Inzenjering dokumenata je inzenjerska disciplina ciji su predmet dokumenti i poslovni procesi u kojima se oni razmenjuju. Koristi metode za analizu i dizajn dokumenata i poslovnih procesa i srodne metode kao sto su analiza sistema, poslovna analiza, analiza zadataka, itd. Inzenjering dokumenata je jedan pristup analizi i dizajnu informacionih sistema.

Dokument je fiksna i strukturirana kolicina informacija kojom se upravlja kao jedinicom i koja se razmenjuje kao jedinica izmedju korisnika i sistema. Predstavlja **jedinicni nosilac informacija izmedju elemenata informacionog sistema**, moze imati strukturu i ima zivotni ciklus (tj. Menja stanje kroz razlicite faze od nastanka do nestanka).

Transakcioni (jako strukturirani) dokumenti: nalog za placanje, vozacka dozvola, zapisnik sa ispita.

Narativni (slabo strukturirani) dokumenti: knjiga, clanak(Wikipedia), zakon. Treba praviti razliku izmedju sadrzaja dokumenta i prezentacije dokumenta.

Metapodaci (meta + data): data – ono sto je dato(latinski), meta – iza (starogrcki). Podaci o podacima, pojam metapodataka je relativan.

Identifikator je objekat koji moze da sluzi kao referenca na nesto sto ima identitet. Identifikator je skup metapodataka koji identifikuju dokument.

Poslovni proces je skup aktivnosti u kojima se jedan ili vise ulaza transformise u jedan ili vise izlaza radi ostvarivanja planiranog cilja. Ulazi i izlazi mogu biti informacije. Aktivnosti (tj. Transformacije ulaza u izlaze) se obavljaju rucno ili automatski.

Sistemi za upravljanje dokumentima su informacioni sistemi koji omogucavaju upravljanje zivotnim ciklusom dokumenata (tj. Inicijalizaciju, pripremu, uspostavljanje, koriscenje, reviziju, arhiviranje i brisanje dokumenata).

Osnovne funkcije sistema za upravljanje dokumentima su: rad sa dokumentima, rad sa metapodacima, indeksiranje i pretraga dokumenata, definisanje poslovnih procesa i upravljanje zivotnim ciklusom dokumenata, saradnja izmedju korisnika i upravljanje zivotnim ciklusom dokumenata, saradnja izmedju korisnika i upravljanje verzijama dokumenata, obezbedjivanje bezbednosti, integracija sa drugim informacionim sistemima. Obicno se implementiraju koristeci sisteme za upravljanje radnim tokovima (workflow management systems) koji posmatraju dokument kao centar radnog toka.

Sistemi za upravljanje radnim tokovima su sistemi koji omogucavaju definisanje, izvrsavanje i nadgledanje izvrsavanja radnih tokova. Nazivaju se i sistemi za upravljanje radnim tokovima bazirani na dokumentima (document-based workflow management systems).

Oblasti primene inzenjerstva dokumenata su: elektronsko izdavastvo, poslovni informacioni sistemi, eUprava, eObrzovanje, eZdravstvo, itd.

W3C (World Wide Web Consortium) je neprofitna organizacija zaduzena za standardizaciju world wide web-a. W3C preporuke su javno dostupne.

Resurs moze biti bilo sta sto ima **identitet** (elektronski dokument, slika, servis, kolekcija resursa). **Informacioni resursi su resursi cije se bitne karakteristike mogu preneti u poruci**. Informacioni resursi tipicno imaju jednu ili vise reprezentacija kojima se moze pristupiti koristeci HTTP protokol. Neinformacioni resursi su resursi koji nisu infomacioni resursi (apstraktni resursi).

Reprezentacija resursa je informacija koja reflektuje proslo, trenutno ili zeljeno stanje resursa, u formatu koji moze da se lako komunicira putem protokola i koja se sastoji iz skupa reprezentacionih metapodataka i potencijalno neogranicenim tokom reprezentacionih podataka. Svaki resurs moze imati vise reprezentacija (XHTML, XML, RDF, JSON, JPEG).

Unicode je standard za konzistentno kodiranje, predstavljanje i rukovanje tekstom, sadrzi vise od 120 000 znakova koji pokrivaju vise nacina (UTF-8, UTF-16 i UTF-32). UTF-8 koristi jedan bajt za kodiranje ASCII znakova i do cetri bajta za

kodiranje ostalih znakova, UTF-16 koristi dva bajta za kodiranje UCS-2 znakova i cetri bajta za kodiranje ostalih znakova. UTF-32 koristi cetri bajta za kodiranje znakova.

URI (Uniform Resource Identifier) je niz znakova za **identifikovanje** apstraktnih ili fizickih **resursa**. URL (Uniform Resource Locator), URN (Uniform Resource Name), URL i URN. URI sema definise prostor imena identifikatora i moze dalje da ogranice sintaksu i semantiku identifikatora (HTTP, FTP, mailto).

URL (Uniform Resource Locator) je podskup URI koji identifikuje resurse preko reprezentacije njihovog primarnog mehanizma pristupa (npr. njihove lokacije na mrezi).

URN (Uniform Resource Name) je podskup URI koji mora ostati globalno jedinstven i perzistentan cak i ako resurs prestane da postoji ili postane nedostupan i bilo koji drugi URI sa svojstvima imena.

HTTP (Hyper-Text Transfer Protocol) je komunikacioni protokol aplikativnog nivoa namenjen prenosu (hiper) teksta. Protokol funkcionise po klijent server arhitekturi, klijent salje serveru zahtev, server salje klijentu odgovor. Server ne cuva stanje klijenta. Struktura zahteva: linija zahteva nula ili vise polja u zaglavlju, prazna linija, telo poruke (opciono).

HTTP metode:

- GET – zahteva prenos izabrane reprezentacije ciljnog resursa.
- HEAD – isto kao i GET metoda, ali prenosi samo statusnu liniju i polja u zaglavlju.
- POST – zahteva da server obradi reprezentaciju sadrzanu u zahtevu prema specificnoj semantici ciljnog resursa.
- PUT – zahteva da se stanje ciljnog resursa izmeni prema stanju definisanom reprezentacijom sadrzanom u zahtevu.
- CONNECT
- OPTIONS – zahteva informacije o dostupnim komunikacionim opcijama.
- TRACE

HTTP metode su sigurne metode, ne menjaju stanje servera. Idempotentne metode - vise identicnih zahteva ima isti efekat kao jedan zahtev.

HTTP Response se sastoji od statusne linije, nula ili vise polja u zaglavlju, prazne linije i tela poruke (opciono).

HTTP status kodovi:

- 1xx – informational (zahtev je primljen i obradjuje se)
- 2xx – success (zahtev je uspesno primljen i prihvacen)
- 3xx – redirection (klijent mora da izvrsi dodatnu akcju da bi se zahtev obradio)
- 4xx – client error (zahtev sadrzi neispravnu sintaksu ili ne moze biti ispunjen)
- 5xx – server error (server nije uspeo da ispuni ispravan zahtev)

Web Architectural Principles

- URL za identifikovanje resursa
- HTTP protokol
- Standardno znacenje HTTP metoda
- Standardno znacenje HTTP status kodova
- Povezati resurse (linkovi)
- Server ne cuva stanje klijenta (statless)
- Visestruke reprezentacije resursa

XML Information Set je konceptualni model XML dokumenta koji je osnova za druge W3C preporuke (standarde). Elementi modela su informacione stavke (information items) i njhove veze.

11 tipova informacionih stavki (dokumenti, elementi, atributi, procesne instrukcije, neprosirene reference na entitete, znaci, komentari, deklaracije tipova dokumenata, neprasirani entiteti, notacije i prostori imena). Informacioni skup se moze shvatiti kao stablo, a informacione stavke kao cvorovi.

RDF (Resource Description Framework) je standardni model za razmenu podataka na web-u. RDF prosiruje strukturu web-a tako sto koristi URI za imenovanje stvari i relacija izmedju stvari. Ova struktura formira usmereni oznaceni graf (cvorovi su stvari a grane relacije izmedju stvari). Graf se moze predstaviti kao skup iskaza oblika <subjekat> <predikat> <objekat>.

XML tehnologije: XML, DTD, XML Namespaces, XML Schema, XPath, XQuery, XPointer, XLink, XSL-T, XSL-FO, DOM.

Linked Data tehnologije: RDF, RDFa, RDFS, SPARQL.

Markup je nacin da se oznaci prezentacija, struktura ili znacenje teksta. Markup je koriscen i na papiru prilikom slaganja teksta za oznacavanje njegovih delova. Markup se ne prikazuje eksplicitno krajnjem korisniku.

XML dokumenti ne sadrze unapred definisane elemente i atribute, nego ih autori sami definisu. XML je metajezik, jezik za definisanje drugih jezika, skup pravila za definisanje drugih jezika. HTML se moze posmatrati kao jedan dijalekt XML-a (XHTML).

Dijalekti XML-a:

- CrossRef – povezivanje bibliografskih podataka
- CML – opis hemijskih jedinjenja
- DocBook – tehnicka dokumentacija
- SVG – vektorska grafika
- MathML – matematicke formule

Ciljevi XML-a:

- Upotreba na internetu
- Koristi se od stane razlicitih aplikacija
- (jednosmerna) kompatibilnost sa SGML-om
- Jednoznacnost prilikom automatske obrade
- Citki i samorazumivi
- Konciznost nije bitna
- Lako se dizajniraju
- Lako se pisu aplikacije za obradu XML dokumenata

Struktura XML dokumenta:

- XML dokument implementira XML informacioni skup
- Sadrzi cvorove razlicitog tipa uredjene u stablo
- Oznake su nacin da se hijerarhijska struktura serijalizuje u linearnu strukturu (niz znakova)

Tipovi cvorova: dokumenti, elementi, atributi, znaci, komentari, procesne instrukcije, itd..

Procesne instrukcije - niz znakova izmedju <? i ?>. Predstavljaju instrukcije programima, nisu namenjene ljudima.

XML deklaracija – vrsta procesne instrukcije, nalazi se na pocetku dokumenta, sadrzi tri atributa: **version** (verzija XML preporuke koja se koristi u dokumentu), **encoding** (kodni raspored koji se koristi u dokumentu), **standalone** (da li je potrebno ucitati i DTD).

CDATA sekcije – niz znakova izmedju <![CDATA[i]]>. Tekst koji se interpretira bez zamene entiteta.

Dobro formiran dokument

XML dokument mora biti **dobro formiran da bi mogao da se masinski obradi**. Specijalni znaci mogu da se nadju u dokumentu samo u svojoj ulozi. Moraju se postovati pravila imenovanja elemenata i atributa. Dokument mora imati jedan korenski element. Elementi se ne mogu preklapati. Vrednost atributa mora biti unutar navodnika (jednostrukih i dvostrukih). Element ne moze imati dva atributa sa istim imenom. Komentari i procesne instrukcije ne mogu se nalaziti unutar taga.

Markup jezici i XML

Markup je oznaka (tag, code) koja opisuje deo sadrzaja. Markup je koriscen i na papiru prilikom slaganja teksta za oznacavanje pojedinih delova (naslovi, citati, brojevi poglavlja, itd...). Markup ne predstavlja sadrzaj, vec **uputstvo kako neki sadrzaj obraditi ili prikazati**. Markup se najcesce ne prikazuje eksplicitno. Krajnji citalac ne vidi markup tagove. Ako se markup koristi za definisanje prezentacije dokumenata on je vidljv iz grafickog izgleda pojedinih elemenata (broj poglavlja, glavni naslov, podnaslov, citat).

Markup tagove definisu **firme** – definicije su u vlasnistvu firme, korisnik ne mora da zna sve detalje o markupu, Microsoft: Word .doc format, Adobe: PostScript, PDF ili **otvoreni standardi** – definisu ih tela za standardizaciju(ISO, W3C,…), svi detalji su poznati i javno dostupni(SGML, HTML, XML).

Nejasan (nedorecen, viseznacan) markup u formatiranju, oslanjanje na vizuelni izgled delova teksta kao opis znacenja.

Razdvajanje strukture i prikaza, osnovni razlozi za pojavu sistema sa nejasnim markupom, fokus na prikaz dokumenata, ogranicenost na sopstveni, zatvoreni sistem. Otvoreni standardi za markup se zasnivaju na konceptima markiranja strukture i markiranja znacenja.

HTML – dokument je hijerarhija elemenata, mogu se koristiti samo unapred definisani tagovi. **Dokument je osnovni nosilac informacija, osnovna jedinica razmene informacija.**

Tri karakteristike dimenzije dokumenta su sadrzaj, struktura, prezentacija.

XML – ne definise nikakve tagove unapred, vec ih korisnik sam definise. XML je jezik za definisanje markup jezika, tj. Metajezika. HTML se moze posmatrati kao jedan od jezika definisanih pomocu XML-a. XML podrazumeva familiju standarda.

XSL (Extensible Stylesheet Language) - vizuelizacija, **XPath** – oznacavanje strukture dokumenata, **XLink** – povezivanje dokumenata, **XQuery** – pretrazivanje dokumenata, po sadrzaju i strukturi.

Ciljevi XML-a:

- da odgovara upotrebi na Internetu,
- mogucnost koriscenja od strane razlicitih aplikacija,
- (jednosmerna) kompatibilnost sa SGML-m,
- da se programi za obradu XML dokumenata pisu lako,
- jednoznacnost prilikom obrade XML dokumenta,
- XML dokumenti citkiji i rezonski jasni,
- dizajn XML-a je formalan i koncizan,
- kreiranje XML dokumenata je lako, konciznost je od minimalne vaznosti.

Tagovi su nacin da se hijerarhijska struktura (stablo elemenata) prikaze pomocu linearne strukture (tekstualnog dokumenta, tj. niza slova).

Dobro formirani XML dokument ima jedan korenski element, elementi se mogu ugnjezdavati ali ne i preklapati, vrednost atributa moraju biti unutar navodnika, element ne moze imati dva atributa sa istim imenom. **XML dokument mora biti dobro formiran da bi mogao biti masinski obradjivan.**

XML za razmenu podataka

XML dokumenti su i human-readable i machine-readable. Softver za rad sa XML dokumentima postoji na svim racunarskim platformama. Za uspesnu razmenu podataka potrebno je da svi ucesnici koriste isti format. Ako koriste XML, format moze da se definise u posebnom dokumentu, dokument koristi posebnu sintaksu, moze se publikovati svim zainteresovanim ucesnicima, predstavlja specifikaciju poruka koje se koriste u komunikaciji.

Document Type Definition (DTD) je deo osnovnog XML standarda. DTD fajl (dokument) propisuje format klase/familije/tipa XML dokumenata, koji elementi i entiteti se mogu pojaviti na kom mestu u dokumentu, sta je sadrzaj elemenata i atributa. DTD se obicno cuva kao poseban fajl. Prethodni DTD je definisao klasu validnih dokumenata, dokument je validan ako odgovara svom DTD-u.

XML Namespaces

Pojam prostora imena - **skup elemenata koji imaju isti prefiks nazvacemo prostor imena** (namespace). Prostor imena je razliciti od tipa dokumenta, prostor imena je skup elemenata, jedan tip dokumenata moze ukljucivati elemente iz vise prostora, jedan element moze biti koriscen u vise tipova dokumenata. Koncept prefiksa nije dovoljno robustan, umesto da se formira novi sistem za administraciju prefiksa, moze da se koristi postojeci – Internet.

Identifikacija prostora imena

Prostori imena se identifikuju nazivom, naziv je niz znakova u URI formatu, URI – Uniform Resource Identifier.

Uniform Resource Identifier (URI)

Resurs je sve sto ima identitet, fizicki objekat (npr. Knjiga), digitalni objekti (fajlovi), apstraktni pojmovi (stanje saobracaja u Novom Sadu). Resurs moze biti dostupan preko Interneta, ali i ne mora. URI standard, IETF RFC 2396, opisuje identifikatore resursa. URI moze biti formiran kao URN ili URL.

Uniform Resource Locator (URL)

Standart definisan u IETF RFC 1738, **predstvlja podatke koji se mogu upotrebiti za dobavljanje resursa**.

Universal Resource Name (URN)

IETF RFC 2141, definise trajno ime resursa koje ne zavisi od njegove lokacije. URN format: urn:nid:nss. urn – fiksno, nid – namespace identifier, nss – namespace specific string. Ako se koristi URL format za identifikator, uvek se koristi puna adresa.

Identifikacija elemenata u prostoru imena

Elementi se identifikuju kvalifikovanim imenima (qualified name, QName). Kvalifikovano ime se sastoji iz dva dela: lokalno ime i namespace. U XML dokumentima se umesto naziva prostora imena koristi kraci XML-dozvoljeni prefiks. Prefiks mora da se poveze sa svojim prostorom imena. Ime prefiksa vise nije bitno, moze biti bilo koje XML dozvoljeno ime.

Deklarisanje prostora imena

U jednom elementu moze se deklarisati vise prostora imena, ne moraju svi prostori imena biti deklarisani u korenskom elementu. Prefiksi imaju znacenje samo u okviru elementa u kome su definisani.

XML Schema

Sta nije dobro kod DTD-a, neuobicajena ne-XML sintaksa, slaba podrska za XML prostore imena, nema tipizacije podataka, narocito za sadrzaj elemenata, nije moguce definisati datumska ili numericka polja. Kod DTD-a postoji 10 tipova, kod XML Schema 45. Ogranicena prosirivost, nije moguc reuse tipova podataka. Ogranicene mogucnosti za

opisivanje strukture podataka, ne moze se nametnuti broj podelemenata bez nametanja redosleda, ne moze se nametnuti redosled i broj podelemenata kada se koristi mesani sadrzaj.

Sema je XML dokument koji opisuje strukturu drugih dokumenata. Prevazilazi mane DTD-a, seme se pisu kao XML dokumenti, a ne koriscenjem sintakse nasledjene od SGML-a. Potpuna podrska za rad za XML namespaces, validacija tekstualnog sadrzaja na bazi ugradjenih ili novo definisanih tipova podataka (datum, ceo broj, itd). Kreiranje i jednostavno ponovno koriscenje novih tipova podataka. Sema prevazilazi mane DTD-a, koncepti podsecaju na OOP, definisanje novih tipova podataka na osnovu postojecih (mehanizmi prosirivanje ili restrikcija). Moze da predstavi skupove (nije bitan redosled podelemenata), jedinstvni sadrzaj elemenata u nekom regionu, vise elemenata istog imena ali razlicitog sadrzaja, null sadrzaj, zamenljivi elementi... Seme nemaju funkcionalnost rada sa entitetima, DTD moze biti ugradjen direkno u dokument koji opisuje dok seme moraju uvek biti smestene u posebne fajlove. Softver i dalje potpuno podrzava rad sa DTD-ima. Sa upotrebom sema dolazi do manje posla i standardizacije – svi ucesnici u jednom sistemu koriste standardan format za specifikaciju podataka, otvoreni sistemi. Mogucnosti za dalji razvoj: generator GUI ekranskih formi na osnovu sema, generator programskog koda (modela podataka), "pametni editori", pomazu pri unosu dokumenata.

Sema standardi

Postoji vise standarda za sema jezike: W3C XML Schema, RELAX NG, Schematron.

XML Schema je standard koga propisuje W3C: najrasireniji, najmocniji, najkomplikovaniji.

W3C XML Schema je kreirana od strane W3C XML Schema Group, na osnovu mnogih predloga.

Tip elementa moze da se navede u okviru njegove deklaracija – **anonimni tip,** kao tip elementa moze da se navede neki postojeci tip – **imenovani tip.** Ako ocekujemo da ce se isti tip pojavljivati na vise mesta, isplati se definisati ga na jednom mestu – koristicemo imenovani tip. Ako ne zelimo da se definicija tipa vidi kao nesto sto je dostupno korisniku seme – koristicemo anonimni tip. Imenovani tip – ne moze se istovremeno referencirati tip i definisati ugnjezdeni tip.

Tipizacija podataka u W3C XML Schema

Prosti tipovi podataka, ne mogu imati podelemente ili atribute. Mogu nastati restrikcijom ugradjenih tipova. Slozeni tipovi podataka mogu imati podelemente ili atribute.

4 osnovna elementa seme

- xsd: element deklarise element i dodeljuje mu tip
- xsd:attribute deklarise atribut i dodeljuje mu tip
- xsd:simpletype definise novi prosti tip
- xsd:complexType definise novi slozeni tip

Deklarisu se one komponente koje imaju reprezentaciju u instancama seme, definisu se one komponente koje se koriste u okviru seme. Deklarisu se elementi i atributi, definisu se tipovi, grupe atributa, itd..

Definicija tipa

Svaka deklaracija novog tipa se oslanja na neki od postojecih (vec definisanih) tipova.

Ugradjeni tipovi – definisani u XML Schema Part 2:Datatypes: logicki, tekstualni, URI, numericki, vremenski, XML tipovi.

Definicija novog prostog tipa

Novi prosti tipovi se definisu na osnovu postojecih – ugradjenih.

Mehanizam definisanja novog tipa

- Restrikcija – navodnje ogranicenja na vrednost novog tipa

- Lista – definisanje konacne liste mogucih vrednosti
- Unija – kombinovanje vrednosti vise razlicitih osnovnih tipova

Lax validacija – validator ne proverava one elemente za koje ne postoji sema.

Strict validacija – validator ne insistira da svi elementi imaju semu i proverava ih u odnosu na nju

Jedinstvenost i kljucevi, kod DTD-a moguce je koristiti tip ID atributa za osiguravanje jedinstvene vrednosti atributa. XML Schema ima vece mogucnosti. Sadrzaj elementa moze biti jedinstven (unique). Vrednosti atributa koje nisu tipa ID mogu biti jedinstvene. Kombinacija sadrzaja elemenata i atributa moze biti jedinstvena, moze se definisati opseg u kome se proverava jedinstvenost. Pravi se razlika izmedju jedinstvene vrednosti (unique) i kljuca (key).

Kvalifikovanje elemenata - u dokumentu smo kvalifikovali sve elemente (i globalne i lokalne). Time smo rekli da svi pripadaju namespace-u. **Lokalni elementi formalno ne spadaju u targetNamespace ali jesu u njemu preko svoje veze sa globalnim elementima**. ElementFormDefault="qualified" - svi elementi u dokumentu moraju biti kvalifikovani. ElementFormDefault="unqualified" - samo globalni elementi u dokumentu mogu biti kvalifikovani, lokalni elementi ne smeju biti kvalifikovani.

Zamena tipova bazira se na sledecem principu – **osnovni tip moze biti zamenjen izvedenim tipom**.

XML Design Patterns

XML obrazac je relacija izmedju konteksta u kome se modeluju XML dokumenti, problema koji se pri tome javljaju I resenja tih problema.

Postoji 7 kategorija XML obrazaca: Document Roots, Metadata, Abstraction, Organization, Flexibility, Consistency , Miscellaneous.

Document Roots – Kako odrediti koreni element?

Metadata – Kako ukljuciti metapodatke?

Abstraction – Koje apstrakcije koristiti u dokumentu?

Organization – Kako strukturirati dokument?

Flexibility – Kako povecati ili smanjiti fleksibilnost dokumenta?

Consistency – kako konzistentno oznacavati dokument?

Misscellaneours – ostali obrasci

StAX

Postoji nekoliko programskih modela za parsiranje XML dokumenta: streaming parseri: pull (StAX parseri) I push (SaX parseri), DOM parseri, XSL-T parseri.

Streaming parseri koriste malo resursa ali imaju sekvencijalni pristup, tezi je za koriscenje.

DOM parser ima slucajan pristup, laksi za koriscenje, ali koristi puno resursa.

Push parseri implementiraju programski model u kome XML parser salje podatke programu koji ga koriste nailazeci na elemente XML informacionog skupa (elemente, atribute, tekst, itd..).

Pull parseri implementiraju programski model u kome programi koji ih koriste pozivaju metode XML informacionog skupa (element, atribut..).

StAX API implementira sto znaci da program trazi podatke od StAX parsera onda kada su mu potrebni. Parser cita XML dokument od pocetka do kraja I prepoznaje elemente XML informacionog skupa (tokene koji cine dobro formiran XML dokument). StAX API moze da se koristi I za citanje I za pisanje XML dokumenata (za razliku od SAX API-ja). XML dokumente moze da parsira koriscenjem iteratora (iterira kroz listu dogadjaja da bi dobio podatke) ili kursora (podatke dobija preko pokazivaca na XML cvorove).

Cursor API koristi kursor koji se kre¢e od pocetka do kraja XML dokumenta i pokazuje na elemente XML informacionog skupa.

Iterator API predstavlja XML dokument kao niz diskretnih dogadjaja (koji odgovaraju XML informacionom skupu) koji se mogu obraditi.

SAX – Simple API for XML

SAX koncept

Parsiranje pomocu SAX-a je event-driven. Parser tokom parsiranja generise dogadjaje, nas kod je zaduzen da obradi dogadjaj.

SAXParserFactory – kreira instance parsera, odredjene sistemskim promenljivama.

SAXParser – interfejs sa nekoliko parse() metoda.

SAXReader – nalazi se unutar SAXparser-a. Ukoliko je potrebno da se preciznije konfigurise pozivamo metod getXMLReader(). SAXReader poziva event handlere.

ContentHandler – interfejs sa callback metodama za obradu dogadjaja vezanih za sadrzaj dokumenta.

DOM - Document Object Model

W3C standard. Standard za objektno orijentisanu reprezentaciju dokumenata sa hijerarhijskom strukturom – stablo. Varijante za razlicite jezike I za razlicite dokumente. Rezultat parsiranja je stablo objekata, stalno prisutno u memoriji, za ceo dokument, stablo se moze serijalizovati nazad u XML.

JAXB – Java Architecture for XML Binding

JAXB je framework za generisanje Java klasa na osnovu DTD ili XML Schema sema (I obrnuto) I uz transformaciju XML dokumenta u graf Java objekata (I obrnuto).

XPath

Jezik za oznacavanje delova XML dokumenata. Zasniva se na konceptu navigacije kroz stablo dokumenta. Ima bibilioteku standardnih funkcija. Neophodan za koriscenje XSLT. W3C standard.

XPath izraz – izraz je namenjen za oznacavane cvora ili skupa cvorova u dokumentu. Ovi izrazi lice na izraze za rad sa fajl sistemom.

XML dokument se sa stanovista XPath-a posmatra kao stablo.

Tipovi cvorova u XPath-u: element, atribut, tekst, namespace, procesna instrukcija, komentar, dokument(koren)

Predikati se koriste za pronalazenje cvora koji zadovoljava dati uslov, uvek se pisu unutra [].

XQuery

Standardni upitni jezik za XML. FLOWR izrazi. Centralni izraz u XQuery. Po prvom slovu klauzula: for, let, where, order by, return

XML I baze podataka

Data-centic dokumeni – obicno se koriste za razmenu podataka koji **poticu iz relacione baze**. Koriste se kao format za razmenu podataka, predvidjeni su za racunarsku obradu. Imaju regularnu strukturu, visoku granularnost, nema mesanog sadrzaja, redosled podelemenata najcesce nije vazan, podaci poticu iz relacione baze ili hocemo da ih skladistimo u relacionoj bazi.

Document-centric dokumenti – struktura I sadrzaj dokumenata nisu cvrsti, ali su **svi detalji vazni**. Predvidjeni su za coveka (knjige, email..). Neregularna ili slabo regularna struktura, niska granularnost, mesani sadrzaj elemenata, redosled podelemenata je bitan. Rucno pisani, retko poticu iz baze podataka.

Data-centric dokumenti se skladiste u relacionoj bazi, a document-centric u native XML bazi.

Skladistenje data-centric dokumenata

Mapiranje XML seme na relacionu obuhvata: elemente, atribute, tekst. Entiteti, CDATA sekcije, komentari, procesne instrukcije, redosled podelemenata se ignorisu. Dokument --> baza --> dokument nece dati originalni dokument.

XSLT

Jezik za opis transformacija XML dokumenta u druge XML dokumente. W3C standard. Za navigaciju koristi XPath izraze. Vizualizacija – samo jedna od primena.

Jedan XML dokument se transformise u drugi, transformacija je opisana XSLT dokumentom. Transformaciju izvodi XSLT procesor. XSLT fajl sadrzi definicije sablona, sablon opisuje transformaciju dela polaznog dokumenta, deo polaznog dokumenta se identifikuje XPath izrazom. Interpretiranje XSLT se svodi na primenu sablona.

XSL Formatting Objects

XSL-FO je dokument koji sadrzi opis rasporeda elemenata na stranicama. Za njegov prikaz je potreban poseban softver za renderovanje.

XLink

XLink je jezik za definisanje linkova izmedju delova dokumenata. Pojam linka je opstiji nego u HTML-u. HTML link povezuje dva resursa jednom usmerenom granom.

XPointer

XPointer je jezik za oznacavanje delova XML dokumenta. Prosiruje XPath, pronalazenje podataka poredjenjem teksa, moze da se doda na kraj URI-ja. Oznacava ne samo cele cvorove u dokumentu, vec I delove cvorova. Koristi se za povezivanje dokumenata.

RDF

DIKW

Podaci (skup simbola koji predstavljaju svojstva objekata, dogadjaja ili njihovog okruzenja).

Informacije (podaci organizovani tako da imaju znacenje).

Znanje (informacije organizovane tako da mogu da se koriste za donosenje odluka)

Dokumenti

Narativni dokumenti (slabo strukturirani)

Transakcioni dokumenti (jako strukturirani)

Linked documents – povezani dokumenti (world wide web)

Linked Data – povezani podaci (semantic web, web 3.0)

Linked Data predstavlja nacin objavljivanja podataka na (semantickom) webu koji ohrabruje ponovno koriscenje podataka, smanjuje redudantnost, maksimizira pravu mreznog efekta I dodaje vrednost podacima.

Linked Data principi

- Koristi URI za imena stvari
- Koristi HTTP URI da bi stvari mogle da se pronadju
- Kada neko trazi resurs, ponuditi odgovor u RDF formatu
- U odgovor ukljuciti RDF iskaze koji povezuju trazeni resurs sa drugim resursima da bi mogle da se pronadju srodne stvari

XML (eXtensible Markup Language) je jezik za oznacavanje strukture tekstualnih dokumenata.

RDF (Resource Description Framework) je standardni model za razmenu podataka sa WWW. Prosiruje WWW (mrezu dokumenata) tako sto koristi URI za imenovanje stvari I njihovih odnosa (I na taj nacin formira mrezu podataka). Ova struktura formira usmeren oznacen graf (ciji su cvorovi resursi I literali, a grane relacije izmedju njih). Graf se moze predstaviti kao skup iskaza oblika <subjekat> <predikat> <objekat>.

Semantic web stek

XML pruza sintaksu za strukturiranje dokumenata, ali ne namece semanticka ogranicenja na znacenje takvih dokumenata. XML Scheme je jezik za ogranicavanje strukture XML dokumenata I takodje prosiruje XML sa tipovima podataka. RDF je model podataka za objekte ("resurse") I odnose izmedju njih, pruza jednostavnu semantiku za ovaj model podatak, I omogucava da se model podataka predstavi u XML sintaksi. RDF Schema je recnik za opis klasa I svojstava RDF resursa, sa semantikom poput specijalizacije I generalizacije takvih klasa I svojstava.

RDF

Problem u pretrazivanju iste semantike predstavljene razlicitim XML stablima. Upit treba da bude nezavisan od nacina na koji je predstavljena semantika. Potreba za pretvaranjem svih mogucih predstava semantike u jedan iskaz.

Standardizovan nacin za pisanje iskaza. Kako god da se ista semantika pojavi u XML dokumentima, predstavljena je na isti nacin RDF iskazima. Vise XML stabala moze da odgovara jednom RDF grafu.

Resource Description Framework (RDF) je jos jedna World Wide Web Consortium (W3C) specifikacija. To je **graf bazirani model podataka za opisivanje stvari** (resursa) I njihovih medjusobnih odnosa. Omogucava interoperabilnost izmedju aplikacija koje razmenjuju masinski citljive I razmenljive podatke na webu.

Svaki RDF iskaz se sastoji iz subjekta (resursa), predikata (svojstva) I objekta (resursa ili literala).

Literali - tipizirane konstante. XML Schema tipovi podataka. ISO kodovi jezika.

Resursi

Stvari koje se opisuju sa RDF. **Resurs moze biti bilo sta sto ima identitet.** Informacioni resursi su resursi cije se bitne karakteristike mogu preneti u poruci. Neinformacioni resursi su resursi koji nisu informacioni resursi (apstraktni resursi). Identifikuju se sa IRI.

Za identifikaciju resursa, svojstava I tipova podataka koriste se URI.

URI (Uniform Resource Identifier) je niz znakova za identifikovanje apstraktnih ili fizickih resursa. URL (Uniform Resource Locator) je podskup URI koji identifikuje resurse preko reprezentacije njihovog primarnog mehanizma pristupa (npr. Njihove lokacije na mrezi)

URN (Uniform Resource Name) je podskup URI koji mora ostati globalno jedinstven I prezistentan cak I ako resurs prestane da postoji ili postane nedostupan.

RDF graf

Vise RDF iskaza cini RDF graf. Graf je uredjeni par (V, E). V je skup cvorova (resursa I literala). E je skup usmerenih veza (svojstava).

RDF formati

Turtle (Terse RDF Triple Language) je konkretna sintaksa za RDF. Tekstualna serijalizacija RDF grafa. Kompaktna, lako citljiva forma.

RDF/XML je konkretna sintaksa za RDF. Tekstualna XML serijalizacija RDF grafa. Obezbedjuje maksimalnu interoperabilnost.

RDFa (RDF in attributes) je konkretna sintaksa za RDF. Omogucava ugradjivanje RDF iskaza u XML dokumente pomocu standardizovanih atributa.

GRDDL (Gleaning Resource Descriptions from Dialects of Languages) je W3C specifikacija koja olaksava ekstrakciju RDF iskaza iz XML dokumenta. Obicno se RDF iskazi u RDFa formatu transformsu u RDF iskaze u RDF/XML formatu koriscenjem XSLT transofrmacije.

RDF iskazi se mogu serijalizovati u vise formata

RDFS

RDFS (RDF Schema) je **semanticko prosirenje RDF.**

Omogucava definisanje domenski specificnih klasa I svojstava. RDF Schema se zapisuje u RDF formatu. Klasa u RDFS je slicna klasi u objektno-orijentisanim programskim jezicima (skup slicnih resursa). Svojstva su (bitne) osobine tih resursa. Za razliku od objektno-orijentisanih programskih jezika, svojstva su "ravnopravna" sa klasama (takodje su resursi I takodje se mogu nasledjivati).

XML Schema deklarise elemente I atribute dokumenata odredjenog tipa. RDFS definise klase I svojstva u domenski specificnom semantickom modelu. XML Schema zadaje ogranicenja nad strukturom XML dokumenta (moze se definisati domenski specificna struktura dokumenta).

RDF Schema zadaje znacenje RDF iskaza (mogu se definisati domenski specificne klase I svojstva)

OWL (Web Ontology Language) moze se koristiti za eksplicitno predstavljanje znacenja tremina I odnosa izmedju tih termina. Pruza formalan opis koncepata, termina I odnosa u zadatom domenu. OWL ima vise mogucnosti za izrazavanje znacenja I semantike od XML, RDF I RDFS. Ova predstava termina I njihovih medjusobnih odnosa zove se ontologija.

RDFS klase I svojstva

RDF seme se specificiraju u RDF formatu pomocu predefinisanih klasa I svojstava.

RDFS specificira semantiku RDF iskaza. Omogucava definisanje klasa kojima resursi pripadaju i njihovih svojstava. RDFS se pise u RDF formatu koristeci predefinisane klase I svojstva.

SPARQL

Linked data je skup principa za objavljivanje, pronalazenje I pregledanje podataka u RDF formatu (ti podaci mogu biti distribuirani na vise servera)

Linked data je koriscenje mreze za povezivanje srodnih podataka koji nisu povezani ili smanjivanje barijere za povezivanje podataka koji su povezani koriscenjem drugih metoda.

RDF graf je skup RDF trojki (iskaza oblika subjekat predikat objekat). RDF graf se moze serijalizovati na vise nacina.

SPARQL je upitni jezik za postavljanje upita nad podacima u RDF formatu. SPARQL je protokol za postavljanje upita nad udaljenim RDF skladistima preko HTTP protokola.

SPARQL (kao upitni jezik) omogucava ekstrakciju podataka iz strukturiranih I polustrukturiranih izvora, transformaciju podataka u RDF formatu iz jednog recnika u drugi. Izvrsavanje slozenih join operacija u jednom jednostavnom upitu.

SPARQL upiti se izvrsavaju nad RDF grafovima. Upiti se mogu izvrsavati nad raznorodnim izvorima podataka, podaci su izvorno skladisteni u drugom formatu (XML, HTML, RDB) koje middleware transformise u RDF format.

RDF je model podataka zasnovan na grafovima. Podaci u RDF formatu se skladiste u RDF skladistima. SPARQL krajnje tacke omogucavaju pristup podacima u RDF formatu. SPARQL je upitni jezik za postavljanje upita nad podacima u RDF formatu (slican je SQL-u).

Poslovni procesi

Poslovni proces je skup aktivnosti koje jedan ili vise ulaza transformisu u jedan ili vise izlaza radi ostvarivanja planiranog cilja. Ulazi I izlazi mogu biti informacije (dokumenti). Aktivnosti u poslovnom procesu su atomicki koraci poslovnog procesa na posmatranom nivou apstrakcije.

Dokument je jedinicni nosilac informacija izmedju elemenata informacionog sistema ili informacionog sistema I okoline.

Sistemi za upravljanje radnim tokovima su sistemi koji omogucavaju definisanje, izvrsavanje I nadgledanje izvrsavanja radnih tokova. Nazivaju se I sistemi za upravljanje radnim tokovima bazirani na objektima.

SOA (Service Oriented Architecture)

Pojam softverske arhitekture

Osnovna organizacija sistema, predstavljena komponentama, njihovim vezama sa drugim komponentama I okruzenjem I principima koji definisu njihov dizajn I evoluciju.

Skup osnovnih odluka o softverskom resenju koje ispunjava zadate parametre kvaliteta. Obuhvata samo osnovne komponente, njihove osnovne atribute I nacin saradnje. Izrazava se na razlicitim nivoima apstrakcije, zavisno od velicine projekta. Opisuje se iz vise perspektiva.

Arhitektura se definise rano, skup najranijih odluka u dizajnu, najteze za kasnije izmene, najvaznije da se dobro odrede. Prvi rezultat dizajna gde se vodi racuna o parametrima kvaliteta.

SOA nije neka konkretna tehnologija, nije novo ime za EAI, nije novi nacin da se radi RPC, nije novo resenje za software reuse.

Web servisi - procedure/metode dostupne putem HTTP-a, lose odabrano ime za ove tehnologije.

Kada se govori o SOA cesto se misli na WS standarde, gomila web servisa nije SOA.

EAI enterprise application integration

Alati za povezivanje razlicitih aplikacija u okviru organizacije, npr. Upravljanje zalihama, odnos sa klijentima, istrazivanje podataka, ljudski resursi. Obezbedjuje konzistentne podatke u razlicitim aplikacijama.

Vendor lock-in – vezivanje ja jednog proizvodjaca softvera

SOA nije EAI

Mane EAI – EAI je data centric, a ne process-centric, ne moze da isprati promene u poslovnim procesima, ne bavi se poslovnim procesima, vrlo slozena tehnicka resenja, retki/skupi kadrovi. Web servisi se mogu koristiti kao sredstvo za EAI, izbegava se vendor lock-in

SOA – easy reause

OO iskustvo, reuse je komplikovanije sto su komponente koje pokusavamo da iskoristimo vece. Servis implementira neku poslovnu funkciju/uslugu, moze se iskoristiti (use), moze se integrisati u veci proces, tesko se moze ponovno iskoristiti (reuse) za implementaciju drugog servisa.

Fokus je na agilnosti, mogucnost brze promene/adaptacije a ne na koriscenju originalnog servisa izvan originalnog konteksta.

Dva vidjenja SOA

Iz poslovne perspektive : Service Oriented Architecture

Analiza poslovanja – identifikacija poslovnih procesa, implemetacija servisa, komunikacija sa servisom putem poruka, koreografija – kako se obratiti servisu, kako ga iskoristiti u nekom kontekstu, orkestracija – kako implementirati servis pomocu poznavanja drugih servisa. Poenta je obezbediti agilnost u poslovanju – softverska podrska mora biti lako izmenjiva.

Iz tehnicke perspektive: softverska arhitektura bazirana na komponentama koje su slabo spregnute, interoperabilne, jednostavne za kombinovanje. Komponente/servisi imaju jasno definisane interfejse. Sistemi se formiraju od komponenti/servisa koje su krupne, autonomne, dostupne na adresama koje se mogu otkriti, komuniciraju putem poruka.

SOA iz poslovne perspektive:

- Analiza poslovanja
- Ustanove se krupne poslovne funkcije
- Krupne funkcije se mapiraju na softverske servise/komponente

Koncepti SOA

Servis – sredstvo koje ispunjava neki zahtev, jasna I jedinstvena funkcija, visoka kohezija, krupno parce poslovne logike, autonoman rad, samodovoljan, donekle I samoisceljujuci.

Ugovor – skup poruka koje poznaje servis. Jednostrani: servis sam definise svoje poruke, dvo/visestrani: poruke se definisu za dva li vise servisa u kombinaciji. Analogno OO pojmu interfejsa.

Endpoint – adresa (URI) na kojoj je servis dostupan, razliciti ugovori jednog servisa mogu biti na razlicitim adresama.

Poruka – jedinica komunikacije sa servisom, razni tehnoloski oblici: HTTP GET, SOAP, SMTP...Razlikuje se od RPC, ima header (za infrastrukturne servise) I body.

Politika – uslovi kada je servis dostupan. Specificira dinamicke osobine servisa (kada I za koga je dostupan), moze da se azurira run-time, odvojena od poslovne logike, ono sto razlikuje servis od OO objekata/komponenti.

Korisnik servisa – bilo koji softver koji komunicira sa servisom putem razmene poruka, klijent aplikacije, drugi servisi.

Koncepti SOA

4 koncepta se bave interfejsom: poruka, ugovor, endpoint, policy.

U OO paradigmi samo jedan, fokus na interfejs omogucava kreiranje komponenti koje su slabo povezane i jednostavne za kombinovanje. Isti koncepti koriste se u oba vidjenja SOA, jednostavnija konvergencija poslovne I tehnicke perspektive.

SOA funkcionise u distribuiranom okruzenju

Poruke se salju ponovo, poruke mogu da stignu vise puta, vazna karakteristika poruka: idempotentnost. Proizvodi isti rezultat I kad se primeni vise puta.

Modeli komunikacije

Zahtev/odgovor - klasicna klijent/server komunikacija

Zahtev/reakcija - odgovor na zahtev se obradjuje dugo, salje asinhrono.

RESTful

Klasicni web servisi: standardi

- SOAP za formatiranje poruka, WSDL za opis servisa
- XML Schema za tipove podataka
- UDDI za pronalazenje servisa

Klasicni web servisi: transport

SOAP ne zavisi od konkretnog transportnog protokola, inicijalno predvidjan HTTP, kasnije napravljeni bindings za SMTP, FTP, JMS...

Klasicni web servisi: vrste servisa

Stil: RPC I document

Prenos podataka, encoded I literal. Definise se u WSDL fajlu.

Klasicni web servisi: problemi

Ne postuje URI (Uniform Resource Identifier) koncept, putem URI-ja su dostupne SOAP operacije, a ne resursi, samo endpointi imaju URI. Nemoguce hesiranje HTTP saobracaja, iz jednog URI-ja nalaze se razlicite operacije.

REST: REpresentational State Transfer

Definise principe softverske arhitekture za web, alternativa za razvoj web servisa u odnosu na standardni SOAP+WSDL+..

REST principi

- Jednostavne operacije nad resursima kao HTTP metode:
    - GET – citanje
    - POST – kreiranje
    - PUT – azuriranje
    - DELETE – brisanje

Representational state transfer?

Klijent se obraca resursu putem URI-ja, dobija reprezentaciju resursa, ta reprezentacija pomera klijenta u novo stanje, klijent se zatim obraca drugom resursu.. Seoba klijenata iz stanje u stanje = transfer.

Motiv za razvoj REST-a, definisanje dizajn sablona koji opisuje kako bi web trebalo da radi, tako da predstavlja okvir za razne web standarde I dizajn web servisa.

Resurs – anotirana POJO klasa

Operacije – anotirane metode u resurs klasi. Ista operacija moze primiti/vratiti podatke u razlicitim formatima.

Karakteristike operacija

PUT I DELETE – idempotentne, vise identicnih zahteva daje isti rezultat

GET – bezbedna (safe metod), samo za citanje, ne sme da menja stanje na serveru.

RESTful servisi su statless – stanje je isljucivo na klijentu, transakcije su u nadleznosti klijenta

WSDL

WSDL – Web Services Description Language

XML gramatika za opsivanje web servisa kao skupa krajnjih pristupnih tacaka (access endpoints) koji mogu da razmenjuju poruke na RPC ili document-style nacin.

Acces endpoint = URL na koji se salje zahtev

WSDL – CORBA IDL : oba standarda su namenjena definisanju interfejsa I tipova podataka za servise dostupne sa udaljenih klijenata. WSDL obezbedjuje prosirivost koju CORBA IDL nema, opsivanje krajnjih pristupnih tacaka (endpoint) i poruka bez obzira na korisceni mrezni protokol ili format poruka za razmenu, tretman poruka kao apstrakthnih opisa podataka koji se razmenjuju, tretman tipova portova kao apstraktnih kolekcija operacija web servisa.

WSDL fajl opisuje sta servis radi, kako pozivati njegove operacije, gde ga pronaci. Na osnovu WSDL specifikacije servisa moguce je generisati implementaciju servisa (delimicnu) I klijentske klase za pristup servisu.

Na osnovu datih interfejsa u implementaciji servisa moguce je generisati WSDL fajl koji opisuje dati servis.

Struktura WSDL dokumenta

- wsdl:definitions – korenski element WSDL dokumenta, cesto se koristi za globalne namespace deklaracije.
- wsdl:import – dodaje sadrzaj datog namespace-a u tekuci WSDL dokument, nacin za modularizaciju WSDL dokumenta, dodatni namespace definisan je u datom fajlu – preporucuje se navodjenje apsolutne URL putanje zbog prenosivosti. Import tipicno sadrzi zajednicke XML Schema definicije tipova.
- wsdl:types – kontejner za definicije tipova podataka koji se koriste dalje u dokumentu, kao jezik za definisanje tipova podataka najcesce se koristi W3C XML Schema. WSDL ne ogranicava izbor jezika za definisanje tipova podataka, moguce je koristiti I druge jezike osim W3C XML Schema. Ugradjeni W3C XML Schema tipovi podataka ne moraju se eksplicitno importovati
- wsdl:message – za definisanje poruka koje se razmenjuju u komunikaciju sa web servisom, poruka se sastoji iz delova, a svaki deo pripada nekom tipu podataka
- wsdl:portType – definise skup operacija koje se mogu izvrsiti nad web servisom. Operacija kao element ovog skupa moze sadrzati ulaznu poruku, izlaznu poruku.
- wsdl:binding - definise konkretan protokol I format podataka za port type. Moze se koristiti standardan protokol (HHTP, SOAP, MIME,..) ili definisati nov. PortType predstavlja apstraktnu definiciju operacija. Pomocu binding elementa ove operacije se konkretizuju u skladu sa izabranim protokolom. za svaku operaciju navedenu u portType sekciji se mora navesti odgovaraju stavka u binding sekciji.
- wsdl:service - u prethodnim elementima nigde nije navedeno na kojoj URL adresi se nalazi web servis. Element service nije obavezan, koristi se kada je potrebno definisati konkretan endpoint za web servis. Servis je skup

portova, svaki port predstavlja po jedan endpoint u komunikaciji, moguce je definisati servis koji se sastoji od portova koji su dostupni na razlicitim adresama.

WSDL definicija predstavlja metapodatke o programskom kodu.

Varijane u dizajnu web servisa: RPC/encoded, RPC/literal, document/encoded, document/ literal...