

E-Üniversite için Esnek bir Framework Geliştirilmesi ve Uygulanması

Fatih Özbek¹, Murat İnce², Murat Turhan¹, Hasan Hüseyin Önder³

¹ Akdeniz Üniversitesi, Bilgi İşlem Daire Başkanlığı, TR07058 Antalya, Türkiye

² Süleyman Demirel Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, TR32200 Isparta, Türkiye

³ Süleyman Demirel Üniversitesi, Mühendislik Fakültesi, TR32200 Isparta, Türkiye

fatih@akdeniz.edu.tr, muratince@sdu.edu.tr, muratturhan@akdeniz.edu.tr, hasanhuseyinonder@sdu.edu.tr

Özet: Üniversitelerde öğrencilerin, akademisyenlerin ve diğer idari personelin kullanması ve üniversitedeki işlemlerin yapılabilmesi için çoğu zaman birbirinden bağımsız birçok otomasyon yazılımı bulunmaktadır. Resmi yönetmelikler, kanunlar sık sık değiştiğinden dolayı bu otomasyon yazılımlarının da sürekli güncel tutulması gerekmektedir. Bir üniversitenin her bir otomasyonu hizmet alımı ile sağlaması her güncellemede ek maliyetler getirmektedir. Güncelleme işlemlerinin üniversitenin bilgi işlem birimince yapılması ise güncelleme şekli ve büyüklüğüne göre oldukça zaman alabilmekte ve yeni alt yapısal değişiklikler getirebilmektedir. Sonuç olarak her yeni değişiklik talebi zaman, finansal ve insan kaynaklarının verimsiz ve gereksiz kullanılmasına yol açmaktadır. Bu tür olumsuzlukları önlemek için üniversitelerde kullanılan otomasyon sistemlerinin yapısı esnek, hızlı, dayanıklı ve kolay kullanılabilir bir şekilde hazırlanmalıdır. Bu çalışmada Akdeniz Üniversitesi'nde kullanılan otomasyon sistemlerinin geliştirilmesi için esnek bir yazılım alt yapısı oluşturulmuş ve bu alt yapı kullanılarak otomasyon sistemleri geliştirilmiştir. Bu yazılım alt yapısında MVP(Model-View-Presenter) tasarım deseni, WCF(Windows Communication Foundation) servis yönelimli mimari, çeşitli nesne tabanlı programlama tasarım desenleri ve çok katmanlı yazılım mimarileri kullanılmıştır.

Anahtar Sözcükler: framework, mvp, wcf, tasarım deseni, kural motoru

Developing and Implementing a Flexible Framework for E-University

Abstract: In this study, a flexible framework was developed and implemented for the automation system needs of Mediterranean University which is called "e-Akdeniz". There are a lot of automation systems in university in order to meet the requirements of students, academicians and administrative staffs. These systems are changing frequently and continuously according to the official regulations. Therefore, changing these systems frequently cause time, financial and human resource waste. In order to prevent these wastes, a flexible, rapid and easy to use framework was developed and implemented using MVP (Model-View-Presenter) Architecture, WCF (Windows Communication Foundation) Service Oriented Architecture, Object Oriented Programming Approach and Software Design Patterns. Moreover, e-Akdeniz framework was developed on N-Tier Software Architecture.

Keywords: framework, mvp, wcf, design pattern, rule engine

1. Giriş

Güncel problemlerin mevcut yazılımlarla çözülmesi yazılım geliştirme işlemlerinde sıkça başvurulan yöntemlerdendir. Daha önceden edinilen bilgilerin ve deneyimlerin yeni bir problem karşısında kullanılabilmesi, bunların kütüphane şeklinde etkin olarak saklanmasıyla mümkün olmaktadır. Bu tür kütüphaneler birbirleriyle ilişkili olan fonksiyonlar, nesnelere, sınıflar, prosedürler gibi yazılım geliştirilmesinde kullanılan yöntemlerden oluşmaktadır. Yazılım geliştiricileri daha önceden kendilerinin veya başkalarının hazırladıkları kütüphaneleri kullanarak yeni yazılım uygulamaları geliştirmekte ve ihtiyaçlara cevap verebilmektedirler [1]. Bir veya belirli amaçları yerine getirmek için kütüphaneler hazırlanmakla beraber daha önceden hazırlanmış kütüphaneler birlikte kullanılarak yazılım sistemlerinin temelini oluşturulan uygulama çatıları (Framework) geliştirilmektedir. Framework kullanımı yazılım

projelerinin geliştirilme sürecini hızlandırarak ve düzenli, anlaşılır temiz kod yazılmasını sağlayarak yazılım standardı oluşturur. Yazılım uygulamalarında framework kullanmanın faydaları[2];

- Sistemin birçok yazılımcı tarafından testi yaptırılarak güvenlik ve performans gibi sorunların giderilmesi
- Yazılım geliştirme sürecinin kolaylaştırılması ve hızlandırılması
- Düzenli ve anlaşılır temiz kod yazımı sağlanarak standartlaşmanın oluşturulması
- Proje ekibinin görev paylaşımının kolaylaşması

- İlerdikleri veritabanı adaptörleri sayesinde farklı veritabanları ile kolayca çalışabilmeleri

Framework bir yazılım için varsayılan davranışları belirleyerek yazılım geliştiricilerin yeni uygulamalar geliştirirken kolayca kod yazabilmelerini sağlamaktadır. Yazılım geliştirmek için gerekli olan kodların tekrar tekrar yazılması framework kullanılarak bu işlemlerin her defasında yeniden tanımlanmasına gerek kalmadan sağlanmaktadır. Bir framework'de yapılabilecek değişikliklerin kolay ve hızlı bir şekilde yapılabilmesi için katmanlı mimariler kullanılmaktadır. Yazılım geliştiriciler tarafından sağlanan hazır framework'ler belirli kütüphanelerden oluştukları için değişik yazılım ihtiyaçlarına çoğu zaman cevap verememekteyler. Böyle durumlarda daha farklı framework'lere ihtiyaç duyulacağından yazılım geliştirme süreci daha fazla bilgi ve zaman gerektireceğinden yazılım süreci zorlaşmaktadır. Bu tür sorunları çözebilmek için çalışmamızda hızlı, dayanıklı, kolay ve anlaşılır temiz kod yazımı içeren esnek bir yazılım çatısı olan e-Akdeniz framework geliştirilmiştir. Ayrıca bu framework ile üniversitenin ihtiyaç duyduğu yazılım otomasyonu uygulamalarının çok kısa bir sürede maksimum verimlilikte ve doğrulukta hazırlandığı gösterilmiştir.

1.1. MVP (Model-View-Presenter) Tasarım Deseni

Model-View-Presenter(MVP) yazılım uygulamalarının sunum katmanında bulunan bir mimari tasarım deseni [3]. Bu tasarım deseni ilk olarak 1990'lı yıllarda Taligent tarafından C++ ve Java dilleri ile uygulanmıştır. MVP'de model ve view denilen yapılar tamamen ayrılmıştır. Model kısmı verilerin alınması, tutulması ve iş mantıklarının kullanılmasını sağlamaktadır. View ise kullanıcı arayüzü olarak tanımlanmaktadır. Bu kısımda kullanıcı istekleri ve hareketleri alınarak işlenmek üzere presenter'a iletilir ve dönen veriler de arayüzlerin güncellenmesi ile kullanıcıya iletilir. Presenter model ve view arasında bulunarak bir köprü vazifesi görmektedir. Model'den gelen verileri presenter'a aktarmaktadır. Aynı şekilde presenter'dan gelen verileri de view'e aktararak arayüzlerin güncellenmesini sağlamaktadır (Şekil 1).

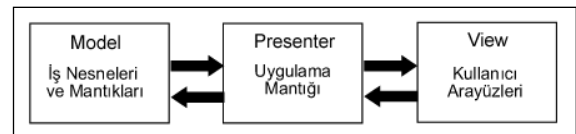
1.2. WCF (Windows Communication Foundation)

Servis odaklı mimari (Service Oriented Architecture - SOA) yazılım sistemlerinin iş süreçleri kapsamında gruplayarak yeni sistemlerin

geliştirilmesini ve bütünleştirilmesini sağlayan bir yazılım tasarım yöntemidir. SOA yazılım süreçlerinde kullanılan farklı uygulamaların ve teknolojilerin birbirleriyle uyumlu olmamalarına rağmen veri alışverişi yapabilmelerini sağlamaktadır[4]. Bu sebeple yazılımcılar tarafından geliştirilen kodların ve uygulamaların tek bir iletişim kanalını kullanabilmeleri için SOA farklı bölümlere ayrılmıştır[5][6]. Servis odaklı mimari yöntemini uygulamak için Microsoft tarafından Windows Communication Foundation (WCF) geliştirilmiştir. WCF yüksek genişleyebilir kapasitesiyle ihtiyaç duyulan uygulama gereksinimlerine çok hızlı, esnek ve kolay bir şekilde cevap verebilmektedir [7]. WCF'in iç mimarisi oldukça karışık olmasına rağmen uygulaması kolaydır ve sisteme getirdiği katkılar oldukça fazladır.

1.3. Yazılım Tasarım Desenleri

Tasarım desenleri yazılım geliştirirken karşılaşılan sorunlara esnek, genişletilebilir, yeniden kullanılabilen çözümler getiren ve çeşitli durumlarda sorunların nasıl çözüleceğini gösteren yöntemlerdir[8]. Tasarım desenleri daha çok nesneye dayalı programlama da sınıf ve nesnelerin birbirleriyle olan ilişkilerini açıklamaktadırlar. Tasarım desenleri sayesinde yazılımdaki kodların tekrar tekrar kullanılabilir olması, sistemin işleyişini çok fazla etkilemeden kolayca sistemden çıkarılıp eklenerek yazılıma büyük bir esneklik kazandırmaktadır. Böylece yazılım uygulamalarına genişletilebilir özellik katılarak yazılım kalitesinin artması, zaman ve iş gücünden tasarruf sağlanmaktadır. Tasarım desenleri kurucu, yapısal ve davranışsal tasarım desenleri olarak gruplandırılmaktadır. Kurucu tasarım desenleri; nesne oluşturma görevini sistemden bağımsız hale getirerek, sisteme hangi nesnenin ne zaman ve nasıl oluşturulacağı konularında esneklik sağlar. Yapısal tasarım desenleri; nesne ve sınıfların daha karmaşık yapıları oluşturmak için nasıl birlikte kullanılabilceği bilgisini sağlamaktadırlar. Davranışsal tasarım desenleri; nesnelerin kendi aralarında nasıl iletişim kurduklarını, çalışma zamanında karışık iş akışlarının kontrol edilmelerini sağlamaktadırlar. En çok kullanılan ve bilinen tasarım desenleri Tablo 1'de verilmiştir.



Şekil 1: MVP yapısı

Kurucu Tasarım Desenleri	Yapısal Tasarım Desenleri	Davranışsal Tasarım Desenleri
Fabrika Yordamı	Adaptör	Arabulucu
Kopya Nesne	Ağaç Yapısı	Durum
Kurulum Nesnesi	Dekorasyon	Gözlemci
Soyut Fabrika	Hafif Ağırlık	Hatırlatıcı
Tek Nesne	Köprü	Kalıp Yordam
	Ön Yüz	Komut
	Özdeş Nesne	Sorumluluk Zinciri
		Strateji
		Yorumlayıcı
		Ziyaretçi

Tablo 1: Tasarım Desenleri[8]

2. E-Akdeniz Framework Tasarımı

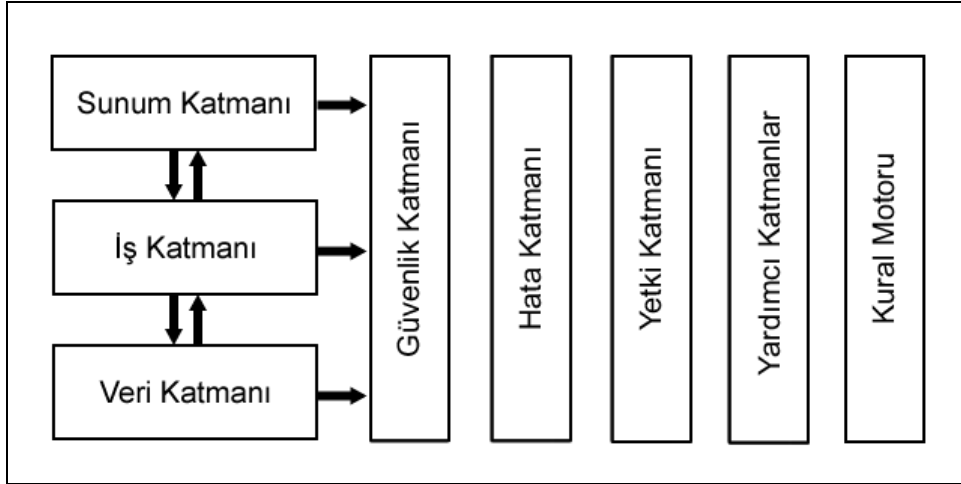
2.1. Sistemin Genel Tanımı

Üniversite otomasyon sistemleri binlerce kişi tarafından sürekli olarak kullanılmaktadır. Sistemde oluşabilecek herhangi bir problem bütün kullanıcıları etkileyebileceği için bu sistemlerin geliştirilmesi, bakım ve idamesi oldukça önemlidir. Örneğin, öğrenci otomasyon sistemi dönem başlarında öğrenci ders kayıt işlemlerinden dolayı çoğu zaman yeterli gelmemekte ve sistem dışı kalabilmektedir. Bu nedenden dolayı işlemler aksayabilmekte ve hatta akademik dönem başlangıç tarihi ileri tarihlere alınabilmektedir. Bu da beraberinde hem ekonomik hem de sosyal sıkıntılar getirebilmektedir. Benzer şekilde üniversite personel bilgi sistemi de dayanıklı ve düzgün çalışmalıdır çünkü hem akademik hem de idari personelin sistem yetkileri, komisyon ve kurul bilgileri, ek ders ücretleri, fazla mesai ücretleri gibi özlük haklarını etkileyebilecek birçok bilgi bu sistem tarafından işletilmektedir. Bu nedenlerden dolayı üniversite otomasyon sistemleri stabil, dayanıklı, birbiriyle uyumlu çalışabilen, kararlı, hızlı ve esnek bir yapıda olmalıdır. Bu özellikleri temel alan e-Akdeniz framework geliştirilmiş ve üniversitenin otomasyon sistemleri bu framework ile yeniden yazılmıştır.

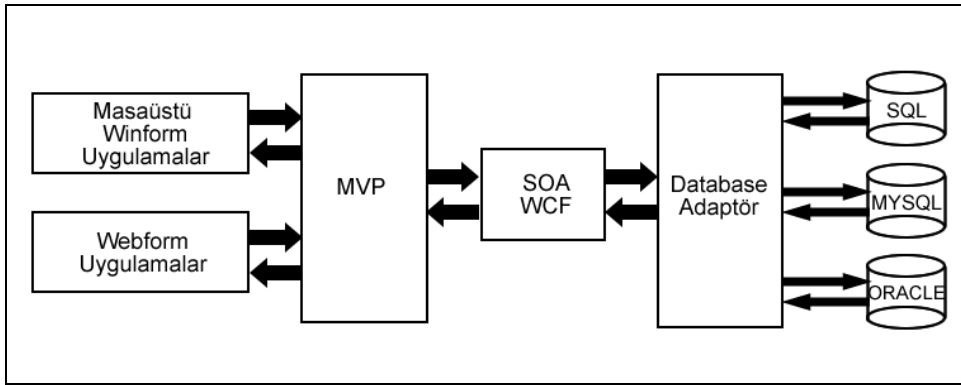
2.2. Sistemin Mimarisi

E-Akdeniz framework çok katmanlı yazılım mimarisi temel alınarak C# dili kullanılarak .NET platformunda geliştirilmiştir. Framework’de yazılım sistemlerine esneklik ve kalite artışı sağlayan tasarım desenleri ve teknolojileri kullanılmıştır. Sistemin esnek ve genişletilebilir olmasındaki en önemli etken çok katmanlı yazılım mimarisi ile geliştirilmiş olmasıdır (Şekil

2). Sunum katmanının da MVP yapısı etkin bir şekilde kullanılmıştır (Şekil 3). Model WCF servis aracılığı ile veritabanı adaptöründen aldığı verileri presenter’a aktarmaktadır. Presenter ise iş mantığına göre view’in bağlı olduğu arayüzleri güncellemektedir. Bu ara yüzler masaüstü uygulamalardaki winform’lar olabileceği gibi webform temelli web uygulamaları da olabilmektedir. Bu üniversite içindeki ihtiyaçlara göre hangisinin en uygun olduğuna göre değişiklik göstermektedir. Örneğin, sadece özlük hakları gibi personel bilgilerinin görülmesinin, işlem yapılmasının istendiği yerlerde personel işlerinin kullanımına masaüstü uygulamalar yapmak daha mantıklıdır çünkü verilerin güvenliği önemlidir (Şekil 4). Benzer şekilde personellerin sisteme girerek kişisel bilgilerini görmeleri, bazılarını güncellemeleri, maaş bordro bilgilerini görebilmeleri gibi durumlarda internet bağlantısı olduğu her hangi bir yerden bu işlemleri yapabilmeleri gerekebilir. Böyle durumlarda web uygulamaları da MVP ile yapılmaktadır. Gerek masaüstü uygulamalar gerekse web uygulamaları MVP ile WCF servise kullanıcı isteklerinin ve işlemlerini aktarmaktadırlar. Bu istekler server uygulaması içerisinde bulunan “database adaptör” yardımı ile herhangi bir veri tabanına bağlanabilmektedir. Sistem kolayca Sql, Mysql ve Oracle gibi veritabanlarına dönüştürülebilmektedir. Bu da veri tabanı seçiminde bir esneklik sağlamaktadır. Veritabanı adaptörü kısmında kurucu tasarım desenlerinden fabrika yordamı, kurulum nesnesi, tek nesne ve soyut fabrika; yapısal tasarım desenlerinden ise adaptör ve dekorasyon tasarım desenleri kullanılmıştır. Bu tasarım desenleri sayesinde kodlama kolaylaşmış, hızlı, stabil ve standart bir sistem geliştirilerek yazılım kalitesi arttırılmıştır.



Şekil 2: Sistemin çok katmanlı mimari yapısı



Şekil 3: Sistemin genel yapısı

Personel No: MURAT TURHAN TC.No: [REDACTED] Kadro Birimi: Kulak Burun Boğaz Hastalıkları Anabilim Dalı Ünvanı: Yardımcı Doçent Doktor

Çalışma Tipi: Akademik Personel Eski Sicil No: [REDACTED] Özlük Hak Kanunu: 2547 SK.
 Kadro Birimi: Kulak Burun Boğaz Hastalıkları Anabilim Dalı Kadro Derecesi: 3 Sigorta Kurumu: Emekli Sandığı
 İşe Başlama Tarihi: [REDACTED] Asalet Alış Tarihi: [REDACTED] Doğum Tarihi: [REDACTED]
 Önceki Soyadı: [REDACTED] Doğum Yeri: ÇAMLIHEMŞİN Pasaport No: [REDACTED]

Birim	Pozisyon	Görev Şekli	Kadroya Kargılık Pozisyon
Kulak Burun Boğaz Hastalıkları Anabilim Dalı	Doçent Doktor	Asaleten	<input checked="" type="checkbox"/>
Kulak Burun Boğaz Hastalıkları Anabilim Dalı	Yardımcı Doçent Doktor	Asaleten	<input type="checkbox"/>

Özlük Bilgileri

Personel Kadro Bilgileri

Personel No: 5 Çalışma Tipi: Akademik Personel
 Eski Sicil No: [REDACTED] SSK No: [REDACTED]
 Başkur No: [REDACTED] Emekli Sandığı Sicil No: [REDACTED]
 İşe Başlama Nedeni: Lütfen seçiniz... Üniversitede İşe Başlama Tarihi: [REDACTED]
 Memuriyete Giriş Tarihi: [REDACTED] Asalet Alış Tarihi: [REDACTED]
 Görev Süresi Son Tarihi: [REDACTED] Ünvan Alış Tarihi: [REDACTED]
 Kadro ve Derecesi: Kulak Burun Boğaz Hastalıkları Anabilim Dalı 3. Derece Yardımcı Doçent Doktor

Kullanıcı Adı: muratince Bilgisayar Adı: ASUS IP Adresi: [REDACTED]

Şekil 4: Personel Otomasyonu Masaüstü Uygulaması

2.3. Kural Motoru

Bir uygulama geliştirilirken iş kuralları genellikle kaynak kodun içerisine onlarca ve hatta uygulamanın büyüklüğüne göre binlerce kod satırından oluşabilmektedirler. Yeni bir iş kuralı olduğunda veya mevcut kurallarda bir değişiklik yapılacağına sistemin kaynak kodlarda değişiklik yapılması gerekmekte ve bu yapılan değişikliklerin güncellenerek yeni bir versiyon şeklinde tüm client'lara yeniden kurulması gerekmektedir. Böyle bir durumda çalışan bir sistemin belirli bir süre için bile durdurulması gerekmekte ve bu da binlerce kişinin kullandığı bir yazılımın kullanılamaz hale gelmesine yol açmaktadır. Sonuç olarak yapılması gereken işler aksamakta, maddi ve manevi kayıplar meydana gelebilmektedir. Bütün bunların önüne geçmek için çalışan sistemin işleyişine hiçbir sorun yaratmadan yapılması gereken değişikliklerin yapılması kural motoru denilen yapı ile sağlanmaktadır. Örneğin, sınıf geçme notunun 2009-2011 kayıtlı öğrenciler için 70, 2012 kayıtlı öğrenciler için ise 65 olduğu bir kanun değişikliğinde sistemin kaynak kodlarını değiştirmeden veri tabanına dinamik kütüphane linkleri (dll) veya değerler şeklinde kurallar eklenir. Öğrencilerin kayıt yıllarına göre hangi geçme notunun kullanılacağı bu kurallar veya dll içinde belirtilmelidir. Sistem o anda öğrencinin kayıt yılını parametre olarak dll'e veya kurala göndererek sonucu döndürebilmektedir. Hazırlanan kural motoru kaynak kodlarda hiçbir değişiklik yapmadan kurallara parametreler yollayarak farklı sonuçlar elde edebilmektedir. Kural motorunun bu yapısı sayesinde verimlilik artırılmış ve sisteme büyük bir esneklik kazandırılmıştır.

3. Sonuç

Bu çalışmada kanunların ve yönetmeliklerin çok sık değiştiği üniversiteler gibi kurumlarda kullanılabilir olan esnek bir yazılım alt yapısı

oluşturulmuştur. Böyle bir alt yapının oluşturulmasında gerekli olan teknolojiler hakkında bilgiler verilmiş ve bunların geliştirilen sistem içerisinde nasıl kullanıldıkları anlatılmıştır. Bu esnek yazılım alt yapısı kullanılarak hem masaüstü winform uygulamalar hem de webform uygulamalar geliştirilmiştir. Böylece geliştirilen sistemle herhangi bir arayüz teknolojisine bağımlı kalmadan istenilen uygulamalar hızlı ve dayanıklı bir şekilde geliştirilmektedir.

4. Kaynaklar

[1] Kim, A. J., Kim, H, K, Park, N., 1998, Development of PDM Framework and Customization Environment

[2]<http://www.webteders.com/blog/detay/Framework-Nedir.-Avantajlari-Nelerdir./12>

[3] Taligent, 1996, MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java

[4] Newcomer, E., Lomow, G., 2005, Understanding SOA with Web Services, Addison Wesley, ISBN 0-321-18086-0

[5] Bell, Michael, 2008, Introduction to Service-Oriented Modeling, Service-Oriented Modeling: Service Analysis, Design, and Architecture, Wiley & Sons. ss. 3, ISBN 978- 0-470-14111-3

[6] Erl, Thomas, 2005, Service-oriented Architecture: Concepts, Technology, and Design, Upper Saddle River: Prentice Hall PTR, ISBN 0-13-185858-0

[7] Zhang, W., Li, J., 2010, Research and Application of WCF extensibility

[8]<http://www.tasarimdesenleri.com/core/home.jsp>