MyFaces Özgür JSF Uyarlaması

Bora Güngören
Portakal Teknoloji
bora@portakalteknoloji.com

Sunum Bilgisi ve Lisans

- Bu sunum 22 Mayıs 2005 günü Linux Kullanıcıları Derneği adına verilmiştir.
- Sunum içeriği GPL ile lisanslıdır. Lisans detaylarını öğrenmek yada sunumu elde etmek için LKD'ye başvurabilir yada bora@portakalteknoloji.com adresine eposta yollayabilirsiniz.

Sunum İçeriği

- Web Uygulaması Kavramı
- Model Görünüm Denetleyici
- Java Server Faces (JSF) Mimarisi
- Sun Referans Uyarlaması
- MyFaces Özgür Uyarlaması

Web Uygulaması Kavramı

- Temel olarak bakarsak HTTP üzerinden çalışan bütün uygulamalar bir **Web Uygulaması** olarak adlandırılabilir.
  - Sun Microsystems, J2EE kapsamında bir web uygulamasını herhangi bir J2EE uygulamasına HTTP tabanlı bir **önyüz** sağlayan uygulama olarak tanımlamaktadır.
  - Java ile yazılan yada Java sistemlerine önyüz oluşturan uygulamalara da kısaca Java web uygulaması diyebiliriz.

Web Uygulaması Kavramı

- Web uygulamaları HTTP üzerinden çalışmak durumunda olduklarından çeşitli HTTP kavramları ile karşılaşmak kaçınılmazdır.
  - **İstemci** ve **sunucu** kavramları uygulamanın nasıl çalışacağına kadar her yerde kullanılacaktır.
    - İstemci tarafında işlevsellik (JavaScript, Applet, vs.) sunulması veya veri saklanması ile aynı işlerin sunucu tarafında yapılması çok farklıdır.
  - HTTP nedeni ile **istek/yanıt** çevrimlerine mahkum oluruz.
    - Kullanıcının çoğu girdisi, zorunlu olarak bir istek/yanıt çevrimine yol açar. Bu yazılımın tasarımını zorlaştırır.

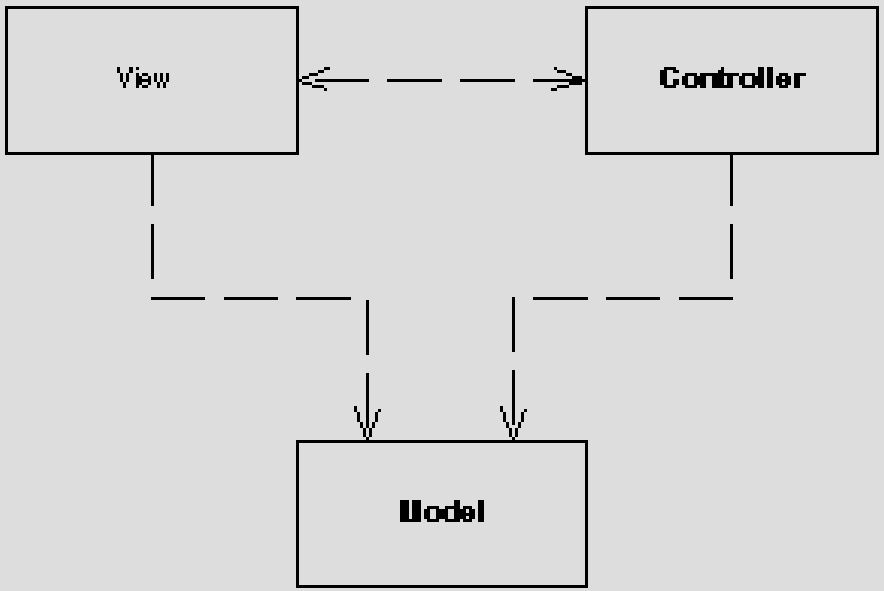
Web Uygulaması Kavramı

- Web uygulamaları HTTP üzerinden çalışmak durumunda olduklarından çeşitli HTTP kavramları ile karşılaşmak kaçınılmazdır.
  - HTTP bir uygulama sözleşmesi olarak **durumsuz** (stateless) bir sözleşmedir. Bunun sonuçları çok önemlidir.
    - Bir noktaya (sayfa) nereden geldiğimizin sözleşme için bir farkı yoktur. Yola bağımlı davranışı planlamak için ileri düzeyde oturum yönetimi tasarımları gerekir.
    - Bir görünümü (sayfayı) yenilediğimiz zaman bir çok işlem yeniden yapılır.

Web Uygulaması Kavramı

- Bu sorunlar elbette çözülebilir ancak web uygulamalarının boyutu büyüdüğünde bu sorunların çözümü için harcanan çaba, esas uygulama için harcananın üzerine çıkar.
- Bir çok görev kritik uygulamanın web uygulaması haline dönüşmesindeki gecikme biraz da bu nedenledir.
  - Finans uygulamaları
  - ERP uygulamaları
  - Askeri uygulamalar

Model Görünüm Denetleyici



- Model Görünüm Denetleyici (Model-View-Controller / MVC) tasarım biçimi görsel bileşen tabanlı uygulamalar için tasarlanmıştır.

Model Görünüm Denetleyici

- Bu biçim 1970'lerde Smalltalk dili içinde zorunlu hale getirilmiş ve 1980'lerden bu yana neredeyse tüm C++ ve ardından tüm Java grafik kullanıcı arabirimi kitaplıklarında yerini bulmuştur.
  - Visual C++ ve MFC'deki **CDocument** (Model) **CView** (View) ve **CApplication** (Controller) sınıfları
  - Java Swing'deki bileşenler ve olay işleme mekanizması

Model Görünüm Denetleyici

- Web uygulamalarında MGD uygulamak masa üstü uygulamalara göre farklı yaklaşımlar ister.
- Bir Java web uygulaması düşünelim.
  - Görünüm olarak (CSS yada XSLT ile biçimlendirilen) bir JSP sayfası.
  - Denetleyici olarak bir Servlet sınıfı.
  - Model olarak Servlet sınıfının eriştiği Java nesneleri.

Model Görünüm Denetleyici

- MGD uygulaması hatalı yapılırsa sadece uygulama mantığında kırılganlık değil güvenlik açıkları da ortaya çıkar.
  - Bu nedenle **uygulama çatıları** kullanarak geliştirme yapmak son derece yaygın ve mantıklı bir çalışmadır.
  - Web uygulamalarında MGD yapısı için yazılan uygulama çatıları her üç rolü de üstlenen sınıflar için üst sınıflar tanımlar.
  - Bunun dışında tipik bazı uygulama teknikleri için ek sınıflar bulunur.

Model Görünüm Denetleyici

- Java/J2EE için popüler **Struts** çatısı ve .NET için **WebForm** yapılarına bakacak olursak, bunlarla yazılan uygulamalar bahsedilen yapıda çalışmaktadır.
  - Ancak bu sistemler hala HTTP' nin belirgin sınırlamalarına göre yazılırlar.
  - Bu nedenle görsel bileşenlere dayanan masa üstü uygulamalarının benzeri biçimde geliştirilemezler.
  - Bu çatıların MGD değil de **Model 2 Web Uygulaması** olarak adlandırılması daha doğru olur.

Java Server Faces

- Java Server Faces, yeni nesil bir önyüz teknolojisidir.
- Aşağıdaki prensipleri uygulamak üzerine kuruludur.
  - Masa üstü uygulamalardaki MGD sistemini ve bileşen modellerini kullanarak geliştirme yapmak.
    - Bu bileşenlerin kurduğu yapıların görünümlerininin seçilen bir önyüz teknolojisine eşlenmesi için gerekli dönüşümün otomatik yapılabilir.
    - Uygulamanın işlemesi için gerekli ek yapı da dönüştürülür.
  - Böylece masaüstü uygulaması gibi çalışan web uygulaması geliştirmiş oluruz.

Java Server Faces

- Bir JSF uygulaması temel olarak fazla bir değişiklik gerektirmeden farklı önyüz teknolojilerine yönelik olarak yeniden yapılandırılabilir.
  - JSP üzerinden HTML
  - WAP
  - XUL

Java Server Faces

- JSF uygulamalarında MGD sistematiği, daha önceki teknolojilerde bulunmayan oldukça gelişmiş bir yapı ile sağlanır.
  - ActionForm'lar yada WebForm'lardan farkın ana nedeni JSF uygulamalarında bileşenler için önyüz teknolojisinden bağımsız nesnelerden oluşan ayrı bir görünüm yapısının bulunmasıdır.
    - Görsel bileşenlerin durumları sunucu tarafında saklanır. Bu da HTTP'nin durumsuz olmasından kaynaklanan sorunları ortadan kaldırır.
    - Bu ayrım sayesinde JSF uygulamalarında doğrulama daha gelişmiş biçimde yapılabilir ve daha çok özelliği olan ileri görsel bileşenler tasarlanabilir.

Java Server Faces

- Bir JSF uygulamasının yaşam çevrimi istek/yanıt çevrimi yerine bir görünüm üzerinde yapılan eylemlere dayanan olayların işlenmesine dayanır.
  - Bu model tamamen masa üstü uygulamalardaki model ile aynıdır.
  - Olay türleri, çeşitli olay işleyiciler, hata durumlarının işlenmesi ile bir JSF uygulaması (sunucu tarafına ulaşınca) pek de bir web uygulaması gibi davranmaz.

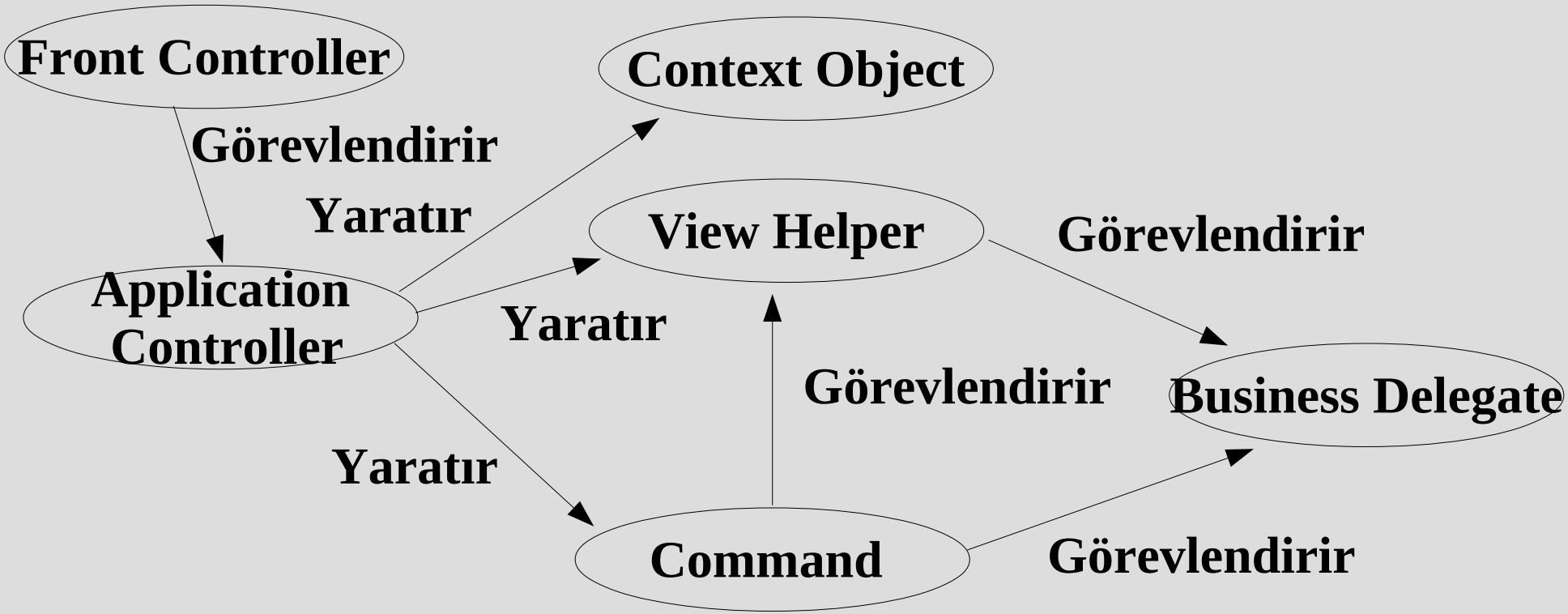
Java Server Faces

- Bir JSF uygulaması yönetilmiş ortamda çalışır.
  - Modeldeki veri içeren bileşenler görünümdeki görsel bileşenlerin davranışını etkileyen durumu saklar.
  - Bu nedenle bu bileşenlerin yönetilmesi uygulama geliştiren kişiye büyük kolaylık sağlar
    - Dört değişik kapsam vardır. İstek /Oturum / Uygulama / Adsız.
    - Nesneler bir soyut fabrika tarafından yaratılır ve ayar dosyaları aracılığı ile belirtilen ayarlara sahip olurlar. Bu fabrika XML dosyalarından aldığı parametreleri kullanır.

Java Server Faces

- Bir JSF uygulamasının JSP önyüzü ile çalışması durumunda HTTP istek/yanıt ikilileri oluşmak zorundadır.
  - Bu durumda JSF'in JSP önyüz sınıfları devreye girecektir. Bunların çalışması Struts'ın çalışmasına çok benzer.
  - Bir **Ön Denetleyici** (Front Controller) görevine sahip olan **FacesServlet** sınıfı
    - İstekleri değerlendirir.
    - Gerekli olayları oluşturur ve olay işleyicilere yönlendirir.
    - Uygulama akışını seçilen bir görünümün bileşen ağacına yönlendirir.

Java Server Faces



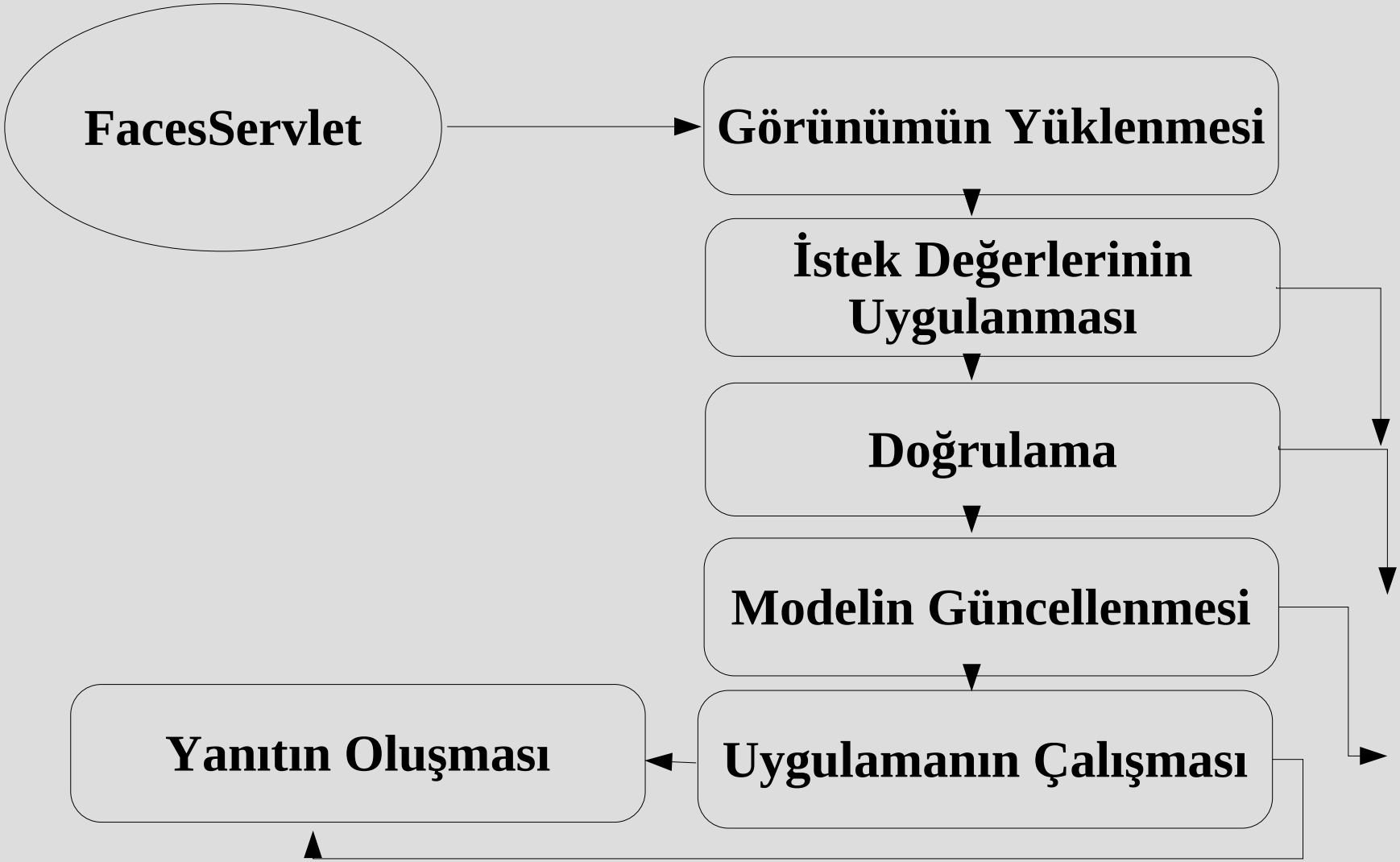
Java Server Faces

- JSF görsel bileşenleri **Bileşke** (Composite) tasarım biçiminin bir örneği olan bir bileşen ağacı içinde yer alırlar. Ağacın kökünde UIViewRoot adını verdiğimiz bir sınıfın nesnesi bulunur.
  - JSF uygulaması aslında bir yada birden fazla görünüm (bileşen ağacı) ve bu görünümler üzerinde tanımlanan olayların işlenmesi için gereken mekanizmadan oluşmaktadır.
  - Her bir görsel bileşenin seçilen önyüz teknolojisi (örneğin JSP) için bir sunum dengi bulunur.

Java Server Faces

- Bir görsel bileşen ağacı üzerinde oluşan olayların çoğunluğu kullanıcı etkileşiminden kaynaklanır.
  - Bu olaylar ilgili olay işleyicilere gönderilip işlendikçe gösterilecek olan görünümde değişiklikler olabilir.
  - Bazı olaylar işlenirken oluşan hatalar yada doğrulamaların sonucu nedeni ile görsel bileşenler üzerinde değer değişiklikleri yapılmayabilir.

Java Server Faces



Java Server Faces

- Şu anda JSF uygulaması örnekleri daha çok JSP önyüz teknolojisi üzerine kuruludur. Ancak bunun ileride çok çeşitli yapılara yönelmesi kaçınılmazdır.
  - JSF kullanarak geliştirilebilecek olan önyüzün bir beceri üst sınırı yoktur.
- JSP önyüzü üzerine kurulu JSF uygulamaları Struts, Spring, Tiles, vb bir çok başka çatı ile entegre edilebilir.

Sun Referans Uyarlaması

- 2001 yılından bu yana geliştirilen JSF mimarisi 2004 yılı içinde 1.1 sürümüne ulaşn ve 2005 içinde çıkacak 1.2 sürümü ile tam olgunluğa ulaşacak bir referans uyarlamaya sahiptir.
  - Eksiksiz JSP desteği
  - Eksiksiz WAP desteği
  - Kısmi XUL desteği

MyFaces Uyarlaması

- Özgür bir JSF uyarlaması olarak planlanan ve 2004 sonunda bir Apache projesi haline gelen MyFaces şu anda 1.0.9RC1 (16 Mart 2005) sürümündedir.
  - Eksiksiz JSP desteği
  - Eksiksiz WAP desteği
  - Çok çeşitli ek bileşenler
- http://myfaces.apache.org/

MyFaces Uyarlaması

- MyFaces ek bileşenleri
  - Menü
  - Ağaç Görünümü
  - Kaydırma çubuğu
  - Sıralama Başlığı
  - Takvim
  - Dosya Yükleme
- MyFaces ek doğrulama sınıfları
  - Eposta
  - Kredi kartı
  - Düzgün deyim

MyFaces Uyarlaması

- Herhangi bir uyarlama ile geliştirilen bir uygulamayı yeniden derlemenize gerek olmadan MyFaces'a çevirebilirsiniz.
  - Uygulamanın /lib dizinlerinde yer alan JAR dosyaları yerine MyFaces'in JAR dosyalarını koyun.
  - Eğer eldeki uyarlama ile gelen ve asgari JSF beklentileri dışında gelen bir bileşen kullanıyorsanız o zaman o bileşenlerin JAR dosyası ayrı bir dosya olmalıdır. Bu dosyayı silmeyin.

Teşekkürler