanizması (validator) kullanılmıştır. Kurallar Xtend programlama dili kullanılarak oluşturulmuştur. DSS-ML modelinden üretilen sınıf isimlerinin de DSS-RYM’de tanımlanan isimlendirme kurallarına uygun olması sağlanmıştır.

¹ Xtext framework, <http://eclipse.org/Xtext>

3.3 Otomatik Kod Üretme

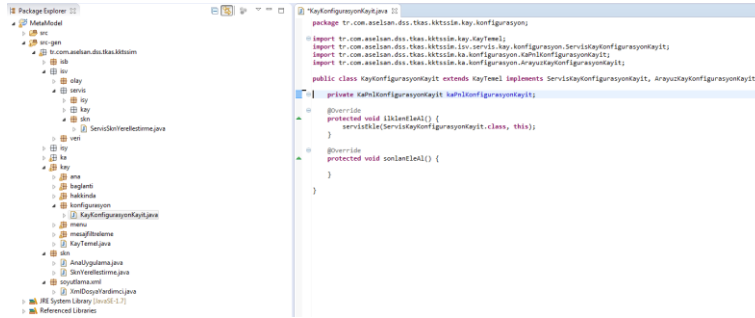
Model-metin dönüşümü, model güdümlü yazılım geliştirme sürecinde anahtar rolü oynamaktadır. Bir model kod, rapor, doküman gibi metin varlıklarına dönüştürülebilmektedir. Bu çalışma kapsamında Xtext DSL olarak tanımlanan DSS-ML dili için Xtend programlama dili kullanılarak Java kodu oluşturma mekanizması kurulmuştur.

Otomatik kod üretme mekanizmasının ilk aşamasında DSS-ML’de bulunan her bir DSS-RYM elemanının içinde yer alacağı paketler yaratılmaktadır. Bu paketlerin içerisine DSS-ML elemanlarına karşılık gelen Java sınıfları koyulmaktadır. Otomatik kod üretme işlemi tamamlandıktan sonra DSS-ML ile oluşturulan uygulama modellerinden üretilen kaynak kodlar, tariflenen paketler altına yerleştirilmektedir.

3.4 Örnek Çalışma

Yapılan tanımlamalar sonrasında, DSS-ML dili kullanılarak yazılım ekibi içerisinde geliştirilen her yazılım projesi için uygulama modeli geliştirilmeye başlamıştır. Bu dilbilgisi kullanılarak oluşturulacak olan modellerin “.dssml” uzantılı dosyalar içerisine yazılması gerekmektedir. Örnek çalışma kapsamında bir simülör projesinde DSS-ML dili kullanılarak uygulama modeli oluşturulmuştur.

Uygulama modeli oluşturulup otomatik kod üretme işlemi tamamlandıktan sonra simülör projesinin kaynak kod dizini altına şablon kodlar eklenmiştir. Şekil 4’de otomatik oluşturulan paket yapısı ve üretilen şablon yönetici sınıfın bir örneği verilmiştir.



Şekil 4. Otomatik Üretilen Paket Yapısı ve Kodlar

Simülör projesi kapsamında DSS-ML dili kullanılarak projenin başlangıçta kullanılacak yazılım geliştirme çatısı üretilmiştir. Elde edilen çıktılar aşağıdaki tabloda paylaşılmıştır.

Tablo 1. Örnek Çalışma Çıktıları

Metrikler	Çıktılar
Paket Sayısı	38
Sınıf Sayısı	89
Kod Satır Sayısı	502

Tablo 1’de verilen çıktılar yazılım ekibi üyeleri tarafından incelenmiştir. Ekip üyelerine göre DSS-ML dili kullanılarak elde edilen çıktıların elle oluşturması durumunda yaklaşık 8.5 saatlik iş gücü harcanması gerektiği ön görülmüştür. Bu işlem DSS-ML ile 0.5 saatte tamamlanmıştır. İş gücü tasarrufunun yanı sıra DSS-RYM’ye uygun olarak kaynak kodlar üretildiği gözlemlenmiştir.

4 Sonuç

Yeniden kullanım çalışmaları kapsamında ortak bir referans yazılım mimarisi oluşturulmuştur. Yazılım ekibi tarafından geliştirilen yazılımların referans yazılım mimarisine uygun olması için yazılım tasarımı DSS-ML uygulama modelleri üzerinden yapılmaya başlanmıştır. DSS-ML otomatik kod üretme mekanizmasıyla üretilen şablon kodlar sayesinde tasarımdan kodlamaya geçiş süreci otomatikleştirilmiştir. Böylece yazılımların geliştirilmesine referans yazılım mimarisine uygun bir şekilde başlanmaktadır. Ayrıca şablon kaynak kodların otomatik üretilmesiyle yazılım mimari tasarımından kaynak koda geçiş sürecinde harcanan iş gücünden tasarruf sağlanmıştır.

Önümüzdeki dönemde tasarımdan kodlamaya geçiş sürecinin yanı sıra DSS-YKY çalışmaları kapsamında Yazılım Geliştirme Çatısı Üretici sürecinin tamamını kapsayacak otomasyon çalışmalarının yapılması hedeflenmektedir. Ayrıca, uygulama modelinden UML diyagramlarını içeren tasarım dokümanlarının da otomatik üretilmesi planlanmaktadır.

Kaynaklar

1. W. Frakes ve K. Kang, “Software Reuse Research: Status and Future,” *IEEE Transactions on Software Engineering*, Cilt 31, No. 7, pp. 529 - 536, 2005.
2. R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole ve M. Bone, “The Concept of Reference Architectures,” *Wiley Periodicals, Inc. Syst Eng*, cilt 13, no. 1, p. 14–27, 29 Ocak 2010.
3. B. M. Yıldız ve B. Tekinerdogan, “Küresel Yazılım Geliştirme Projeleri İçin Alana Özgü Dil,” *Ulusal Yazılım Mühendisliği Sempozyumu*, Ankara, 2011.
4. E. Kahraman, T. Ipek, B. Iyidir, C. Bazlamaci ve S. Bilgen, “Bileşen Tabanlı Yazılım Ürün Hattı Geliştirmeye Yönelik Alan Mühendisliği Çalışmaları,” *Ulusal Yazılım Mühendisliği Sempozyumu*, İstanbul, Türkiye, 2009.
5. R. Atas ve O. Kalipsiz, “Servis Tabanlı Yazılım Ürün Hattı Mimarileri,” *Fırat Üniversitesi Elektrik-Elektronik ve Bilgisayar Sempozyumu*, Elazığ, Türkiye, 2011.
6. B. Durak, E. K. Akbiyik ve I. O. Yigit, “Deniz Savunma Sistemleri Alanında Sistematik Yazılım Yeniden Kullanım Yaklaşımı,” *Ulusal Yazılım Mühendisliği Sempozyumu*, Güzelyurt, KKTC, 2014.
7. E. K. Karatas ve B. Iyidir, “Yazılım Ürün Hattı Yaklaşımında Model Güdümlü Uygulama Mühendisliği,” *Ulusal Yazılım Mühendisliği Sempozyumu*, İstanbul, Türkiye, 2009.
8. A. Goknil ve Y. Topaloglu, “Yazılım Geliştirmede Model Dönüşümü ve Model Dönüşüm Dilleri,” [Çevrimiçi]. Available: www.emo.org.tr/ekler/95d2a757eb34bf2_ek.pdf. [17 Nisan 2015 tarihinde erişilmiştir].