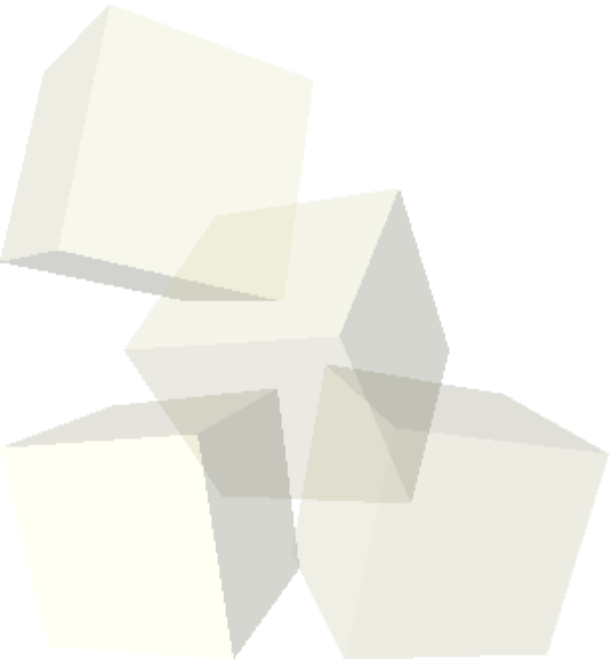




XML ve Web Servisleri





- Bora Güngören - Portakal Teknoloji
 - bora@portakalteknoloji.com
- Sunum 4 Ağustos 2004 Çarşamba günü Ankara Üniversitesi'nde LKD adına verilmiştir. Sunum için hazırlanan slaytlar GPL lisansı ile dağıtılmaktadır.
- GPL lisansının detayları için LKD ile <http://www.linux.org.tr/> adresinden bağlantıya geçebilirsiniz.
- Seminerin kendi bölgenizde tekrarı yada LKD'den benzeri başka seminer talepleri için seminer@linux.org.tr adresine bir eposta yollayabilirsiniz.



- Veri ve Meta-Veri
- XML Nedir?
- XML'i Nerede kullanabiliriz?
- Nelere İhtiyacımız var?
- XML Kuralları
- DTD ve XML-Schema
- Apache Xerces
- DOM ve SAX
- Web Servisi Kavramı
- XML-RPC
- PHP ile bir XML-RPC Örneği
- Web Servisi Mimarileri



- “Veri” (data) herhangi bir konuda topladığımız değerlerdir.
 - ♦ Verinin belli bir biçimde organize olması durumunda bu verileri bilgisayarda işleme şansımız olur.
 - ♦ Veri yapıları, dosya yapıları, veri tabanlarındaki tablolar, vb. Yapılar hep bu organizasyonu sağlamak için yaratılmıştır.
- Ancak verinin “anlamı” söz konusu olduğu zaman bilgisayarlar yetersiz kalır.
 - ♦ 10-10-2004 : 10 Ekim 2004
 - ♦ 04-08-2004 : 4 Ağustos 2004? 8 Nisan 2004?



- “Meta-veri” (meta-data) “veri hakkında veri” demektir.
 - ♦ Meta veri sağlandığı zaman, bir bilgisayar programı bile verinin ne anlama geldiğini anlama şansına sahiptir.
 - ♦ [Gün:04, Ay:08, Yıl:2004]
 - ♦ “Gün”, “Ay” ve “Yıl” meta-veri olacaktır.
- Meta verinin açık seçik tanımlı olmadığı yerde programcı kendi kurallarını koymaya başlar.
 - ♦ [Gün:04, Ay:08, Yıl:2004]
 - ♦ İlk “iki nokta üst üste”den sonrası gün, ikinciden sonrası ay, üçüncüden sonrası yıldır.



- Aynı veriyi kullanmak isteyen başka bir programcı da başka kurallar koyabilir.
 - ♦ İlk “virgül”den önce gün, ikinciden önce ay, “]” işaretinden önce de yıl verisi gelir.
- İki programcı aynı veriyi ve aynı kuralları kullanmakta anlaşsalar bile yine de sorunlar çıkabilir.
 - ♦ Bitlerin yorumlanması farklı bilgisayar mimarilerinde farklı olabilir.
- Bu durumda, ortak tanımlanmış bir “meta-veri” tekniği gereklidir.



- XML, eXtensible Markup Language sözcüğünün kısaltmasıdır.
- XML bir işaretleme dilidir. Yani verileri işaretleriz.
 - ♦ `<işaret>veri</işaret>`
 - ♦ İşaretlenen her veriye “eleman” (element) adını veriyoruz.
- Bir XML belgesi belli kurallara göre organize edilmiş elemanlardan oluşacaktır.
- XML'in güzel yanı bu kuralların genişletilmesidir.
 - ♦ Yeni işaretler tanımlayabiliriz.
 - ♦ Her elemanın içerdiği verinin doğrulanması için kurallar koyabiliriz.
 - ♦ Bu tanımları ve kuralları ortak kullanmak için mekanizmalar tanımlayabiliriz.



XML'i Nerede Kullanabiliriz?

- XML sayesinde veri aktarımı yapılan her yerde, aktarılan veri, “meta-verisi ile birlikte” aktarılabilir.
 - Meta-veri hakkındaki kurallar, yalnızca bir kez, ayrıca aktarılır.
- Bankacılık işlemleri başta olmak üzere bütün finansal işlemlerde XML standart veri aktarım tekniğidir.
- Gelişmiş ERP yazılımları aralarındaki haberleşmeyi XML kullanarak yaparlar.
- Bir çok sistem yazılımının ayar dosyaları XML dosyasıdır.
- Java Serileştirme (serialization) mekanizması nesneleri XML kullanarak kaydeder.



Nelere İhtiyacımız Var?

- XML Belgeleri birer metin belgesidir.
- Bu belgenin okunması ve bellekteki bir modele eşlenmesi gerekir.
 - ♦ Belgeyi bu modele eşlemek için bir XML Ayırıştırıcısı (XML Parser) gereklidir.
 - Apache Xerces.
 - ♦ Belleğe eşlenmiş modeli işlemek için bir programlama aracı gereklidir.
 - Dile ait DOM ve SAX API'leri. C/C++, Java, PHP, Python, ASP, VB, C#, vs. için var.
- Okunan belgelerin doğrulanması için bir mekanizma gereklidir.
 - ♦ DTD ve XML-Schema



- XML belgeleri, doğrulama dışında, katı biçimsel kurallara sahiptir.
- Bir XML belgesinin “iyi biçimlenmiş” (well formed) olması zorunludur.
 - `<bir>veri</bir>` doğru
 - `<bir özellik="değer" />` doğru
 - `<bir>veri` yanlış
 - `<bir><iki>veri</iki></bir>` doğru
 - `<bir><iki>veri</bir></iki>` yanlış

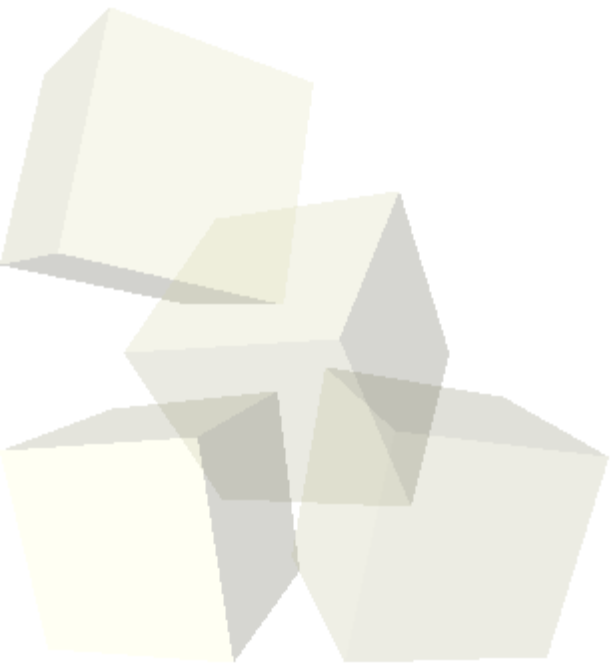




- DTD (Document Type Descriptor) ve XML-Schema, XML belgelerini doğrulamak için kullanılan iki alternatif yoldur.
- Her iki teknikte de şunları doğrularız
 - ♦ Elemanların özelliklerini (attributes)
 - ♦ Elemanların aldıkları verileri ve bu verilerin türlerini
 - ♦ İstenirse elemanların aldıkları veriler üzerinde sayı aralığı, düzgün deyim işleme gibi karmaşık doğrulamalar yapılabilir.
 - ♦ Ayrıca elemanların aldıkları değerleri bir sıralama (enumeration) olarak tanımlama şansı da vardır.
 - Bir şifrenin 19xx biçiminde olmamasını doğrulatabiliriz.
 - Bir banka hesap numarasının biçimini ve şube no alanının değerini XML'de doğrulatabiliriz.



- DTD ve XML-Schema sayesinde belgelerin değerleri daha okunurken ve standart doğrulama teknikleri kullanılmadan (örneğin VT bağlantısı) biçimsel doğrulama tamamlanır.
- Biçimi uygun olmayan bir elemanın verilerinin doğrulanmaya çalışılmasına zaten gerek olmaz.





■ DTD Örneği

```
<!DOCTYPE SEMINER [  
  
  <!ELEMENT SUNUM (TANISMA, SLAYT+, SORU+)>  
  <!ELEMENT TANISMA (#PCDATA)>  
  <!ELEMENT SLAYT (#PCDATA)>  
  <!ELEMENT SORU (#PCDATA)>  
  
  <!ATTLIST SUNUM SUNAN CDATA #REQUIRED>  
  <!ATTLIST SUNUM TARİH CDATA #IMPLIED>  
  
  <!ENTITY SUNUM "XML ve Web Servisleri">  
  
>]
```



■ XML-Schema Örneği

```
<!-- XML dosyası -->
```

```
<?xml version="1.0" encoding="ISO-8859-9"?>
```

```
<gonderi id="889923"
```

```
xmlns:xsi="http://www.linux.org.tr/2004/XMLSchema-Ornegi"
```

```
xsi:noNamespaceSchemaLocation="gonderi.xsd">
```





■ XML-Schema Örneği

<gonderen>Bora Güngören</gonderen>

<alici>

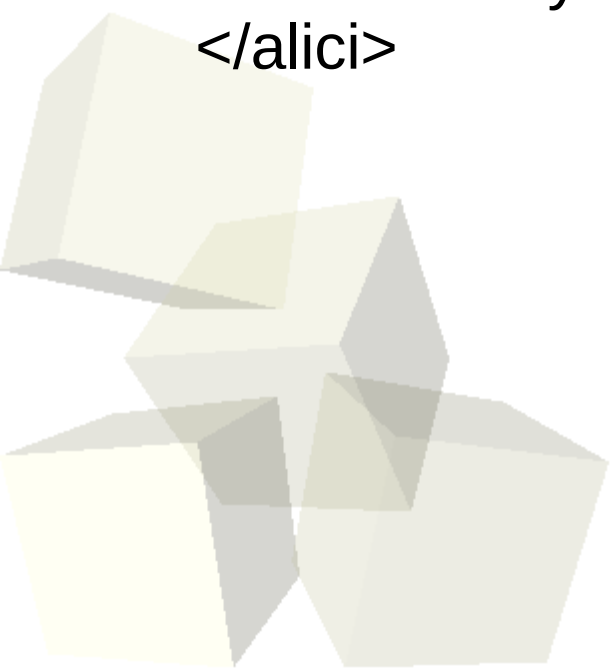
<ad>Cem Vedat Işık</ad>

<adres>Portakal Teknoloji</adres>

<sehir>Ankara</sehir>

<ulke>Türkiye</ulke>

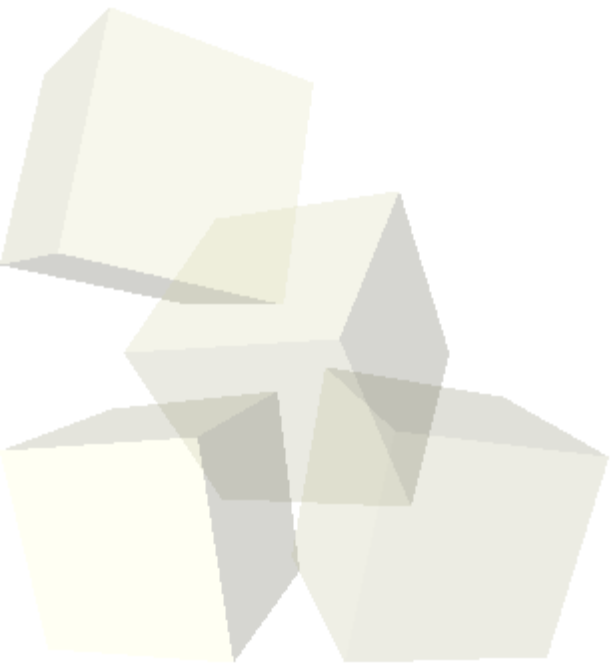
</alici>





■ XML-Schema Örneği

```
<kalem>  
  <adi>USB Kahve Isıtıcısı</adi>  
  <not>Hediye Paketi</not>  
  <adet>1</adet>  
  <eder>10000000</eder>  
</kalem>
```

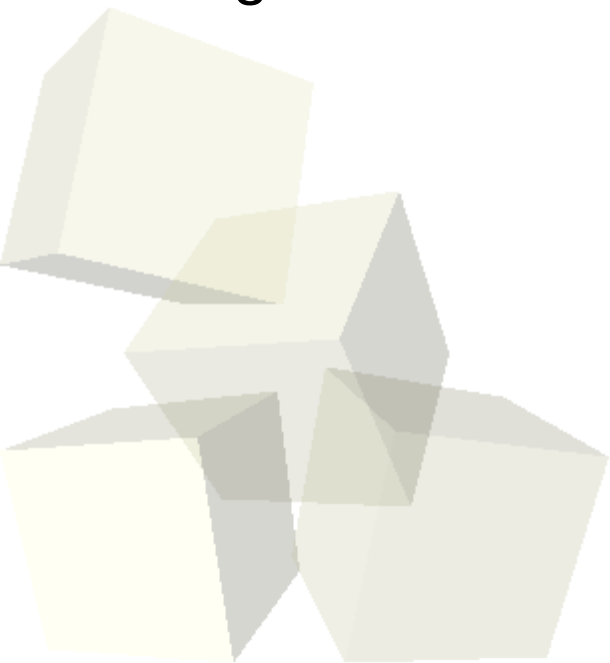




■ XML-Schema Örneği

```
<kalem>  
  <adi>CD</adi>  
  <adet>50</adet>  
  <eder>20000000</eder>  
</kalem>
```

```
</gonderi>
```





■ XML-Schema Örneği

```
<!-- XSD Dosyası -->  
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  
  <xs:element name="gonderi">  
    <xs:complexType>  
      <xs:sequence>  
  
        <xs:element name="gonderen" type="xs:string"/>  

```





■ XML-Schema Örneği

```
<xs:element name="alici">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="ad" type="xs:string"/>  
      <xs:element name="adres" type="xs:string"/>  
      <xs:element name="sehir" type="xs:string"/>  
      <xs:element name="ulke" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



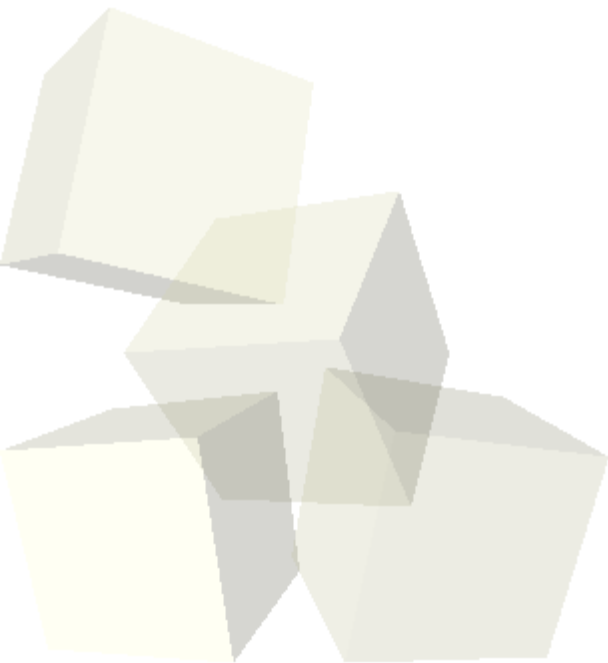
■ XML-Schema Örneği

```
<xs:element name="kalem" maxOccurs="unbounded">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="adi" type="xs:string"/>  
      <xs:element name="not" type="xs:string" minOccurs="0"/>  
      <xs:element name="adet" type="xs:positiveInteger"/>  
      <xs:element name="eder" type="xs:decimal"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



■ XML-Schema Örneği

```
</xs:sequence>  
<xs:attribute name="id" type="xs:string" use="required"/>  
</xs:complexType>  
</xs:element>  
  
</xs:schema>
```





- XML'in artan önemi Apache grubunun kendi uygulamalarında XML desteğini sağlamak için kendi XML projelerini başlatmalarına neden oldu.
- Elbette ki bütün projeler açık kaynak kodludur.
- Xerces bu projelerin en çok duyulanı ve en çok kullanılanıdır.
 - Xerces bir XML ayrıştırıcısıdır (XML Parser).
 - Dünyada en yaygın kullanılan ayrıştırıcı olduğu tahmin edilmektedir.
 - DOM Seviye 2 (3 test ediliyor) ve SAX 2 desteği verir.
 - C++ ve Java için API'leri vardır. Bunlara Perl ve COM ile de ulaşılabilir.



- Apache Xerces'in Java kitaplıkları Java 2 Sürüm 1.4 ile birlikte gelmektedir.
- Java'da XML kullandığınızda aslında Apache Xerces'i otomatik kullanırsınız.
- Apache Vakfı'nın bazı XML projeleri
 - Xalan: Bir XSL İşleyicisi
 - AxKit: Apache için XML eklentisi yazmaya yarayan bir araç kutusu.
 - FOP: XSL tabanlı bir yazdırma biçimleyicisi.
 - Forrest: Belgeleme motoru.
 - Batik: SVG yaratmak için bir kitaplık.
 - XML Security: XML belgeleri için sayısal imza desteği.
 - Xindice: XML veritabanı.



- DOM (Document Object Model) ve SAX (Simple API for XML) iki değişik belge modeli sunar.
- DOM belgeyi bir belge ağacı olarak görür.
- Ağacın her seviyesinde çeşitli DOM nesneleri olur.
 - Örneğin bir HTML belgesi bir DOM nesnesi olarak alt nesnelerden oluşur. `<body>` elemanının altında bir `<table>` elemanı, onun altında çok sayıda `<tr>` elemanı, her `<tr>` elemanının altında da bir yada daha fazla `<td>` elemanı olabilir.
- DOM ağaç üzerinde alt ağaçları blok halde işlemeye izin verir.
 - Yani tablonun satırlarının yerini değiştirdiğimizde `<td>` elemanları da `<tr>` yi takip eder.



- DOM tamamı doğrulanmış bir belgenin yapısı üzerinde işlemler yapmak yada karmaşık yapıları elemanları kullanmak için ideal olabilir.
- SAX ise daha farklı bir bakış açısı sunar. Belgeyi bir insanın okuduğu gibi sıra ile okur.
 - Örneğin ardışık gelen siparişlerin işlenmesi için SAX daha uygun olabilir.
- SAX arada hatalar geldiği zaman hatalı elemanın atlanacağı ve sıradakine geçileceği uygulamalarda ideal olabilir.



- Bir uygulamanın dağıtık yazılması yeni bir kavram değildir.
- Bir işlemin bir kısmının uzakta yapılıp sonucunun alınması için nedenler olabilir:
 - ♦ İşlem yükünün fazlalığı
 - ♦ İş kurallarının sürekli değişmesi
 - ♦ Emir – komuta zinciri türü yapılar.
- 1970'lerden bu yana Uzaktan Yordam Çağırma (Remote Procedure Call - RPC) teknikleri çeşitli alanlarda kullanılmaktadır.
 - ♦ Askeri Uygulamalar
 - ♦ ERP Uygulamaları
 - ♦ Finans Uygulamaları



- RPC teknikleri genellikle soketler üzerinden uygulamaya özel paketlerin aktarılması ile uygulanır.
- Bu durumda iki yabancı uygulamanın haberleşmesi çok büyük bir sorun olmaktadır.
- Bu sorun haberleşmesi gereken kişilerin kullandığı sistemlerin sayısı arttıkça büyür.
 - ♦ TCMB ile bankaların (bir kaç on tane) haberleşmesi.
 - ♦ Gümrükler ile dış ticaret şirketlerinin (bir kaç yüz tane) haberleşmesi.
 - ♦ Bir otomotiv şirketinin tedarikçileri (bir kaç bin tane) ile haberleşmesi



- Bir web servisi (web service) temel olarak klasik RPC tekniğini uygular. Ancak onu ayıran çok önemli bazı kavramlar vardır.
 - Web servisleri HTTP üzerinden çalışır. Özel hatlar ve sözleşmeler kullanmazlar.
 - Bir web servisini aramak, sunduğu olanakları tanımak için yollar vardır.
 - Web servislerinde karşınızdaki kişiyi ve gelen sonuçları doğrulamak için yollar bulunmaktadır. Ortadaki adam (man in the middle) saldırılarına karşı güvenlik önlemi alınabilir.
 - Aktarım XML tabanlı yapılır. Bu nedenle değişik dillerde yazılmış uygulamalar da web servisleri aracılığı ile haberleşebilir.



- XML-RPC, bir web servisi yazmanın en ilkel yoludur.
- Bir istemci (client) ile bir süreç (process) arasında eş zamanlı iletişimden oluşur.
 - Her XML-RPC isteği daima bir XML-RPC yanıtı ile karşılanır.
 - İstemci, bir başka uygulama tarafından kullanılır.
 - Süreç de bir dinleyici (listener) tarafından kullanılır.
 - Ayrıca bir web sunucusunun kurulmasına gerek vardır.



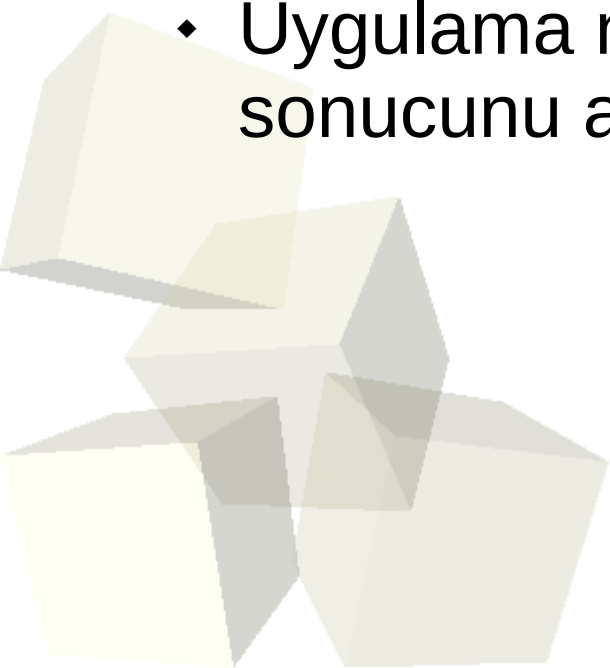


■ Kısaca özetlersek.

- RPC hizmetlerinden yararlanması gereken uygulama yazılımı XML-RPC istemcisinden bir yöntem (işlev) çağırır. Bunu yöntemin adı, parametrelerin değerleri ve hedeflediği sunucuyu (bir URL ile) belirterek yapar.
- İstemci, yöntemi çağırmak için bir HTTP POST isteği oluşturur. Bu POST isteğini sunucuya yönlendirir.
- POST isteğini alan sunucu basit bir web sunucusudur. Ancak sunucu URL'inde bir XML-RPC dinleyicisi bulunmaktadır. Web sunucusu dinleyiciye XML belgesi olarak isteği aktarır.
- Dinleyici XML belgesini ayrıştırır. Buradan yöntemin adını, parametrelerini alır. Süreci kullanarak yöntemi çağırır.



- Kısaca özetlersek (devam ediyor).
 - Süreç, gereken işlemleri yapar ve bir sonuç elde eder. Sonuç XML-RPC'ye uygun bir XML belgesine çevrilir.
 - Bu XML belgesi web sunucusu tarafından kendisine verilen HTTP POST isteğine bir yanıt olarak yollanır.
 - XML-RPC istemcisi, XML belgesini HTTP yanıtı olarak alır. Ayırıştırır ve yöntemin sonucunu kendisini kullanan uygulamanın istediği biçime çevirir.
 - Uygulama normal bir yöntem (işlev) çağrısının sonucunu almış gibi çalışmasına devam eder.





- İletişim ağ üzerinden olduğu için, uygulama yazarken dikkat edilmesi gereken yerler vardır.
 - XML-RPC istemcisini kullanarak sonuç bekleyen uygulamaların, bu sonucun gelmesinin zaman alacağını düşünerek yazılması gerekir.
- Çok kanallı uygulama geliştirme pratikleri önem kazanır.
 - MVC mimarisinde XML-RPC istemcisi bir model, uygulama yazılımı ise bir denetleyici olacaktır.
 - Denetleyicinin uyuması yada bir mekanizma ile bloklanması uygun bir pratiktir.
 - Görünümün denetleyiciden soyutlanmış olması gereklidir.



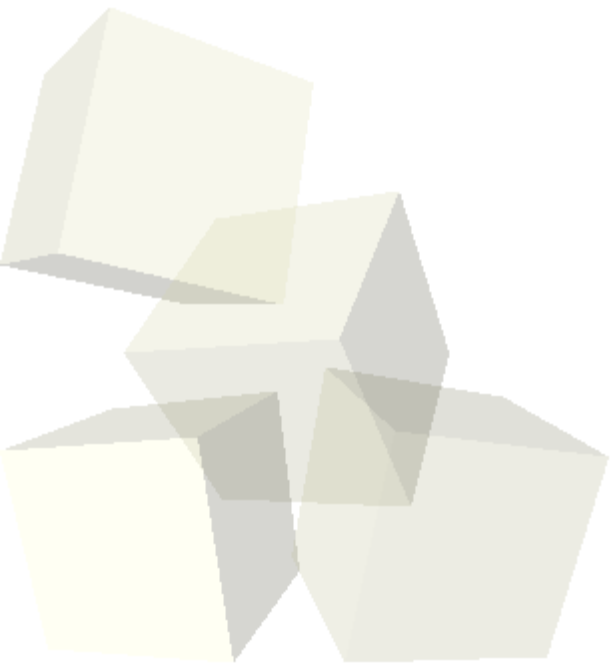
- XML-RPC'nin eş zamansız yazılması mümkündür.
 - Ancak bu durumda her iki tarafta da bir istemci ve bir sunucu bulunmalıdır.
 - Bu da her iki uygulamayı da karmaşık hale getirir. İstemci uygulamalarının çok basit bir doğaya sahip olması istemci-sunucu mimarisinin gereğidir.
- XML-RPC için varsayılan kapı 80 yada 8080 olacaktır. Bu kapılar için gerekli güvenlik ayarlarının yapılması servisin taciz edilmemesi için gereklidir.
 - Buna ek olarak XML biçiminde olmayan GET ve POST istekleri için basit bir işleyici yazılarak sunucu URL'ini korumak da mümkündür.



- XML-RPC web tabanlı çalıştığı için, kodlar genellikle PHP, Java Servlet, ASP gibi dillerde yazılır.
- Ancak kullanılan veri tipleri XML-RPC'nin izin verdiği veri tipleri olacaktır.
 - `<value><int>25</int></value>`
 - `<value><double>-2.9</double></value>`
 - `<value><boolean>1</boolean></value>`
 - `<value><string>Büyüktür işareti: ></string></value>`
 - Standart sadece ASCII metinlerini destekler. Ancak uygun bir XML ayrıştırıcınız varsa, diğer ISO ve UNICODE kodlamaları da desteklenir.
 - `<value><dateTime.iso8601>20040804T18:30:00</dateTime.iso8601></value>`



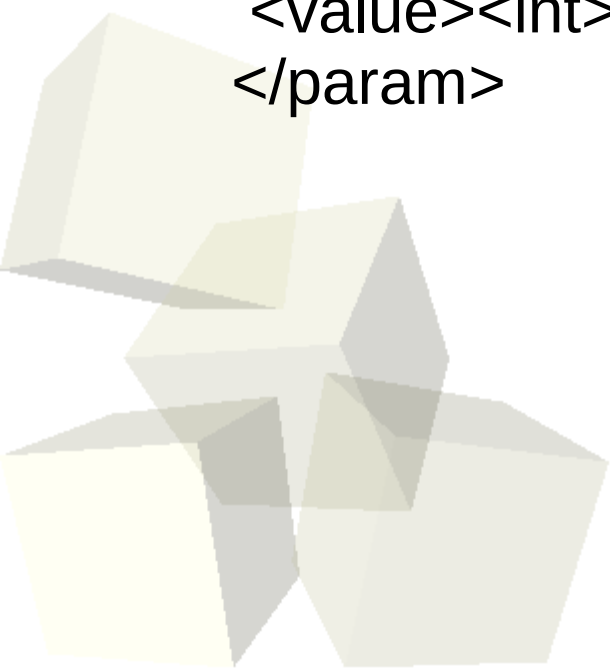
- XML-RPC standart veri tiplerine ek olarak üç değişik veriyi de aktarmaya izin verir.
 - ♦ İkili kodlama olarak yollanmış metinler
 - ♦ Diziler
 - ♦ Yapılar





- Bir XML-RPC isteğinin XML'e çevrilmiş hali

```
<methodCall>  
  <methodName>ekle</methodName>  
  <params>  
    <param>  
      <value><int>12345</int></value>  
    </param>  
    <param>  
      <value><int>67890</int></value>  
    </param>  
  </params>  
</methodCall>
```





```
<param>
  <value><struct>
    <member>
      <name>kullanici</name>
      <value><string>Bora</string></value>
    </member>
    <member>
      <name>dogum</name>
      <value><dateTime.iso8601>
        19790819T08:00:00
      </dateTime.iso8601></value>
    </member>
  </struct></value>
</param>
</params>
</methodCall>
```



■ Peki istemcinin HTTP isteği neye benzer?

POST /servis.php HTTP/1.0

User-Agent: PHP XMLRPC 1.0 (xml@sunucum.com)

Host: xml.sunucum.com

Content-Type: text/xml

Content-Length: 340

■ XML-RPC sunucusu başka (düz HTTP) hizmetleri ile aynı IP üzerinden verilebilir.

- Bu durumda, web sunucusunda isim tabanlı sanal sunucu (name based virtual host) ayarının yapılması gereklidir.



- Sunucu tarafında işlemler yapıldıktan sonra gelecek olan yanıt da XML olarak biçimlendirilmiş olacaktır.

```
<methodResponse>  
  <params><param>  
    <value><int>1</int></value>  
  </param></params>  
</methodResponse>
```

- Bir XML-RPC yanıtında bir ve yalnız bir değer verilmesi zorunludur. XML-RPC “void” yöntem desteği vermez.
 - Kullanılmayacak bir değer dönerek bunu aşabiliriz.



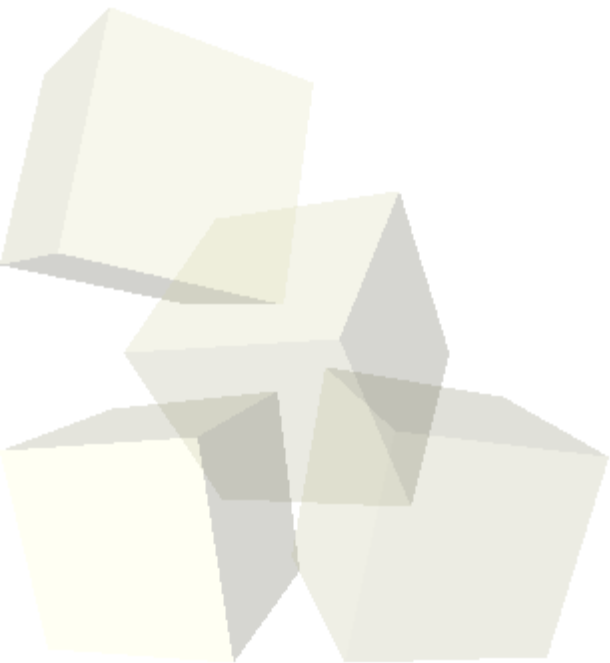
- XML-RPC hata durumları (exception) desteği verir. Bu durumda yanıt aşağıdaki gibi olur.

```
<methodResponse>  
  <fault>  
    <value> ... </value>  
  </fault>  
</methodResponse>
```

- Ancak hata durumlarına destek olup olmaması, yada var olan desteğin farklılığı nedeni ile XML-RPC'de bu konu ucu açık bırakılmıştır.



- XML-RPC yanıtlarında HTTP başlığı web sunucusu tarafından yaratılır.
 - ♦ XML-RPC yanıtının Content-Type başlık elemanının text/xml olarak ayarlanması yeterlidir.
 - ♦ Geri kalan kısımları web sunucusu halledecektir.





- Kolaylık olması açısından aşağıdaki kitaplığı kullanacağız.
 - ♦ <http://xmlrpc.usefulinc.com/php.html>
 - ♦ http://sourceforge.net/project/showfiles.php?group_id=34455
- Bu kitaplıkta PHP'de XML-RPC kullanımı için gerekli tüm detaylar, PHP sınıfları olarak tasarlanmıştır.
 - ♦ İstemci sınıfı
 - ♦ Sunucu sınıfı
 - ♦ Sık kullanılan XML elemanları için sınıflar



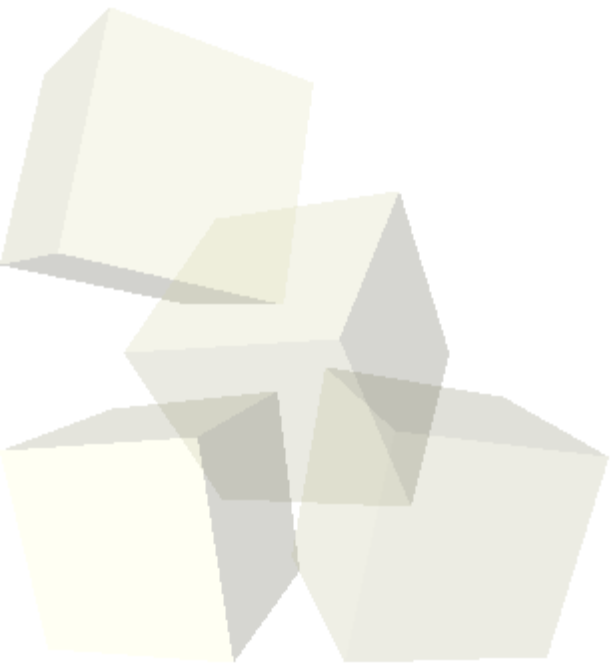
PHP ile bir XML-RPC Örneği

```
include("xmlrpc.inc")
$sorgu = "Bora";
$istemci = new xmlrpc_client("/RPC", "xml.sunucum.com", 80);
$mesaj = new xmlrpcmsg("yontem",
                        array( new xmlrpcval($sorgu, "string") ) );
$mesaj->addParam(
    new xmlrpcval("Güngören", "string")
);
$zamanasimi = 200;
$yontem='http';
$sonuc = $istemci->send($mesaj, $zamanasimi, $yontem);
// $sonuc xmlrpcresp sınıfından
if (! $sonuc->faultCode() ) {
    $deger = $sonuc->value(); //$ deger xmlrpcval sınıfından
    echo $deger->scalarval();
}
```



PHP ile bir XML-RPC Örneği

- PHP genellikle XML-RPC istemcilerini yazmakta kullanılır.
- Sunucularda ise güvenlik konusunda daha güçlü bir dil olan Java kullanılabilir.
- Örnek olarak kullandığımız kitaplıkta son derece gelişmiş XML-RPC sunucu sınıfları da bulunmaktadır.





- XML-RPC ile çizdiğimiz resmi genişletmek istersek ve tam ölçekli web servislerine göz atarsak neler göreceğiz?
 - Web servislerinin kaydedildiği ve üzerinde arama yapılan bir “servis aracısı” (service broker)
 - XML dışında SOAP, WSDL, UDDI ve ebXML standartları.
- SOAP (Simple Object Access Protocol), XML tabanlı haberleşme için hafif bir sözleşmedir.
 - SOAP mesajları XML biçimli bir zarf içerisinde saklar ve mesajlar hakkında meta-veri sağlar.
 - Web servisleri arasındaki mesajlar SOAP ile yollanır.



- WSDL (Web Services Definition Language) ağ kaynaklarını ve onlara erişimleri tanımlamakta kullanılan bir dildir.
 - Yine XML tabanlıdır.
 - WSDL olarak ifade edilen bilgi hizmet sağlayıcı tarafından otomatik yaratılır ve depolanır.
- UDDI (Universal Description, Discovery and Integration) ağ servislerinin tanımlarını XML biçiminde saklamak ve yayınlamak isteyen mekanizmalar için ortak arabirimler tanımlar.
 - UDDI servis aracısı tarafından yoğun olarak kullanılır. İstemciler servis aracısının UDDI hizmetini SOAP mesajları ile kullanırlar.



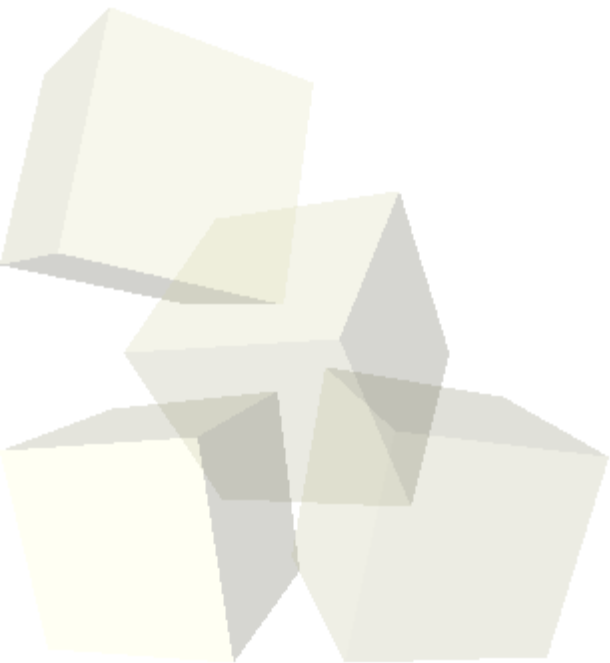
- ebXML(eBusiness XML) Birleşmiş Milletler'in girişimi ile ortaya çıkartılmış standart XML elemanları tanımları ve bunları doğrulamak için geliştirilen DTD ve Schema'lardır.
 - Ancak ebXML ayrıca iş süreci tanımlaması, iş olaylarının tanımlanması gibi başka çalışmaları da içerir.
 - Uzun dönemde her türde iş akışı uygulamasının ebXML uyumlu olması beklenmektedir.
 - ebXML, diğer standartlardan farklıdır. UDDI, WSDL gerektirmez.



- Bir web servisi, daha farklı bir yazılım mimarisi sunar.
 - Veri tabanlı (data oriented) mimariler, verilerdeki değişiklikler tarafından güdülenir. Yani bir eyleme geçilmesi için verilerde bir değişiklik olması gerekir.
 - Olay tabanlı (event oriented, event driven) mimariler, bir eyleme geçilmesi için bir olayın olmasını beklerler.
 - Servis tabanlı (service oriented) bir mimaride ise uygulama bileşenleri ağ üzerinde dağılmış ve her biri de bir “servis” olarak soyutlanmıştır.
 - Var olan uygulamalar (legacy applications) bir sarıcı (wrapper) ile sarılarak yeni yazılan uygulamalara kolayca entegre edilebilir.



- Web servislerinin çalışması aşağıdaki gibi olur.
 - Web servisini sağlayan sunucuda servis için bir SOAP arabirimi ve WSDL tabanlı servis tanımlayıcısı oluşturulur.
 - Servisler servis aracısına kaydedilir.
 - Servis aracısı servis tanımlarını bir şablonda tutar. Her servis için bir de bağlantı URL'i bulunur.





- Web servislerinin çalışması aşağıdaki gibi olur (devam ediyor.)
 - Web servisi istemcisi, servis aracısına SOAP mesajları ile UDDI sorguları geçer ve servis arama isteğini belirtir.
 - İstemci servisin yerini (URL) ve diğer bağlantı bilgilerini aracıdan alır.
 - İstemci aracıdan aldığı bilgiler ile sunucuya bağlanır ve sunucudaki WSDL tanımlayıcılarını alır. Bu tanımlayıcıları kullanarak kendisinde bir vekil oluşturur.
 - İstemci sunucu haberleşmesi bu vekil üzerinden sağlanır. Sadece SOAP haberleşmesi değil RPC haberleşmesi de kullanılabilir.
 - Bu nedenle XML-RPC de bir web servisidir.

