

## VERİ YOĞUN WEB UYGULAMALARI MODELLEME VE TASARIM KONULARI

### Ali KAYA

Hava Lojistik Komutanlığı  
Bil.Sis.Glş. ve Des. D.Bşk.lığı  
Etimesgut, Ankara  
alikaya72@hotmail.com

### Sefer KURNAZ

Hava Harp Okulu Komutanlığı  
Havacılık ve Uzay Teknolojileri Enstitüsü  
Yeşilyurt, İstanbul  
s.kurnaz@hho.edu.tr

### ÖZET

Günümüzde Web üzerinde çalışan programlara olan ihtiyacın ve buna paralel olarak geliştirme çözümlerinin artmasıyla, Web uygulamaları geliştirilirken hangi teknolojilerin seçilip kullanılması gerektiği ve bu teknolojilere uygun hangi modellemenin kullanılması gerektiği çözümlenmesi gereken temel problemlerden biri olmuştur.

J2EE ortamı karmaşık ve çok sayıda teknoloji içermesi sebebiyle başarılı bir uygulama geliştirmek bir çok bilgi ve tecrübeye sahip olunmasıyla mümkün olabilir.

Bu çalışmada, J2EE ortamında veri yoğun web uygulamaları geliştirirken karşılaşılan bazı problemlere ve bunlara getirilen çözüm önerilerine yer verilmiştir. Veri yoğun web uygulamaları geliştirmeye başlamadan burada sayılan problemlerin bilinmesi geliştiricilerin daha hızlı ve problemsiz yazılım geliştirmesine önemli ölçüde yardımcı olacaktır. Bazı verilen çözüm önerileri ise genel olarak uygulama geliştirmeye yöneliktir.

**Anahtar Kelimeler:** J2EE, Modelleme, Web, Yazılım Süreci

### ABSTRACT

In last ten years high proportion increase of web-based software developments and paralld processes results high demand of know-how about which web-technology will be used in developings and which modelling will be suit requirements that fits currently used web-technology. Also this technology and modelling choice becomes one of the basic question that must be solved.

J2EE is a complex and too much technology included structure so to be successful one must have so many experience and too much information about the subject.

In this study the problems faced while developing data-intensive projects and answers related to this questions, in J2EE, outlighted. Before initiating a data-intensive project, developers information about these problems and answers will help the project's fast and reliable process management. Some of the suggestions can be useful in general application development.

**Key Words:** J2EE, Modelling, Web, Software Process

### 1. GİRİŞ

Java 2 Kurumsal Sürüm (Java 2 Enterprise Edition-J2EE), ölçeklenebilirliği, ortam bağımsız olması ve başarımı ile yıllardır geliştiricilerin veri yoğun uygulamalarda tercih sebebidir. J2EE ortamında uygulama geliştirmede özüm kompozisyonları

sayılamayacak kadar çoktur. J2EE ortamında üretilen çoğu uygulamalar bazı şartları ve tecrübeleri gözetmeden yapıldığından başarımı (performansı) ve bakımı yapılabilirliği düşük uygulamalar olmaktadır. J2EE ortamında başarılı uygulama yapmak amacıyla yürütülen çalışmalarda karşılaşılan problemler ele alınmıştır.

## 2. YAZILIM TEKNOLOJİLERİNDEKİ HIZLI GELİŞİM

Masaüstü uygulama geliştirmede karşılaşılan problemlere nazaran yapısı gereği bir web göz atıcı (browser) içinde çalıştırılmak zorunda olan web uygulamaları geliştirmek daha problemlidir ve daha fazla çaba gerektirir. Ancak yeni geliştirilen araçlar ile web uygulamalarına yönelik hızlı geliştirme ortamları (Rapid Application Development- RAD) sunulmaya başlanmıştır. Örneğin, ASP.NET ile birlikte gelen WebForm kavramı Web programcılığına yeni ve daha kolay geliştirmeyi doğuran bir yaklaşımı getirmiştir. WebForm'lar kullanılarak yapılan Web programları normal Windows programlamadan farksız hale gelmiştir. En göze çarpan özellikleri aşağıda sıralanmıştır.

- Bir Web göz atıcıda en çok ihtiyaç duyulan yapılar bileşen halinde sunulmuştur.
- Bir Web bileşeni normal bileşenden farksız olarak, bir Web form üzerine sürüklenip bırakılabilmekte, özellik penceresinden istenen özelliklerin değerleri değiştirilebilmekte, olayların metodlarına kodlar yazılabilmektedir.
- Herhangi bir Web bileşenine veri bağlama hazır halde sunulmuştur.
- Bileşenlerin HTML sayfa içindeki görünüm kodları ile program kodları ayrı ayrı dosyalarda tutularak model görünüm denetleyici (Model View Controller-MVC) yaklaşımı korunmuştur.
- Bileşen teknolojisine geçmenin bir avantajı olarak, üçüncü parti bileşen üreticileri daha özel amaçlı ve karmaşık bileşenleri üretebilmekte ve bu ürünler geliştiriciler tarafından kolayca kullanılabilirlerdir.

Yazılımın başarımından daha önemli bir konu, yazılımı geliştiren modelleyen kişilerin bu teknolojileri anlamakta ve kullanmakta harcadıkları zamanın kısa olması gerektiğidir. Bu sebeple Java dünyasının sonuçta bir uygulama mantığı olan bu WebForm'larını adapte etmeleri kaçınılmaz olacaktır. Oracle firması JDeveloper 9.0.5'te sunacağını duyurduğu Oracle Uygulama Geliştirme Ortamı (Application Development Framework-ADF) buna bir örnektir. ADF'in amacı da bu yeni Web geliştirme anlayışının önümüzdeki zaman içinde daha belirleyici ve çok kullanılacağı görüşünü kuvvetlendirmektedir.

## 3. ORTAM SEÇİMİYLE İLGİLİ KONULAR

Özellikle Web programlama ile ilgili ortam seçimlerinde alternatifler neredeyse sayılamayacak kadar çoktur. Bunlar büyük ve küçük çözümler olarak ikiye ayrılabilir. Büyük çözümlerden en bilinenleri şu

an Microsoft'un .NET ortamı ile Sun'ın J2EE ortamıdır. .NET ortamının sunduğu yol tek seçimli, üreticinin belirlediği bir yol olmasına rağmen J2EE'de çözüm alternatifleri sınırlı değildir.

Her ne kadar üreticinin belirlediği yol geliştiricileri sınırlar gözüke de, küçük yazılım firmaları veya Microsoft'a ve onun sunduğu teknolojinin başarımına güven sorunu yaşamayan firmalar için hedefe çok çabuk ulaştırarak avantajları vardır.

Kabaca .Net geliştirme mimarisi şu şekildedir:

(1) sunum katmanında, ASP.NET Web formları;

(2) iş mantığı katmanında iş bileşenleri

(3) veri erişim katmanında veri erişim bileşenleri

Geliştiriciler herhangi bir C#, VB.NET veya ASP.NET kitabını alıp inceledikten ve .NET ortamıyla ilgili Microsoft'un zengin kütüphanesini (MSDN) ellerinin altında bulduktan sonra, geliştirmeye veya entegrasyonla ilgili herhangi bir problem yaşamadan ürünlerini geliştirebilirler; çünkü çözüm yolu tek ve verilen kod örnekleri bu yolda gereken bütün bilgileri sağlamaktadırlar.

J2EE yolu tercih edildiğinde ise hedefe çok çabuk ulaşmak mümkün değildir. En yeni iskeletlerden haberdar olunmalı, bu iskeletler, geliştirme safhasının bütün aşamalarında yardımcı olacak bileşenlere, dokümantasyon desteğine, kolay öğrenilebilir, uygulanabilir bir anlayışa sahip olup olmadıkları iyice araştırıldıktan sonra kullanılmaya başlanmalıdır. Böyle bir ürünün bulunması ve bulunan ürünün bu sayılan özelliklere sahip olup olmadığının araştırılması çok zaman alıcı bir iştir. Ancak Java'nın büyük avantajları düşünülürse (ortam bağımsız olması gibi) böyle bir çabaya değer olarak düşünülebilir.

Küçük çözümlere örnekler olarak, PHP, MySQL, Apache üçlüsü, ASP, IIS ve MSAccess üçlüsü verilebilir. Bu çözümler birbirinin karşılığı olarak görülebilir, biri Linux için tercih edilirken diğeri Windows için tercih edilmektedir. Bütün işlemler bir betik (script) dili ile HTML içine yazılmakla yapılmaktadır. Ancak çok büyük projelerde kullanıldığında bakımı yapılabilmesi zorlaşmaktadır. En önemli avantajı ise kolay öğrenilebilir olmalarıdır; geliştiriciler sadece küçük bir PHP veya ASP kitabı ile bir iki hafta içinde başları ağırmadan çözümler üretmeye başlamaktadır. J2EE gibi karmaşık bir teknolojiye sahip değildirler.

Bunların dışında özel çözümler de mevcuttur. Macromedia'nın ColdFusion buna en iyi örnektir.

Hem küçük hem büyük ölçeklidir. Diğerleri arasında şunlar sayılabilir: Python, Perl, Isapi geliştirme vs.

#### 4. MODELLEME SÜRECİNİN SEÇİMİYLE İLGİLİ KONULAR

Modellemeye başlarken yapılan araştırmalarda her zaman karşılaşılan UML olmaktadır. Ancak geliştiriciler, birkaç UML kitabı okusa bile hemen projelere tatbik etme imkanı bulamazlar, hatta kafa karışıklıklarına bile sebep olabilir.

Özellikle Web programlamada en gelişmiş Bilgisayar Destekli Yazılım Mühendisliği (Computer Aided Software Engineering-CASE) araçlarının bile istenen yeterlilikte olmaması bu araçların kullanımında pek motive edici olmamaktadır. Ayrıca, bir CASE aracının kullanılmasının öğrenilmesinin neredeyse ortamı öğrenmek kadar zor olabilmekte ve zaman alabilmektedir.

Süreçlerden Rational Rose'un RUP'un çok geniş yazılım grupları için ideal olduğu düşünülse de, öğrenmesi ve tatbik edilmesi o kadar kolay bir süreç değildir. Öğrenilmesi ve uygulanması daha kolay olan ICONIX süreç her yazılım grubu tarafından rahatlıkla kullanılabilir.

Son olarak, modellemeden maksat, geliştiricilerin ortak bir şekil dilinde anlaşabilecekleri, modelden koda, koddan modele gidişte uygun bir araç olmasıdır. Özellikle kullanıcının istediğine uygun ürün ortaya çıkarmak, perspektifi kaybetmemek için kullanılan Usecase'ler modelleme de en önemli araçlardan bir tanesidir. Usecase'ler ve diğer modelleme çizgeleri (sağlamlık ve sıra) için sadece çizim amaçlı CASE araçlarının kullanılmasının modelleme sürecinde üretkenliği çok fazla etkilemez.

#### 5. YENİ ARAÇLAR KULLANMANIN FAYDA VE ZARARLARI

Kurumsal olarak yeni teknolojilere adaptasyon her zaman problemlidir ama bir o kadar da kaçınılmaz bir konudur.

Böyle kaçınılmaz bir sorunla karşılaşıldığında, mevcut geliştiricilerin bilgi birikimi üzerine inşa edilebilecek yeni teknolojilerin seçilmesi kritik bir karardır. Örneğin, JDeveloper geliştirme ortamı olarak seçildiyse, yeni gelecek olan Oracle uygulama geliştirme iskeletinin kullanılması bu iskelet daha önceki sunulan teknolojiler üzerine inşa edildiği için hiç problemlidir olmayacaktır. Ancak, klasik JSP, servlet, EJB geliştirmesinde kişisel bir yaklaşım kullanılıyorsa, böyle bir yeni teknolojinin kullanılmaya başlanması pek o kadar da kolay olmayacaktır.

Tamamen farklı bir ortam seçilecekse, yeni ortamın öğrenilmesinin kolay, başarımının yüksek, dokümantasyonunun çok iyi, geliştirmenin herhangi bir safhasında ihtiyaç duyulabilecek bileşenlere sahip olmasına ve bu bileşenlerin kendi arasında çok uyumlu olmasına dikkat edilmelidir. Aksi halde geliştiriciler eskiden kullandıkları ortamın her zaman daha iyi olduklarını düşünecekler ve belki de geçiş başarısız olacaktır.

Günümüzde çözümler birbirine çok yaklaştığından birbirine yakın anlayıştaki çözümler arasında geçiş daha kolaydır. Örneğin, .NET yaklaşımı ile Oracle'ın yeni teknolojisi ADF benzer bir anlayıştır. Bunlar arasında kolayca geçiş yapılabilir. Ancak bir PHP, ASP gibi yalnızca HTML içine betik yazarak uygulama geliştirmeye imkanı veren ortamlarla bileşen teknolojilerini kullanan ortamlar çok farklıdır; geçiş biraz zaman alabilir. Buna örnek olarak mevcut ASP kullanıcılarının NET ortamına geçmemekte direnmesi örnek olarak verilebilir.

#### 6. DOKÜMANTASYONUN YAZILIMI ETKİN OLARAK KULLANMADAKİ ETKİSİ

Bir uygulama geliştirirken çok değişik yerlerde dokümantasyona ihtiyaç duyulur. Böyle durumlarda bir satırlık bir kod bile çok işe yarayacakken bu kod bir türlü bulunamayabilir. Eğer böyle bir durumla karşılaşırsa, dokümantasyon yetersizliğinden bahsedilebilir. Dokümantasyondan maksat ne kadar başarılı olduğunun anlatılmasından çok geliştiriciye her safha da rehber olabilmesidir.

Geliştiriciler, ortamı seçerken dokümantasyon desteğine ayrı bir önem vermelidirler.

#### 7. TEKNOLOJİLERİN ENTEGRASYONU İLE İLGİLİ KONULAR

Geliştiriciler, öncelikle tam bir entegrasyon sunan çözümleri tercih etmelidirler. Başka türlü, bu iş geliştiriciye kaldığında zaman kaybı çok fazla, başarımları da pek o kadar yüksek olmayabilir. Özellikle entegrasyon probleminin çözülmemesi bu ürünü kullanmak isteyen geliştiricinin bu ürünü çıkaran firmanın gelecekteki ürünlerini kullanmaya bir önyargı oluşturması durumunu doğurabilir, bundan hem geliştirici hem de ürünü çıkaran firma zarar görebilir.

Ürünü çıkaran firma eğer birkaç alternatif sunuyorsa, sunduğu alternatifler arasında en uygunun ne olduğu konusunda herhangi bir açıklama yapmaktan kaçınmaktadırlar. Örneğin, bir uygulamayı geliştirmenin dört ayrı yolu olsun, gerçekte en kolay ve problemsiz yol belli iken, geliştiriciye bu bilgi verilmemektedir; bundan maksat diğer ürünlerini kötülememek olsa da bütün bu teknolojiler arasında tercih yapmak zorunda olan geliştirici için bu iş

problemlerle bir hal alır. Teknoloji hakkındaki forumlarda daha önce bu ürünü kullanmış olan veya halihazırda kullanmakta olan kişilerin açık fikirleri bu konuda haftalarca gereksiz çalışmanın önüne geçebilir.

## 8. İLİŞKİSEL VERİTABANLARININ ROLÜ

Bütün veri yoğun uygulamalarda (WEB veya diğer) amaç verinin en hızlı bir şekilde kaydedilmesi ve kaydedilen bilginin yine en hızlı bir şekilde geri getirilmesidir. Bu işlemleri yapan veritabanları arasında en güçlü ve başarılı olanları ise hala ilişkisel veritabanlarıdır; örneğin Oracle'ın Oracle Database, Microsoft'un MSSQLServer, MySQL, SAP vs. sayılabilir.

Ortam geliştiriciyi ne kadar ilişkisel veritabanından soyutlamaya çalışırsa çalışsın, nesne yönelimli programlarda ilişkisel den nesne'ye konuları ne kadar işlenirse işlensin, bir açık kapı bırakarak SQL komutlarının çalıştırılmasına izin vermektedir. SQL temelde dört adet komuttan (Select, Insert, Delete, Update) oluşmasına rağmen, bütün istenen işlemleri gerçekleştirebilecek kabiliyete sahip olağanüstü kolay ve etkin bir araçtır. Bu sebeple, önümüzdeki yıllarda bu dört komut yaşamını sürdürmeye devam edecek gözüküyor.

Her türlü veri ile ilgili işlemde başarılı uygulamalar geliştirmek için geliştiricilerin iyi bir veritabanı tasarımı ve kullanımı konusunda bilgilerinin olması gerekmektedir. Örneğin iyi tasarlanmamış bir veri tabanı üzerinde hangi ortam çalışırsa çalışsın bu ortam ne kadar etkin programlanırsa programlansın yine de uygulama başarılı olmayacaktır.

## 9. SONUÇ

J2EE, geliştirilmiş standartları ve taşınabilir (portable) olmasının büyük avantajlarına rağmen, Java programlamanın karmaşıklığı ve şimdiye kadarki hızlı geliştirme araçlarındaki yetersizlik sebebiyle, bu ortamda uygulama geliştirilmesi zor ve yorucu bir süreçtir.

Java uygulama geliştirmede de tam entegre görsel geliştirme araçlarının geliştirilmesiyle Java'nın bu gereksiz kod yazma ve karmaşıklığı gizlenebilir. Bunu başarma amacıyla olan JDeveloper 9.0.5'de geleceği ifade edilen ADF geliştiricilerin J2EE'de de çok uzmanlık gerektirmeden uygulama geliştirilebileceğini vaat etmektedir.

Geleneksel yaklaşımlar artık son bulmaktadır yeni yaklaşımların geliştirmede ve tasarımda büyük kolaylık ve hız sağlayacağı açıktır.

ICONIX sürecinin öğrenilmesi ve modellemede kullanılmasında daha kolaydır. Özellikle Usecase'lerin

ve sağlamlık çizgilerinin kullanılması modellemede büyük fayda sağlamaktadır.

Sonuç olarak, J2EE ortam bağımsız ve veri yoğun uygulamalar için başarılı bir standart olması sebebiyle geliştiricilerin önümüzdeki yıllarda da kullanacağı ve bu ortamdaki yeni geliştirme araçlarıyla daha hızlı, kolay veri yoğun web uygulamaları üretilebilecektir.

## 10. KAYNAKLAR

- [1] Ambler Scott, <http://www-106.ibm.com/developerworks/library/tip-docusecase.html> Documenting a Usecase
- [2] Ambler Scott, <http://www-106.ibm.com/developerworks/library/tip-essentialuse.html>, Modeling essential Usecases
- [3] Bayern, Shaun. Manning Publications Co., 2002, JSTL In Action
- [4] Bergsten Hengs, <http://www.onjava.com/pub/a/onjava/2002/08/14/jstl1.html>, JSTL 1.0
- [5] Bond, Martin, Sams, 2002, Teach Yourself J2EE in 21 Days
- [6] Cockburn Alistair, <http://members.aol.com/acockburn>, Basic Usecase Template
- [7] Darwin Ian, O'Reilly, 2001, Java CookBook
- [8] Duncan Susan, <http://otn.oracle.com>, Optimizing Development Productivity Using UML in Oracle9i JDeveloper

## ÖZGEÇMİŞLER

### Hv.Uçk.Bkm.Yzb. Ali KAYA

Kayseri'de 1972 yılında doğdu. İlk ve orta öğrenimini Kayseri'de tamamladı. 1986 yılında Işıklar Askeri Lisesi'ne girdi. 1990 yılında Hava Harp Okulunda Bilgisayar Mühendisliği lisans eğitimine başladı. 1994 yılında Hava Harp Okulundan Teğmen rütbesiyle 2nci Ana Jet Üs Uçuş Eğitim Merkezi Komutanlığına atandı.

1995-1996 yılları arasında Hava Teknik Okullar Komutanlığı Gaziemir / İZMİR de Uçak Bakım Okulunda Uçak Bakım Subaylığı kursu aldıktan sonra 1996 yılında Kayseri 12nci Hava Ulaştırma Ana Üs K.lığı Uçak Bakım Komutanlığına atandı. 1996-1997 yılları arasında 12 nci Hava Ulaştırma Ana Üs K.lığı Uçak Bakım Komutanlığı 222 Hat Bakım Komutanı, 1997-2001 yılları arasında 12 nci Hava Ulaştırma Ana Üs K.lığı Uçak Bakım Komutanlığı Motor Pervane

Atelyeler Komutanı görevlerinde bulunduktan sonra Ekim 2001- Ekim 2003 tarihleri arasında Hava Harp Okulu Havacılık ve Uzay Teknolojileri Enstitüsünde Bilgisayar Mühendisliği Ana Bilim Dalında Yüksek Lisans öğrenimini tamamladı.

Evli ve iki çocuk babası olup, İngilizce ve Almanca bilmektedir.

**Hv.Dr.Müh.Alb. Sefer KURNAZ**

1978 Yılında Hava HarpOkulu Elektronik Mühendisliği Bölümünden lisans, Ege Üniversitesi Bilgisayar Mühendisliği Bölümünden yüksek lisans, İstanbul Üniversitesi Bilgisayar Mühendisliği Bölümünden doktora derecesi aldı. Halen Hava Harp Okulu Komutanlığı Havacılık ve Uzay Teknolojileri Enstitüsü Müdürü olarak görev yapmaktadır.