

## Predavanje 4

INTERNET TEHNOLOGIJE 2009/2010

# Kreiranje Web aplikacija

## XML, Web servisi

Razmotrimo uticaj Interneta i World Wide Weba na razvoj preduzetničkih, za Web osposobljenih (*Web-enabled*) IT aplikacija. Razmotrimo kako se uz pomoć **XML-a i web servisa** mogu razmenjivati podaci i međusobno povezivati aplikacije iz različitih preduzeća.

Projektanti sistema sve češće nastoje da u preduzetničke sisteme inkorporiraju **karakteristike Interneta**, u stručnim krugovima se obično označava terminom **Web-enablement** (u slobodnom prevodu, "osposobljavanje za Web"). Ovakvi, za Web osposobljeni sistemi pojavljuju se iz tri razloga:

- Široka upotreba web čitača (*browsera*),
- Očuvanje vrednosti nasleđenih sistema,
- Kreiranje među-preduzetničkih aplikacija.

## Širina upotrebe web čitača

Web čitači (*browseri*) su izazvali najveću pažnju individualnih korisnika i IT profesionalaca. Kao rezultat toga, danas postoji ogromna baza personalnih kompjutera na kojima je instaliran softver nekog web čitača. U najširoj upotrebi su *Microsoftov Internet Explorer* i *Netscape Navigator*. Zbog čega su web čitači postali tako značajni? Najvažniji razlozi za to su sledeći:

- **Novi kompjuteri sa instaliranim web čitačima** Većina novih kompjutera se kupcima isporučuje sa unapred instaliranim, jednim ili više web čitača. Nema, dakle, potrebe da se korisnik brine oko nabavke i instaliranja odgovarajućeg softvera.
- **Ogromna baza korisnika web čitača** - Zbog široke dostupnosti web čitača, ogroman je broj pojedinaca i poslovnih preduzeća koja ih koriste. Većina od preko 500 miliona korisnika PC računara (2004) danas aktivno upotrebljava svoje web čitače.
- **Pojava Web čitača na drugim IT uređajima** Velika popularnost web čitača na PC računarima dovela je do njihove upotrebe i na ručnim IT uređajima. Lični digitalni pomoćnici (PDA uređaja) i mobilni telefoni opremljeni su web čitačima.
- **Jednostavno korišćenje** - Pojedincima nije potrebna gotovo nikakva obuka za korišćenje web čitača. Nakon svega par minuta objašnjavanja korisnik se lako adaptira na njihove mogućnosti i karakteristike, tako da njihova upotreba ubrzo postaje potpuno rutinska i intuitivna.
- **Fleksibilnost** Izgled i mogućnosti web čitača mogu se prilagođavati ukusu i potrebama najrazličitijih tipova korisnika. Osim toga, u web čitače se mogu ubacivati mnoge dodatne karakteristike preko takozvanih **dodataka (plug-ins)**, softverskih programa koji na neki specifičan način proširuju mogućnosti datog web čitača (na primer, dodavanje mogućnosti za reprodukciju muzičkih samplova, pregled video isečaka ili animiranih demonstracija).
- **Univerzalna front-end aplikacija** Funkcionalne prednosti web čitača, potpomognute njihovom najširojom prihvaćenošću od strane korisnika, za posledicu imaju sve veće interesovanje projekatara sistema za njihovu upotrebu kao univerzalnog "**prednjeg kraja**", odnosno interfejsa. Web čitači se koriste za uspostavljanje interakcije sa preduzetničkim sistemima.

## Nasledeni sistemi

U poslednje vreme, sve je veće interesovanje za **osposobljavanje nasledenih sistema** za Web, pri čemu se web čitač koristi kao novi interfejs. Termin nasledeni sistem (*legacy system*) odnosi se na veliki broj mainframe, midrange, klijent/server ili PC aplikacija, koje se koriste za upravljanje poslovnim funkcijama kao što su: finansije, računovodstvo, proizvodnja i ljudski resursi, ali koje su **projektovane pre nego što je došlo do eksplozije Interneta** i njegove upotrebe u preduzetničkim aplikacijama.

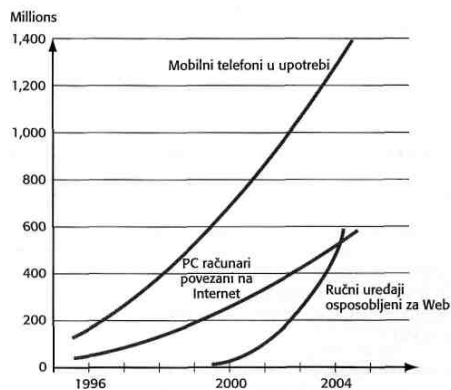
**Situacija:** U većini organizacija, nasledene aplikacije su razvijane tokom dugog niza godina uz značajne troškove. Na žalost, one često otežavaju funkcionisanje tzv. među-preduzetničkih aplikacija, zbog toga što obično predstavljaju zamršenu mešavinu mainframe, midrange i PC aplikacija, centralizovanih i distribuiranih, i bez jedinstvene kontrole nad njima. Osim toga, nasledeni sistemi se često sastoje od čvrsto ukorenjenih platformi za upravljanje podacima, koje u sebi sadrže vlasnički dizajnirane softverske programe. Oni, takođe, mogu biti u geografskom smislu rasepkani, sadržati u sebi međusobno nekompatibilne baze podataka, pa čak i potpuno neprirodne spojeve preduzetničkih i poslovnih jedinica. Nadalje, korisnički interfejsi su često neprikladni, a samo mali broj njih sadrži grafičke karakteristike koje su korisnici navikli da vide na svojim PC računarima.

### dodaci (plug-ins)

Softverski programi koji proširuju mogućnosti web čitača, obogaćujući ga dodatnim karakteristikama.

### nasledeni sistem

Ovaj termin se odnosi na veliki broj mainframe, midrange, klijent/server ili PC aplikacija, koje se koriste za upravljanje poslovnim funkcijama.



Slika 12.1 Rast broja alternativnih IT uređaja

Uprkos svim ovim nedostacima, poslovna preduzeća se i dalje u velikoj meri oslanjaju na nasledene sisteme prilikom obavljanja svojih svakodnevnih aktivnosti. Upravo zbog toga, mnoga preduzeća smatraju izuzetno važnim očuvanje vitalnosti svojih nasledenih sistema, uz njihov dalji razvoj i modernizaciju. Osposobljavanje za Web je uobičajena **taktika** za povećanje korisnosti nasledenih sistema i njihovo integrisanje sa drugim aplikacijama.

## Međusobno povezivanje preduzeća

Danas je postalo uobičajeno da preduzeća svoje poslovne sistema povezuju sa odgovarajućim IT aplikacijama u drugim preduzećima. Evo nekoliko praktičnih **primera**:

- Kompanija **Dell Computer Co.** neprekidno teži povećanju efikasnosti svojih poslovnih operacija, koju zatim može pretvoriti u ključnu prednost nad konkurencijom, u formi snižavanja cene svojih proizvoda. Kada u kompaniju Dell stigne neka narudžbina za PC (kompjuter i monitor), bilo putem Weba ili preko telefona, detalji u vezi sa tom narudžbinom se elektronskim putem šalju ka različitim dobavljačima (na primer, ka dobavljaču monitora). Pritom je proizvođač monitora dužan da Dellovom kupcu monitor isporuči u isto vreme kada mu kompanija Dell isporuči kućište kompjutera i prateću opremu pa zato suštinski zavise jedan od drugog.

- U vodećoj osiguravajućoj kompaniji ukazala se potreba za **ubrzavanjem** procesa **kvotacije** novih polisa osiguranja. Međutim, kompanija je posedovala samo deo neophodnih podataka. Svi ostali detalji bili su u posedu nekih drugih firmi, uključujući kreditni biro, kojem je kompanija ugovorom poverila izvršavanje ovih usluga. Radi uvođenja novog procesa kvotacije, kompanija je projektovala jedan, za Web osposobljeni sistem, koji je bio **interoperabilan** sa IT aplikacijom u kreditnom birou. Danas ova nova aplikacija savršeno funkcioniše, bez obzira što se prostire preko granica više odvojenih kompanija.

**Primer:** "Cisco Systems međusobno povezuje sve svoje dobavljače u jedan ekosistem", opisano je kako je ova moćna kompanija, koja svoje poslovanje fokusira ka Internetu, ostvarila zavidan poslovni uspeh **integriranjem svojih poslovnih sistema** sa sistemima svojih ključnih proizvodnih partnera.

Ključ međusobne integracije preduzeća leži u **interoperabilnosti**, prikupljanju i zajedničkom korišćenju podataka od strane različitih aplikacija. Naizgled neprimetno i glatko integrisanje IT aplikacija omogućava efikasno međusobno povezivanje poslovnih preduzeća.

**Heterogenost:** Najveću **prepreku** za *interoperabilnost* predstavljaju **heterogene aplikacije**. S obzirom na to da su ove aplikacije projektovane od strane različitih organizacija, a često i u različitim vremenima, veoma je teško u njima pronaći bilo kakve sličnosti sa aspekta dizajna i karakteristika koje pružaju. Stoga se one nazivaju *heterogenim aplikacijama*, pri čemu se ta heterogenost može ogledati u tome što su **pisane različitim programskim jezicima**, namenjene su korišćenju na **različitim tipovima kompjutera** ili u tome što koriste **raznorodne komunikacijske mreže** i metode prenosa podataka. Da li će interoperabilnost postati realnost ili zauvek ostati samo jedan interesantan koncept, zavisi prvenstveno od sposobnosti datog preduzeća da prebrodi barijere uzrokovane postojanjem heterogenih aplikacija.

#### **Interoperabilnost**

Savršena forma razmene podataka i informacija, u svim njihovim oblicima (podaci, tekst, zvuk i slike, uključujući i animacije), između pojedinih komponenti neke aplikacije (hardvera, softvera, mreže).

#### **heterogene aplikacije**

Aplikacije koje su pisane različitim programskim jezicima namenjene su korišćenju na različitim tipovima kompjutera ili koriste raznorodne komunikacijske mreže i metode prenosa podataka.

U cilju osposobljavanja aplikacija za Web i takozvanog *s kraja na kraj* (**end-to-end**) integrisanja između preduzetničkih aplikacija, razvijene su **dve značajne strategije**: **proširenje jezika označavanja** (*extensible markup language*) i **web servisi**.

## **Extensible markup language (XML)**

Takozvani **hypertext markup language** - HTML, predstavlja osnovno sredstvo kojim se opisuje način prezentovanja informacija na World Wide Webu. HTML jezik se sastoji od takozvanih **tagova**, kojima su ovičene određene sekcije teksta. Ovaj jezik opisuje način na koji bi neki web čitač trebalo da formatira tekst i rasporedi slike na nekoj web stranici. Primera radi, uz pomoć HTML-a možete definisati željeni stil i boju slova, kao i koji će red na web stranici biti prikazan podebljano (bold) ili kurzivom (italic). Za činjenicu da danas postoji toliki broj web dizajnera, razlog treba tražiti prvenstveno u izuzetnoj jednostavnosti HTML-a za učenje i upotrebu.

Međutim, kako iskustvo sa upotrebom Interneta raste, tako individualni korisnici i preduzeća nastoje da ga upotrebe za mnogo toga drugo, a ne samo za prosto prikazivanje informacija. Neke organizacije bi, na primer, žele da Web upotrebe za kontrolu naučnih instrumenata konektovanih na Internet, za prenošenje medicinske dokumentacije pacijenata od jednog do drugog lekara ili za prijem i obradu **online** narudženica i za mnoge druge stvari. HTML jednostavno nema mogućnosti da zadovolji sve ove potrebe. Primera radi, detalji iz medicinske dokumentacije nekog pacijenta, koji su prezentovani preko HTML-a, mogu biti pregledani na ekranu kompjutera, ali se zato ne mogu lako obrađivati.

Potreba za novim načinima korišćenja Weba, uticala je da organizacija pod nazivom **World Wide Web Consortium (W3C)** kreira novi jezik za označavanje (**markup language**), pod nazivom **eXtensible Markup Language (XML)**.

XML je ništa drugo do **skup pravila za kreiranje tagova** u cilju opisivanja podataka, čime se ujedno obezbeđuje fleksibilnija i prilagodljivija upotreba web podataka. Ovaj jezik se naziva *proširivim* (*extensible*) zbog toga što projektanti mogu dizajnirati svoje sopstvene, prilagođene *markup* jezike za opisivanje podataka. (Strogo tehnički govoreći, XML je *metajezik* - to jest, jezik koji služi za opisivanje drugih jezika.) Nadalje, on je po svojoj prirodi neograničen, što praktično znači da se bilo koji od podataka u nekom za Web osposobljenom sistemu može bez problema opisati uz pomoć XMLa.

## Struktura XML-a

Kao i u HTML-u, u XML-u se takođe koriste parovi tagova za označavanje web podataka. Međutim, **XML tagovi definišu stvaran sadržaj podataka** (umesto formata njihovog predstavljanja), tako da ih sistem primaoca može pročitati i na odgovarajući način obraditi. Parovi tagova oivičavaju podatke na koje se ti tagovi odnose. **Primer**a radi, sledeći tag **HTML-a** služi za prikazivanje oivičenih reči podeljenim slovima: *first class*:

```
<b>first class</b>
```

On, međutim, ne pruža nikakav kontekst, pa prema tome ni značenje ovih reči. Neko ko pregleda podatke može ove reči interpretirati kao prvi čas novog semestra na nekom fakultetu. Uporedite to sa mnogo korisnijim **XML tagom**, poput ovog:

```
<ticket class>first class</ticket class>
```

Ako ovu liniju ubacimo u, recimo, neki sistem za rezervaciju avionskih karata, biće evidentno da se oivičeni podatak odnosi na avionsku kartu za kabinu prve klase, za razliku od karata za vožnju ekonomskom klasom, na primer. (**Zapamtite**, pomoću XML-a podaci se označavaju tako da se ukazuje šta oni predstavljaju - opisivanjem njihovog sadržaja. Nasuprot tome, HTML označava podatke ukazujući na koji bi način oni trebalo da budu prezentovani, ali ne opisujući pritom njihov sadržaj.)

**Struktura:** XML element je sačinjen od početnog taga, završnog taga i podatka smeštenog između ovih tagova. Početni i završni tag opisuju podatak koji se nalazi između njih koji se ujedno smatra vrednošću (*value*) tog elementa.

Tagovi se mogu umetati (**gnjezditi**) unutar drugih tagova. Ova karakteristika omogućava definisanje čitavog zapisa za potrebe neke aplikacije. Na slici 12.2 prikazana je jedna XML specifikacija za podatke unutar **zapisa o rezervaciji avionske karte**. Ona se sastoji od podataka koji služe za identifikaciju broja leta, datuma, polazišta, odredišta i tako dalje.

```
</xml version="1.0"?>
<reservations>
  <flight number="DL55">
    <date>"10122004"</date>
    <origin>London</origin>
    <destination>Paris</destination>
    <passenger>Michel Feifer</passenger>
    <frequent flier>3116534592</frequent flier >
    <ticket class>first class</ ticket class >
    <seat assignment>4D</ seat assignment >
  </flight number>
</reservations>
```

**Slika 12.2** Primer XML koda za sistem rezervacije avionskih karata

S obzirom na to da je XML proširiv (*extensible*), projektanti mogu dizajnirati svoje sopstvene tipove dokumenata, što znači da nisu ograničeni na upotrebu unapred definisanih HTML tagova. Štaviše, dizajn ovih dokumenata može biti precizno skrojen prema potrebama konkretnog korisnika. Tako, na primer, dokumenti namenjeni upotrebi od strane proizvođača i prodavaca u nekom maloprodajnom okruženju biće dizajnirani bitno drugačije od onih koji su namenjeni, recimo, lekarima, terapeutima i medicinskom osoblju u nekoj medicinskoj aplikaciji.

## Prednosti XML-a

XML ima šest važnih prednosti (tabela 12.1) u odnosu na HTML. Pored toga što je **zasnovan na Webu** i **ima mogućnost proširenja**, što su dve izuzetno važne karakteristike, XML omogućava nezavisnim kompjuterskim sistemima da stupaju u **međusobnu interakciju**, te da **razmenjuju, interpretiraju i obrađuju** podatke čak i ukoliko se izvršavaju pod potpuno različitim operativnim sistemima. Nadalje, ove aplikacije mogu biti napisane **različitim programskim jezicima**.

**Četvrto**, XML omogućava **integrisanje aplikacija** kreiranih od strane različitih projekatana, to jest takozvano "s kraja na kraj" (**end-to-end**) integrisanje. Poslovni partneri - kupci i prodavci - mogu nesmetano razmenjivati podatke, pri čemu XML definiše format podataka.

**Peto**, XML je **javni format**, što znači da nije u privatnom vlasništvu bilo koje kompanije. Njegova javna priroda podrazumeva da se za *njegovu upotrebu ne mora kupovati licenca*, niti plaćati naknada.

**Šesto**, XML uživa široku podršku proizvođača softvera. Naime, XML u podjednako meri podržavaju i **Microsoftovi** i **Netscapeovi web** čitači. Proizvođači softvera sve više inkorporiraju karakteristike XML-a u svoje baze podataka i najrazličitije preduzetničke aplikativne programe.

Tabela 12.1: Prednosti XML-a
Zasnovan je na Webu
Ima mogućnost proširenja
Nezavisan je od upotrebljene kompjuterske i softverske platforme
Omogućava "s kraja na kraj" integrisanje aplikacija
Predstavlja javni standard, za čiju upotrebu se ne mora plaćati licenca
Ima široku podršku od strane proizvođača softvera

## Definicija tipa dokumenta

Ukoliko nekoliko zainteresovanih strana (na primer, timovi naučnih istraživača, lekara ili poslovnih partnera) želi da redovno razmenjuje podatke, putem odgovarajućih, za Web osposobljenih aplikacija, onda one mogu među sobom **sklapati precizne ugovore** o vrsti i tipu podataka koje će razmenjivati. Takozvanom *definicijom tipa dokumenta* (**document type definition** - DTD) definiše se **rečnik** (ili standard) koji podatke čini lako razumljivim i jednostavnim za upotrebu. (DTD možete posmatrati kao *rečnik* ili *šemu*, koja svim korisnicima XML dokumenata omogućava da eliminišu svaku sumnju u vezi sa značenjem pojedinih termina.)

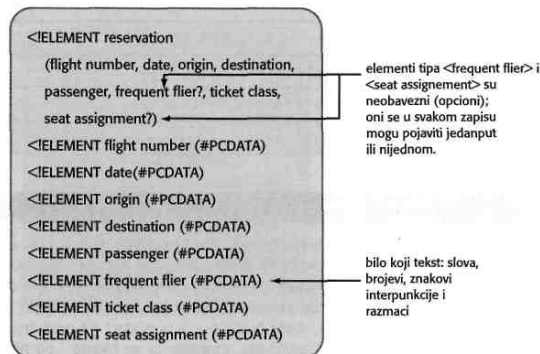
DTD predstavlja **formalan opis nekog konkretnog XML dokumenta**, koji je napisan primenom XML deklarativne sintakse. On definiše nazive koji su upotrebljeni za različite tipove elemenata, lokacije na kojima se ti elementi mogu pojavljivati i načine njihovog spajanja u celinu. **Primer**a radi, grupa avio-kompanija kreira DTD, kako bi se svim preduzećima stavilo do znanja da se web podaci nalaze u konzistentnom formatu. Da bi se nekim tipom dokumenta opisale liste koje u sebi sadrže stavke (**items**), odgovarajući deo DTD dokumenta (slika 12.3) bi se mogao sastojati od sledećeg koda:

```
<(ELEMENT List (Item) + >  
</ELEMENT Item (#PCDATA)>
```

Ovaj kod definiše listu kao tip elementa koji u sebi sadrži jednu ili više stavki (na šta ukazuje znak "+"); nadalje, on definiše stavke kao tipove elementa koji se sastoje isključivo od običnog (plain) teksta (**Parsed Character Data** ili PCDATA).

Aplikacija očitava DTD pre nego što pristupi obradi pristiglog web dokumenta, tako da ona unapred zna redosled pristizanja pojedinih tipova elemenata i relacije među njima. DTD unapred pruža aplikacijama obaveštenje o tome koji se nazivi i strukture mogu upotrebiti u nekom konkretnom tipu dokumenta. Dizajner date aplikacije može, oslanjajući se na DTD, pretpostaviti da će svi dokumenti koji pripadaju određenom tipu biti konstruisani i imenovani na dosledan i prikladan način.

Projektant se može opredeliti i za pisanje potpuno novog DTD-a, koji će zadovoljiti konkretne zahteve date aplikacije. Ipak, treba imati u vidu da već postoji na hiljade gotovih DTD-a. Industrijska udruženja uspostavljaju DTD definicije tako što kompanijama iz određene grane industrije omogućavaju da se međusobno dogovaraju oko standarda za deljenje/razmenu podataka. Primera radi, *RosettaNet* je jedan neprofitni konzorcijum koji okuplja trgovačke partnere koji učestvuju u lancima snabdevanja. Članice ovog konzorcijuma su, između ostalih, kompanije *American Express*, *Hewlett Packard*, *Microsoft*, *Oracle* i *United Parcel Services*. RosettaNet je kreirala DTD definicije koje partnerima u trgovinskoj razmeni omogućavaju komunikaciju upotrebom XML-a.



Slika 12.3 Primer DTD-a

## Predstavljanje uz pomoć XML stilova

Jedna od prednosti XML-a ogleda se u jasnom razdvajanju sadržaja podataka i načina njihovog predstavljanja. Međutim, kada projektanti ili objavljiivači XML podataka (na primer, kompanija koja objavljuje red letenja svojih aviona) žele da upravljaju prikazom i predstavljanjem informacija u nekom dokumentu, oni tada kreiraju takozvane *XML liste stila (style sheets)*, upotrebom **XML stylesheet language (XSL)** jezika. Prema tome, XSL definiše način konvertovanja nekog XML fajla, kako bi on u web čitaču mogao biti prikazan u željenom formatu.

**Primer:** U nekom XML fajlu projektant može da deklariše sledeći tag:

**<showbold> Ovo je tekst zaglavlja </showbold>**,

pri čemu bi tekst unutar ovih tagova želeo da prikaže upotrebom različitih fontova i boje slova. U tom slučaju, u XSL fajlu bi pomenuti tag trebalo deklarirati kao:

```

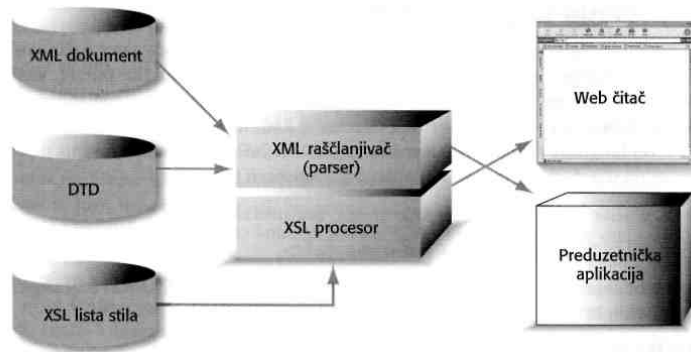
<xsl:template match= showbold >
  <bxfont color= #FF0000 face= Arial size= 3 > <xsl:apply-templates/>
  </fontx/b>
</xsl:template>
  
```

Čim se, prilikom obrade datog XML fajla naide na tag *showbold*, tekst – **Ovo je tekst zaglavlja** - biće ispisan drugačijim fontom (Arial 3) i bojom (FF0000, što je heksadecimalni kod za crvenu boju), uz podebljanje teksta (bold). XML stilovi su naročito korisni u slučajevima kada projektant želi da kontroliše izgled izlaznih rezultata, odnosno kada taj izgled treba da varira u zavisnosti od toga da li su podaci prikazani na velikom kompjuterskom terminalu ili malom displeju nekog PDA uređaja.

## Obrada XML-a

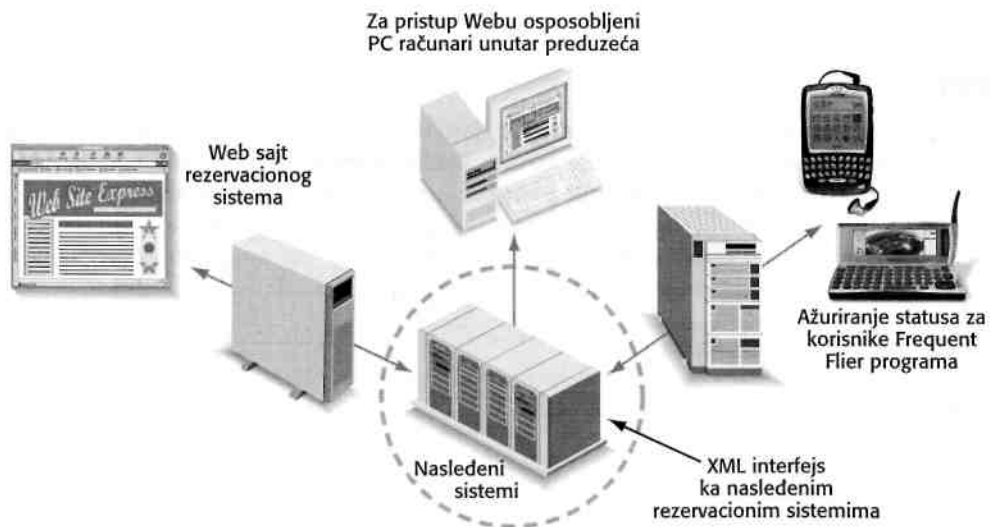
XML se, dakle, sastoji od sledećih komponenata: XML dokument, DTD i XML lista stila. Obrada započinje tako što odgovarajuća aplikacija skenira XML dokument i DTD (ovaj proces se naziva *rašćlanjivanjem*, ili *parsing*) radi identifikacije podataka. Zatim se vrši obrada XML listova stila kako

bi se odredio format za prezentovanje ili prikazivanje podataka (videti sliku 12.4). Rezultati mogu, na primer, biti predstavljani u **prozoru nekog web čitača**. Alternativno, oni mogu biti poslani ka nekoj drugoj **aplikaciji**, gde će podaci biti upotrebljeni za izvršavanje njenih aktivnosti na obradi.



Slika 12.4 Obrada XML komponenata

Na slici 12.5 upotreba XML-a ilustrovana je na primeru jednog sistema za rezervaciju avionskih karata. Ovi sistemi se prilikom upravljanja redovima letenja i putničkim rezervacijama, tipično oslanjaju na dobro razrađene nasleđene sisteme. Pa ipak, veliki broj avio-kompanija je razvio mogućnost online prodaje karata preko Interneta, kao i slanja ažuriranih informacija o letovima direktno ka displejima putničkih PDA uređaja i mobilnih telefona. Dovoljan je XML i neki mali program za obradu podataka pa da se dobije interfejs sa nasleđenim sistemom i sredstvo njegovog efikasnog oposobljavanja za Web. Prednosti XML-a podstiču najšire interesovanje za njegovu upotrebu u mnogim tipovima preduzetničkih aplikacija; stoga je gotovo sigurno da će se njegov razvoj nastaviti i u budućnosti.



Slika 12.5 XML interfejs namenjen za Web osposobljenim nasleđenim sistemima

# Web servisi

**Web servisima** sa nazivaju **modularne Internet aplikacije**, koje izvršavaju neke sasvim specifične funkcije i zadatke. S obzirom na to da su napisani u skladu sa **standardnim tehničkim formatom**, oni se mogu **međusobno povezivati** i neprimetno integrisati sa drugim web servisima radi kreiranja za Web osposobljenih (*Web-enabled*) aplikacija. Web servisi predstavljaju glavni pravac razvoja u budućnosti.

## Principi na kojima se zasnivaju **web servisi**

**Primer:** Razvoj web servisa započeo je istovremeno sa komercijalizacijom Interneta, mada u početku većina korisnika uopšte nije bila svesna njihovog značaja. Uzmimo za primer **Yahoo!**, jedan od najpopularnijih **Internet portala**. Svako ko prvi put poseti Yahoo!, odmah će na njemu uočiti širok asortiman ponuđenih servisa, među kojima su: vremenska prognoza, finansijski servisi, turističke informacije, kompjuterske igre i drugo. Prikladnom kombinacijom korisnih servisa Yahoo! je uspeo da postane jedan od najčešće posećivanih sajtova na Internetu.

Mada se svi ovi servisi obično *pripisuju* Yahoo!-u, većina njih je, u stvari, *samo dostupna preko* Yahoo!-a, a ne i kreirana od strane ove kompanije. Pravi ponuđači ovih servisa su neke druge firme, koje svoje usluge samo čine dostupnim preko Yahoo! web sajta.

**Uopštimo:** Naime, kao i kod Yahoo!-a, nezavisni projektanti kreiraju aplikativne module (koji su analogni vremenskim, finansijskim i putničkim servisima), koje zatim čine dostupnim drugim projektantima preko Interneta. Ovi poslednji zatim inkorporiraju izabrane modularne aplikacije i koriste ih kao sastavne komponente svojih preduzetničkih aplikacija. Međutim, kao i kod Yahoo!-a, softverski kod ovih modula nije fizički *umetnut* (embedded) u datu aplikaciju. Umesto toga, oni su *međusobno povezani*, pri čemu se softver fizički nalazi na web sajtu provajdera, odnosno ponuđača servisa.

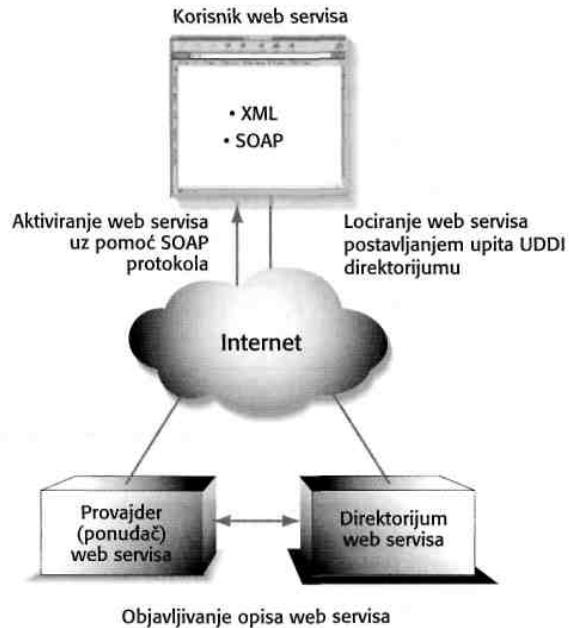
Alati koji obezbeđuju ove mogućnosti su: SOAP, WSDL i UDDI. Zajedno sa XML-om ovi alati čine srž tehnologije za kreiranje Web osposobljenih aplikacija (tabela 12.2).

NAZIV TEHNOLOGIJE	AKRONIM	FUNKCIJA
<i>Extensible Markup Language</i>	<b>XML</b>	XML predstavlja skup pravila za kreiranje tagova kojima se opisuje sadržaj i prikaz web podataka
<i>Simple Object Access Protocol (ili Services Oriented Architectural Protocol)</i>	<b>SOAP</b>	Protokol za aktiviranje (invoking) udaljenih objekata web servisa uz pomoć XML-a
<i>Universal Discover, Description, and Integration</i>	<b>UDDI</b>	Na XML-u zasnovan, javni direktorijumski servis, koji pruža informacije o raspoloživim web servisima
<i>Web Services Description Language</i>	<b>WSDL</b>	Protokol koji opisuje mogućnosti nekog objekta web servisa.

## Simplified Object Access Protocol (SOAP)

Takozvani protokol za pristup jednostavnim objektima (**simple object access protocol** - SOAP) jeste protokol koji neka aplikacija može upotrebiti za aktiviranje odgovarajućeg web servisa, koji je smešten na nekom drugom kompjuteru. Ovaj protokol, dakle, omogućava međusobnu komunikaciju različitih aplikacija (to jest, objekata), smeštenih bilo gde unutar **intraneta** ili na javnom Internetu, bez obzira na razlike u tipu kompjutera na kojima se te aplikacije izvršavaju (slika 12.6). Prema tome, SOAP igra ključnu ulogu u **obezbedenju nesmetane interoperabilnosti heterogenih sistema**.





Slika 12.6 Interakcije između učesnika u web servisima

**Procedura:** Objekat jedne SOAP poruke, izražen u XML-u, šalje se ka udaljenom web servisu sa zahtevom za obradu. Na drugoj strani, web servis obavlja zahtevanu obradu i kao reakciju (odgovor) šalje rezultat izvršavanja određenog metoda.

Primera radi, neka avio-kompanija bi mogla poželeti da svojim komercijalnim klijentima omogući korišćenje servisa za **praćenje transporta robe**. Umesto da od svakog svog klijenta zahteva da kôd za unošenje broja pošiljke i ostalih detalja ugradi u sopstveni aplikativni softver, ta avio-kompanija bi mogla da kreira poseban web servis koji bi u sebi sadržao sav neophodan kompjuterski kôd za praćenje pošiljaka. Na taj način klijent bi morao samo da, uz pomoć sopstvenog programa, pošalje odgovarajuću SOAP poruku sa informacijama o traženoj pošiljci ka centralnom kompjuteru avio-kompanije. Ova poruka bi, naime, aktivirala pomenuti web servis za praćenje pošiljaka i zatim klijentu poslala izveštaj o lokaciji na kojoj se tražena pošiljka trenutno nalazi.

Projektant aplikacije koja vrši aktiviranje nekog web servisa uopšte ne mora da poznaje tehničke detalje tog servisa (na primer: jezik u kome je programiran, tip operativnog sistema ili vrsta kompjutera na kome se taj servis izvršava). Njega interesuje samo da li taj web servis može da generiše korektan rezultat i da mu ga pošalje u očekivanom formatu.

Na taj način SOAP oslobada projektante potrebe za pisanjem programskog koda radi dobijanja željenih rezultata ili za poznavanjem načina na koji je određeni web servis implementiran na nekom udaljenom kompjuteru. Za razvoj SOAP protokola zadužen je W3C konzorcijum.

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=
    "http://www.w3.org/2001/06/soap-envelope"
  SOAP-ENV:encodingStyle=
    "http://www.w3.org/2001/06/soap-encoding"

  <SOAP_ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>IBM</symbol>
    </m:GetLastTradePrice>
  </SOAP_ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP omotnica  
(envelope)

### Povratna SOAP poruka

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=
    "http://www.w3.org/2001/06/soap-envelope"
  SOAP-ENV:encodingStyle=
    "http://www.w3.org/2001/06/soap-encoding"

  <SOAP_ENV:Body>
    <m:GetLastTradePrice
      xmlns:m="Some-URI">
    </m:GetLastTradePrice>
  </SOAP_ENV:Body>
</SOAP-ENV:Envelope>
```

Rezultat se šalje unutar tela poruke

SOAP omotnica  
(envelope)

Slika 12.7 Primer jednog SOAP odgovora na pristiglu poruku