

Funkcije i tehnologije, relevantne za izradu, primenu i korišćenje poslovnog softvera

U ovom poglavlju, predstavljene su funkcije i tehnologije, relevantne za izradu, primenu i korišćenje poslovnog softvera.

U prvom delu, definisani su informacioni sistemi, odnosno vrste i komponente informacionih sistema. U drugom, predstavljen je način na koji se uobičajeno informacioni sistemi koriste u poslovanju, odnosno, definisan je poslovni softver. U trećem delu, opisane su funkcije, arhitektura i način korišćenja dve izabrane vrste poslovnog softvera: sistema za upravljanje odnosima sa klijentima (CRM sistema) i sistema za upravljanje sadržajem na Internetu (CMS sistema). Cilj ovog dela je da se demonstriraju dve velike prednosti poslovnog softvera, proistekle iz koristi stečenih masovnom obradom velikog broja podataka o poslovanju i koristi stečenih otvaranjem poslovnog softvera prema Internetu, odnosno deljenjem podataka i interakciji u njihovom stvaranju i obradi. Poglavlje se završava opisom tehnologija i pristupa u izradi, implementaciji i integraciji poslovnih softvera.

Uvod u informacione sisteme

Informacioni sistem je integrisani skup komponenti za sakupljanje, snimanje, čuvanje, obradu i prenošenje informacija.

Poslovna preduzeća, druge vrste organizacija i pojedinci u savremenom društvu, zavise od informacionih sistema za upravljanje svojim operacijama i aktivnostima, održavanje konkurentnosti na tržištu, ponudu različitih usluga i unapređivanje kompetencija i kapaciteta. Na primer, moderne korporacije zavise od računarskih informacionih sistema koji su neophodni za obradu njihovih finansijskih i poslovnih transakcija, upravljanja ljudskim resursima; opštinske uprave zavise od informacionih sistema bez kojih se ne mogu pružati osnovne usluge njihovim građanima na efikasan i transparentan način; pojedinci koriste informacione sisteme da bi unapređivali svoja znanja, za kupovinu, upravljanje bankovnim računima i transakcijama, kao i za različite finansijske operacije.

Izumom i dostupnošću novih informacionih tehnologija (IT), javljaju se nove mogućnosti. Pošto su informacioni sistemi omogućili različite ljudske aktivnosti, samim tim su izvršili i veliki uticaj na društvo, odnosno izmenili životne stilove. Ubrzali su obavljanje svakodnevnih aktivnosti, uticali na strukturu organizacija, izmenili načine ponude i potražnje proizvoda na tržištu, kao i načine i shvatanje rada. Informacije i znanje, danas čine vitalni ekonomski resurs.

Komponente informacionih sistema

Osnovne komponente informacionih sistema su hardver i softver računara, baze podataka, telekomunikacioni sistemi i tehnologije, ljudski resursi i procedure, odnosno metodologije obrade i prenošenja informacija.

Hardver. Danas, čak i najmanja firma, posjeduje ili iznajmljuje računare. Obično se radi o personalnim računarima (PC). Veće organizacije koriste više računarskih sistema, počev od moćnih radnih stanica, miniračunara pa do efikasno umreženih personalnih računara. Zajedno sa perifernim uređajima, kao što su ulazni i izlazni uređaji (monitori i štampači, itd.), i telekomunikacije čine hardver informacionih sistema. U novije vrijeme, cene hardvera ubrzano opadaju, ali se brzina obrade i kapacitet uređaja za čuvanje podataka, konstantno uvećava.

Softver. Softver se obično deli na dve šire klase: sistemski ili operativni softver i aplikativni softver. Osnovni sistemski softver se obično naziva operativnim sistemom. Operativni sistem upravlja hardverom, datotekama i drugim resursima sistema, obezbeđujući sistematsko, konzistentno obavljanje zadataka i kontrolu računara, najčešće preko grafičkog korisničkog interfejsa. Aplikativni softver obuhvata programe dizajnirane za obavljanje specijalizovanih zadataka od kojih se mnogi nude na tržištu kao paketi pripremljeni za korišćenje odmah nakon instaliranja.

Baze podataka. Ogromna većina informacionih sistema se primarno oslanja na baze podataka. Baza podataka je kolekcija međusobno povezanih podataka, organizovanih na način koji omogućava najefikasnije korišćenje i ekstrakciju po zadatim kriterijumima. Tipičan primer baza podataka u jednom preduzeću su podaci o radnicima i katalozi proizvoda. Naročitu vrednost karakterišu baze o kupcima, koje se koriste za dizajn i marketing novih proizvoda. Bilo ko je ikada kupio nešto sa kreditnom karticom, direktno u prodavnici, preko elektronske pošte, ili na Internetu, postaje deo neke od mnogobrojnih baza o kupcima.

Telekomunikacije. Telekomunikacije se koriste za povezivanje, ili umrežavanje računarskih sistema, kao i za prenošenje informacija. Moguće je podesiti više različitih računarskih konfiguracija, zavisno od potreba svake organizacije. Lokalne mreže (LAN) povezuju računare određenog mesta, kao što su poslovne zgrade ili akademski kampus. Mreže širokog opsega (WAN) povezuju računare koji se nalaze na različitim mjestima i često iz različitih organizacija. Internet je mreža koja povezuje ogroman broj računara sa svih kontinenta. Zahvaljujući umrežavanju, korisnici personalnih računara dobijaju pristup različitim resursima i informacijama, velikim bazama podataka i ljudskim resursima, mogućnost obavljanja istih poslova u saradnji sa kolegama, kao i kontakt sa ljudima koji dele ista profesionalna i privatna interesovanja.

Ljudski resursi i procedure. Kvalifikovani stručnjaci su vitalna komponenta svakog informacionog sistema. Tehničko osoblje se sastoji od razvojnih i menadžera operacija, analitičara i projekatara sistema, programera i administratora sistema. Takođe, radnici u jednoj organizaciji moraju biti osposobljeni za efikasno korišćenje kapaciteta informacionog sistema. Svi ljudi svesno ili nesvesno uče da koriste informacione sisteme koristeći Internet. Procedure za korišćenje, upravljanje i održavanje informacionih sistema su deo njihove dokumentacije. Na primer, neophodno je definisati procedure za pokretanje programa za isplatu ličnih dohodaka, uključujući procedure za trenutak pokretanja, ko ima neophodnu dozvolu da ga pokrene i ko će imati pristup izlaznim informacijama.

Vrste informacionih sistema

Informacioni sistemi nude operacionu podršku, implementiranje znanja i podršku menadžmentu u raznim organizacijama i preduzećima.

Operaciona podrška. Na operacionom nivou nalaze se sistemi za obradu transakcija preko kojih se proizvodi projektuju, prave marketinške strategije, kontrolišu proces proizvodnje i šalje gotov proizvod na tržište. Ti sistemi akumuliraju informacije u baze koje su osnova sistema višeg nivoa. U današnjim naprednim organizacijama i preduzećima, informacioni sistemi koji podržavaju više funkcionalnih jedinica (marketing, finansije, proizvodnju i ljudske resurse), integrisani su u tzv. sisteme za planiranje i upravljanje resursima preduzeća (eng. Enterprise resource planning - ERP). Sistemi za planiranje i upravljanje resursima preduzeća podržavaju kompletnu sekvencu aktivnosti preko kojih preduzeće može dodati vrednost na svoja dobra i usluge. Na primer, pojedinac ili preduzeće, mogu poslati personalizovanu narudžbu preko interneta, koja će automatski pokrenuti proizvodnju u skladu sa specifikacijama poručioaca, preko sistema koji se zove sistem za masovnu personalizaciju proizvoda i usluga. Narudžbina se šalje u magacine

preduzeća, u slučaju potrebe i dobavljačima materijala neophodnih za proizvodnju. Na kraju, ažuriraju se računi i započinje novčana transakcija u skladu sa vrednošću proizvoda. Mnogi sistemi za obradu transakcija podržavaju elektronsko poslovanje preko Interneta. Između ostalih, to su online sistemi za kupovinu, online banking i sistemi koji osiguravaju bezbednost transakcija. Drugi sistemi nude informacije, usluge obrazovanja i zabave po zahtevu.

Podrška za unapređivanje znanja. Veliki deo posla u informacionom društvu, sastoji se, pored direktne obrade egzaktnih informacija i od manipulacije apstraktnih delova informacija i znanja. Postoje tri opšte kategorije informacionih sistema koji podržavaju rad na unapređivanju znanja: sistemi za profesionalnu podršku, kancelarijski informacioni sistemi i sistemi za upravljanje znanjem.

- **Sistemi za profesionalnu podršku.** Sistemi za profesionalnu podršku omogućavaju obavljanje poslova specifičnih za određenu profesiju. Na primer, mašinski inženjeri koriste CAD softver zajedno sa sistemima za virtuelnu realnost, za projektovanje i testiranje novih modela različitih mašinskih delova, sistema za zaštitu putnika u vozilu, itd., pre pravljenja prototipa. Kasnije, koriste CAE za projektovanje i analizu rezultata realnih testova. Biohemičari koriste specijalni softver za trodimenzionalno modeliranje i predstavljanje molekularnih struktura i analiziraju moguće efekte novih lekova, pre njihovog kliničkog istraživanja. U bankama se koristi finansijski softver za izračunavanje dobiti i analizu potencijalnih rizika različitih finansijskih ulaganja i strategija. Danas, specijalizovani profesionalni sistemi za podršku su prisutni u skoro svim profesijama.
- **Kancelarijski informacioni sistemi.** Glavni zadatak kancelarijskih informacionih sistema je olakšavanje komunikacije i saradnje svih članovima jedne organizacije, ali i sa članovima drugih organizacija. Arhiviranje dokumenata i pošte u elektronskom formatu olakšava organizacijama njihovu klasifikaciju, indeksiranje i čuvanje, kao i brz i efikasan pristup po potrebi.
- **Sistemi za upravljanje znanjem.** Sistemi za upravljanje znanjem se sastoje od metoda sakupljanja znanja, odnosno njegovog organizovanja u celinu, kao i procedura i metodologija korišćenja akumuliranog znanja u organizacijama. Ova vrsta znanja se obično sastoji od pisane dokumentacije i slika koje sadrže patenti, metoda dizajna, zbiru efikasnih poslovnih strategija, različitih studija i analiza kompetencije i sličnih resursa. Baza znanja jedne organizacije nije eksplicitnog karaktera, tako da ti sistemi pomažu korisnicima da stignu do onih članova organizacije čiji stručni profil najbolje odgovara njihovom upitu. Pristup znanju jedne organizacije se obično realizuje preko lokalne mreže (Intranet, ili mreže zatvorenog tipa), koje su snadbevene specijalizovanim softverom za pretragu.

Sistemi za podršku menadžmentu. Postoji široka kategorija informacionih sistema koji su dizajnirani za podršku menadžmentu u jednoj organizaciji. Ti sistemi se baziraju na podacima sistema koji obrađuju transakcije organizacije, podacima koji dolaze izvan organizacije i podacima koji stižu od poslovnih partnera, provajdera i klijenata. Informacioni sistemi generalno podržavaju sve nivoe menadžmenta, počev od onih zaduženih za planiranje i nadgledanje budžeta i troškova malih radnih grupa kratkoročno, pa do onih za planiranje i nadgledanje budžeta i troškova planova i strategija dugoročno gledano, koji se obično tiču cele organizacije. Sistemi za kreiranje izveštaja nude detaljne i obilne informacije u obliku izveštaja, predstavljenih i organizovanih na najpogodniji način, za svaku oblast odgovornosti menadžera. Generalno, ti izveštaji su fokusirani na prošla i sadašnja događanja u organizaciji. Radi prevencije zatrpavanja informacijama, izveštaji se automatski šalju samo u izuzetnim slučajevima, ili na konkretan zahtev menadžera.

Uvod u poslovne informacione sisteme (Enterprise Information System)

Poslovni informacioni sistem je bilo koji računarski sistem, sposoban da manipuliše velikim brojem podataka, koji se primenjuje u preduzeću. Poslovni informacioni sistemi su tehnološke platforme koje omogućavaju da preduzeća integrišu i koordinišu svoje poslovne procese. U jednom preduzeću se instalira i koristi jedan poslovni informacioni sistem, koji centralizuje obradu podataka, ali i obezbeđuje deljenje informacija kroz sve funkcionalne nivoe preduzeća i hijerarhije upravljanja. Implementacija centralnog poslovnog informacionog sistema podrazumeva kreiranje i korišćenje standardne strukture podataka i time uklanja probleme u poslovanju proistekle iz tzv. fragmentacije podataka.

Poslovni softver (Enterprise Software)

Poslovni softver predstavlja softver koji se koristi na nivou celog preduzeća. On realizuje egzaktni model celog preduzeća i predstavlja informatičko jezgro za podršku komunikaciji i kolaboraciji unutar njega. S obzirom na njegovu cenu i složenost, uglavnom ga grade i/ili implementiraju srednja i velika preduzeća.

Sva preduzeća karakteriše slična poslovna organizacija na funkcionalnom nivou, kao i slični procesi. Ovo je naročito izraženo u sektorima kod kojih postoje referentni modeli procesa i podataka. Referentni modeli procesa i podataka predstavljaju terminološke i sadržajne smernice zahvaljujući kojima se pospešuje kolaboracija između različitih preduzeća, ali i između različitih organizacionih celina jednog preduzeća. Primeri referentnih modela su SCOR i RosettaNet referentni modeli procesa u upravljanju mrežama snabdevanja, UNSPSC i eClass klasifikacija proizvoda, itd. Bez obzira na ove sličnosti i postojeće opšte smernice za poslovanje, ne postoje dva identična preduzeća. Zato, poslovni softver mora da karakteriše postojanje razvojnih standarda, jezika i alata, čijom primenom se on može prilagoditi okruženju u kojem se implementira. Ovi razvojni alati podrazumevaju korišćenje nekog programskog jezika i zato ih koriste isključivo programeri. Oni se razlikuju od jednog do drugog proizvođača poslovnog softvera, od kojih svaki koristi svoju ili otvorenu tehnologiju, ali i više njih, istovremeno (portabilnost). Glavni proizvođači poslovnog softvera su kompanije IBM, HP, SAP, Microsoft, Sun Microsystems, Adobe Systems, Oracle Corporation i drugi.

Osnovne karakteristike poslovnog softvera su: performanse, proširivost (skalabilnost) i pouzdanost (robustnost). U poslednje vreme, naglašava se važnost i interoperabilnosti kao karakteristike poslovnog softvera, odnosno, njegove sposobnosti da se integriše sa drugim poslovnim softverima, ili aplikacijama.

Uobičajeno, poslovni softver se razvrstava prema poslovnim funkcijama koje podržava ili prema sektoru industrije u kojoj se primenjuje. Prema poslovnim funkcijama, poslovni softver može da bude softver za prodaju, računovodstvo, upravljanje ljudskim resursima, planiranje poslovnih resursa, upravljanje sadržajem na webu, obradu plaćanja, proizvodnju, itd. Prema sektoru, poslovni softver može da bude specijalizovan za farmaceutsku, mašinsku, duvansku, ili bilo koju drugu industriju.

Poslovni softver se u jedno preduzeće implementira na različite načine. Njega može da razvije i implementira lokalno IT odeljenje, ili može biti nabavljen (zajedno sa podrškom instaliranja, tranzicije podataka, obuke, itd.) od proizvođača poslovnog softvera. Konačno, sa razvojem interneta kao medija i za poslovanje, pojavljuje se i novi model korišćenja poslovnog softvera – softver kao servis. Naime, postoje proizvođači poslovnog softvera koji svoje proizvode nude po principu usluge, pri čemu se naplaćuje složenost i količina podataka koja se obrađuje, broj korisnika, vreme korišćenja, itd.

U svakom slučaju, poslovni softver se instalira i održava centralno, tj. smešta se na specijalizovanim server računarima, prilagođenim za istovremeni rad većeg broja korisnika, velikim brojem računskih operacija, sa pouzdanim sistemom za skladištenje i backup podataka.

Primeri poslovnih softvera

U ovom poglavlju, opisan je rad dve kategorije poslovnog softvera: sistema za upravljanje klijentima (Customer Relationship Management sistemi - CRM) i sistema za upravljanje sadržajem na Internetu (Content Management Sistemi - CMS). Ovo poglavlje ima za cilj da demonstrira funkcije i način korišćenja poslovnog softvera, kao i da demonstrira mogućnosti za unapređenje poslovanja, proistekle iz dva aspekta:

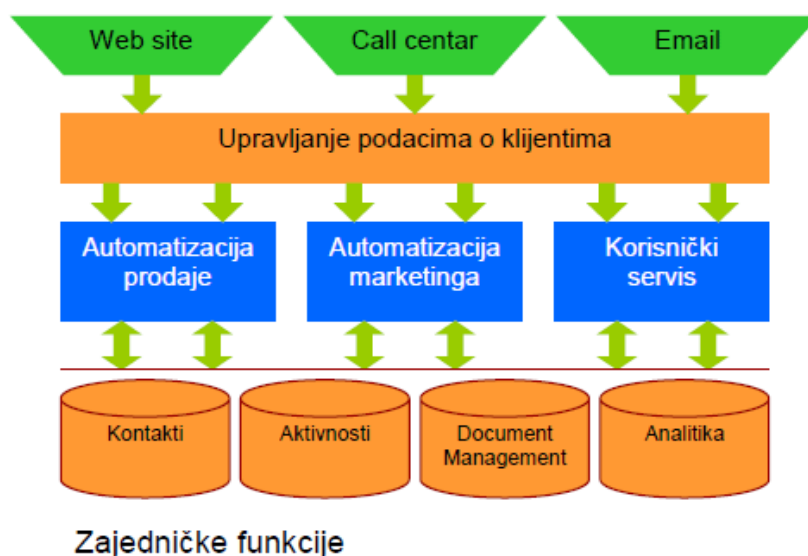
- Mogućnosti vezane za donošenje poslovnih zaključaka, proistekle iz kapaciteta za obradu velikog broja podataka (na primeru CRM sistema) i
- Mogućnosti kolaboracije proistekle iz integracije poslovnog softvera sa web tehnologijama (na primeru CMS sistema).

CRM sistemi (Customer Relationship Management - CRM)

CRM je industrijski termin koji se odnosi na upravljanje odnosom preduzeća sa svojim klijentima na organizovani način. CRM se koristi za izbor i upravljanje najvrednijim relacijama koje se uspostavljaju između klijenata preduzeća i njegovih proizvoda, usluga i drugih poslovnih objekata.

Klijent je centar poslovanja oko kojeg orbitiraju sve ostale poslovne aktivnosti i resursi. U customer-centric poslovnoj filozofiji njega možemo nazvati i osnovnim sredstvom i delom vrednosti kapitala jednog preduzeća. Njegova lojalnost je najveća poslovna snaga preduzeća. Iako mnoge studije pokazuju da privući klijenta košta 5 puta više nego zadržati ga, istraživanje Harvard Business Review časopisa otkriva neverovatan podatak da neke kompanije mogu povećati profit za više od 100%, ukoliko povećaju procenat zadržavanja klijenata za samo 5%. Osnovna misija CRM strategije nije upravljanje odnosom sa klijentima, već upravljanje odnosom sa klijentima koji preduzeću najviše vrede. Uloga CRM-a je upravo u tome da otkrije tih 5% u koje treba ulagati vreme i resurse.

Osnovni elementi savremenog CRM sistema su Upravljanje podacima o klijentima (Account Management), Automatizacija prodaje (Sales Force Automation), Korisnički servis (Customer Service & Support) i Automatizacija marketinga (Marketing Automation). Kao pomoćni delovi sistema koje koriste sve službe tu su i Upravljanje kontaktima (Contact Management), Upravljanje aktivnostima (Task management), Upravljanje dokumentima (Document Management) i Analitika (Analytics).



Slika 1. Arhitektura savremenog CRM sistema

Predmet sistema za automatizaciju prodaje (Sales Force Automation) je upravljanje Sales Pipeline procesom. On počinje identifikovanjem ciljnog tržišta (klijenata) koje preduzeće želi da osvaja i završava narudžbinama odnosno isporukom proizvoda ili usluga. U međuvremenu, sektor prodaje obrađuje prodajne mogućnosti (sales leads) koje generiše služba marketinga i, kroz komunikaciju sa potencijalnim klijentom ih evoluiru u prodajne prilike (sales opportunities) i konačno, narudžbine. Kroz ovaj proces, CRM sistem treba da omogući sektoru prodaje da beleži ostvarene i buduće kontakte sa klijentom, predviđene i ostvarene aktivnosti i ostale podatke koji mogu poslužiti za bržu i lakšu evoluciju prodajnih mogućnosti u narudžbine.



Slika 2. Sales Pipeline proces

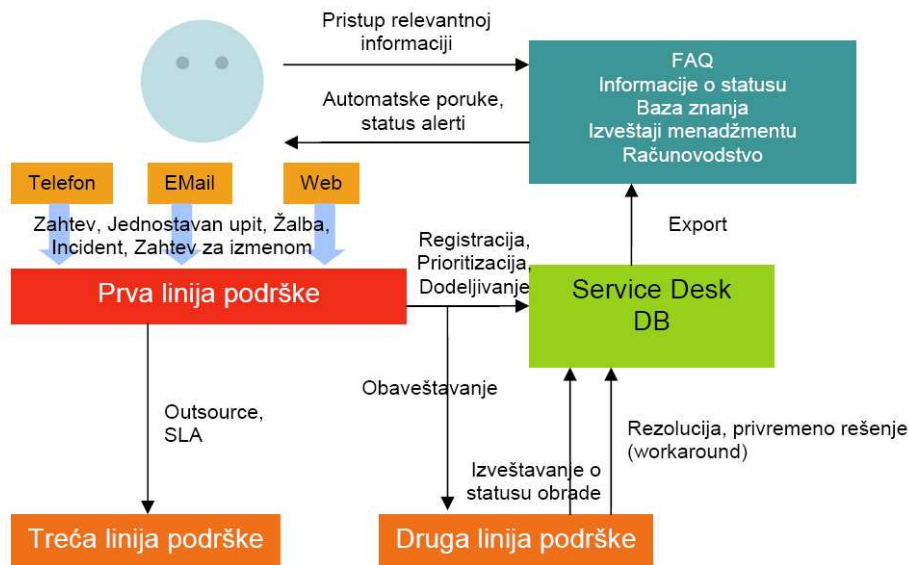
Disciplinovanje prodaje kroz Sales Pipeline stvara uslove za primenu metrike koja pomaže da se poslovanje jednog preduzeća bolje razume, lako uoče rizici i problemi u prodaji, predvide se i poboljša proces u najslabijoj kariki sistema. Ova metrika može da da odgovore na pitanja, kao što su "Koliko ste puta morali da ostvarite kontakte i sa kim da biste ugovorili sastanak ili prezentaciju? Koliko sastanaka ste morali da održite da biste dobili posao?". U jednoj analizi ovih podataka može se zaključiti da je npr. u proseku bilo potrebno 10 poziva da bi se zakazala poslovna prezentacija, odnosno 5 prezentacija da bi se uspešno zaključila prodaja. Dalje, može se podvući crta, pa zaključiti da je u toku fiskalne godine obavljeno 10.000 poziva iz kojih je proisteklo 200 ugovora sa ukupnim prihodom od 20.000 EU.

Funkcije sistema za automatizaciju prodaje su:

- Upravljanje prodajnim mogućnostima (Lead Management),
- Upravljanje prodajnim prilikama (Opportunity Management),
- Prognoze poslovanja i analitika prodaje (Global Forecasting, Sales Analytics),
- Teritorijalna organizacija prodaje (Territory Management),
- Automatizacija tokova (Workflow automation),
- Obaveštavanje u realnom vremenu (Real time alerts, Update reminders, Big Deal Alerts),
- Upravljanje podacima o klijentima (Account management),
- Katalog proizvoda i usluga (Product Catalog),
- Održavanje uputstava i standarda (Sales Literature) i druge.

Funkcije CRM modula za automatizaciju marketinga su planiranje aktivnosti i budžeta za marketinške kampanje, obrada digitalnih kanala kroz koje pristižu informacije koje mogu prerasti u prodajne mogućnosti i analitičke aktivnosti na merenju učinka kampanja.

Konačno, korisnički servis je prva linija fronta prema korisnicima proizvoda i usluga. Ovaj segment CRM sistema može biti kritičan za njegovu ukupnu efikasnost, jer se kroz njega ostvaruje direktan kontakt sa klijentom, veoma često – nezadovoljnim. Brza i efikasna usluga i rešavanje problema povećava lojalnost klijenata i otvara vrata za nove narudžbine.



Slika 3. Šema poslovanja korisničkog servisa

Osnovne karakteristike korisničkog servisa su:

- **Jedinstveno mesto kontakta (SPOC – Single Point Of Contact).** Kontakt sa klijentima može da u ime preduzeća ostvari samo ovlašćeno lice. Na taj način se obezbeđuje da klijent uvek dobije preciznu, tačnu i ažurnu informaciju. Takođe, samo ovlašćeno i obučeno lice može da odluči ko će iz firme biti odgovoran za obradu nekog zahteva, ukoliko on nije sposoban za to. Konačno, jedinstveno mesto kontakta omogućava praćenje aktivnosti servis-deska, jer se svi oni beleže u jedinstvenu bazu podataka.
- **Servis desk je prva linija fronta zaštite ugovora.** Preduzeća koja se bave pružanjem usluga, sa svojim klijentima sklapaju tzv. Service Level Agreement (SLA) ugovore. U svakom od ovih ugovora je čvrsto definisan nivo obima i kvaliteta usluga, kao i rokovi njihovih pružanja. Vrlo verovatno su definisane i sankcije koje se moraju podneti, ukoliko se definisani nivo ne ostvari. Preduzeća koja plasiraju svoje proizvode imaju svoju garantnu politiku. Ovlašćeni predstavnik korisničkog servisa mora da u svakom registrovanom problemu nedvosmisleno prepozna kršenje SLA, rizik od kršenja SLA, odnosno usklađenost prijavljenog problema sa garantnom politikom. Na osnovu ovih zaključaka, vrši se prioritizacija obrade svakog kontakta.
- **Servis desk koordinira aktivnostima na obradi zahteva.** Iako je ovlašćeno lice neretko u situaciji da samostalno odgovori na zahtev klijenta, često se dešava da priroda kontakta zahteva angažovanje lica uže specijalnosti. Ponekad je, čak, potrebno i eksterno angažovanje profesionalca. Pošto nije racionalno da se prva linija kontakta sa klijentima opterećuje skupim ekspertima, neki od njih, pored redovnih obaveza imaju i zadatke da u određenim stepenima prioriteta pomažu korisničkom servisu. Na ovaj način, njegove aktivnosti se raslojavaju – prvu liniju podrške čine ovlašćena lica (jedinstveno mesto kontakta), drugu liniju podrške – određena odeljenja preduzeća koja sa servis deskom imaju tzv. OLA (Operational Level Agreement) interni ugovor (kojim se definiše npr., u kojim slučajevima je odeljenje obavezno da prekine tekući posao i prione na obradu zahteva nekog klijenta) i konačno, treću liniju – eksterni profesionalci i druge firme (ovlašćeni servisi).
- **Servis desk se stara da je klijent uvek obavešten o statusu obrade njegovog zahteva.** Ponekad, složenost obrade zahteva klijenta podrazumeva da on mora da prođe kroz nekoliko faza sa jedinstvenom ili podeljenom odgovornošću. Obaveštavanje klijenta

o statusu obrade, odnosno fazi u kojoj se njegov zahtev nalazi predstavlja vrhunski izraz poštovanja.

- **Servis desk koristi bazu podataka za beleženje svih zahteva klijenata.** Bez obzira na obim posla u prvoj liniji podrške, jedinstvena baza servisnih informacija je neophodan sastojak ovakvog koncepta njegovog funkcionisanja. Pored toga što sadrži registrovane kontakte, ona služi kao posrednik za komunikaciju između prvog, drugog i trećeg sloja podrške; kao svojevrsna baza znanja iz koje se može jednim klikom izvući rešenje prijavljenog problema; kao osnova za automatsko, online kreiranje izveštaja menadžmentu; kao izvor podataka za FAQ sekcije web sajta preduzeća, ali i računovodstveni sistem (ukoliko obrada nekog zahteva podrazumeva naplatu), itd. Ova baza predstavlja integralni deo jedinstvenog CRM sistema.

CRM omogućava preduzećima i njihovim klijentima da dobiju:

- **jedinstveno mesto pristupa svim informacijama o klijentu.** Svim informacijama o potencijalnim ili ostvarenim odnosima kompanije i njenog klijenta se može pristupiti sa jednog mesta. Na ovom mestu se mogu saznati direktne informacije o klijentu, ali i posredne – niz aktivnosti korisničkog servisa, ostvareni i zakazani kontakti sa njim, istorija kupovine itd.
- **informaciju u realnom (ili skoro realnom) vremenu.** Uz pomoć CRM sistema, radnici službe prodaje, marketinga ili korisničkog servisa, ali i sam klijent, mogu da dobiju tačnu i validnu informaciju o raspoloživosti nekog proizvoda, statusu neke narudžbine, itd.
- **moгуćnost da vrše procenu mogućnosti prodaje svojim klijentima.** Detaljne informacije o klijentima mogu poslužiti kao potpuna osnova za pretpostavke o njihovim potrebama i kupovnoj moći. Na osnovu ovih pretpostavki se može izvršiti kvalitativna segmentacija klijenata preduzeća i poslovni nastup po grupama prioriteta.
- **kvantitativno izraženi učinak službe prodaje, marketinga i korisničkog servisa.** Na osnovu procene transformacije marketinških kampanja u individualne prodajne mogućnosti, prodajne prilike i konačno – narudžbine, vrlo je lako najobičnijim brojem iskazati učinak neke službe i njenih poslovnih poteza.
- **analitičke izveštaje koji ukazuju na pravce poslovanja u budućnosti.** Analize i prognoze, proistekle iz CRM sistema, omogućavaju menadžmentu preduzeća da prepozna i promoviše praksu koja daje rezultate.

Sistemi za upravljanje sadržajem (Content Management Systems - CMS)

U praksi postoji mnogo proceduralnih rešenja za održavanje sadržaja nekog web sajta. Sva su se uspešno primenjivala mnogo pre pojave CMS-a sa zajedničkom misijom sjedinjenja napora ljudi različitih profesija i nivoa odgovornosti u cilju stvaranja kvalitetnog sadržaja koji se posredstvom web sajta prezentuje korisnicima. Sve ove procedure se zasnivaju na čvrstim, nekada vrlo komplikovanim, organizacionim formama kod kojih se težište nalazi na odgovorima na pitanja KO radi, ŠTA radi i KADA radi. S obzirom na to da su ova proceduralna rešenja nastajala u vreme kada je nivo upotrebljivosti razvojnih tehnologija bio daleko od današnjeg, na poslednje pitanje – KAKO radi, davani su veoma komplikovani odgovori. Iz ovog razloga, u životnom veku sadržaja za web, on je prolazio mnogo puta kroz ruke veoma stručnih lica, čiji rad jednu kompaniju košta mnogo.

Content Management je mlada disciplina koja je svoju osnovu nasledila od nekoliko drugih, kao što su Document Management, Information Design, News Publishing i Change Management, a evoluirala preslikavanjem određenih njihovih koncepata i poslovnih procesa na novi medij - web.

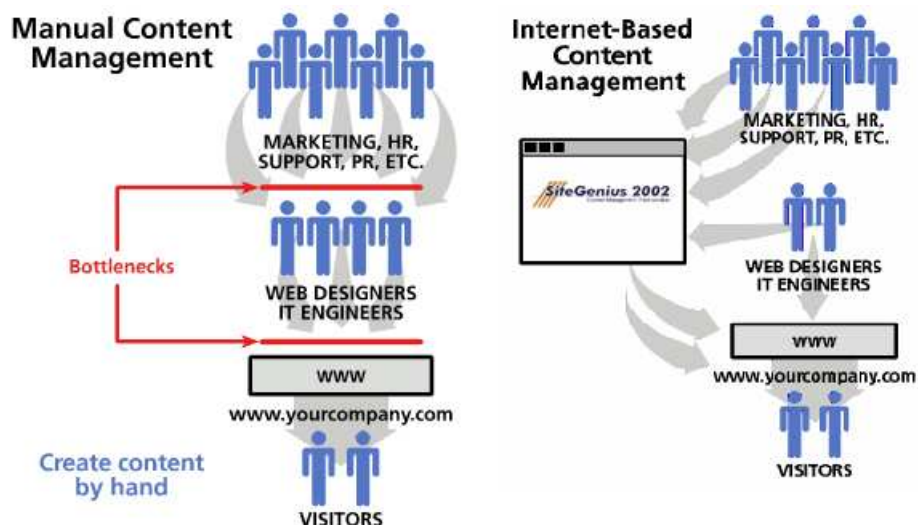
Content Management predstavlja skup procesa, tehnologija, koncepata i praktičnih rešenja za podršku razvoju, prikupljanju, održavanju i objavljivanju određenog sadržaja na web sajtu.

Pred CMS se postavlja niz zahteva koji se mogu sažeti u nekoliko tačaka:

- Sadržajem web sajta može upravljati više autora, urednika i drugih lica određenih specijalnosti sa različitim lokacija.
- Procedure za upravljanje sadržajem i kontrolu njegovog kvaliteta su upravljive - mogu se kreirati i modifikovati.
- Učesnici u procesu upravljanja sadržajem ne moraju da imaju nikakvo napredno tehničko znanje.
- Ukoliko se pojavi potreba, moguće je ostvariti re-kreiranje verzija web sajta u bilo kojoj tački u prošlosti.
- Sadržaj će biti prezentovan dinamički, imajući u vidu niz pravila po kojima se vrši njegova selekcija i prilagođavanje.
- Korisnicima web sajta se nude različiti nivoi personalizacije, pri čemu će administraciji biti dostupni ovi podaci sa analitičkim i statističkim proračunima.
- Postojeći sadržaj se distribuira u različitim oblicima, zavisno od tehnologije klijenta.

CMS sistemi su nastali kao odgovor na probleme, proistekle iz tzv. ručnog kreiranja sadržaja, njegovim prostim kodiranjem primenom HTML jezika. Čak, i ukoliko se web strane kreiraju primenom određenih alata za poboljšanje produktivnosti u kodiranju, kao što su FrontPage, GoLive ili Macromedia Dreamweaver, ovakav proces karakterišu visoki troškovi plaćanja HTML koda koji kreiraju strane i odgovarajuće linkove, nabavke alata za razvoj i specijalističke obuke, kao i veliki broj radnih sati potreban za proveru kvaliteta – linkova, strana, itd. Osnovne karakteristike procesa ručnog kreiranja web strana su:

- Mala inicijalna investicija
- Predvidljivi ali značajni periodični troškovi
- Za svaku stranu je potrebno 1-2 sata da se napravi i integriše
- Nema strukturane informacije
- Nema automatizacije
- Razvojni tim je "slepo crevo" u firmi
- Nepredviđena odsustva mogu zaustaviti proces
- Lako dolazi do grešaka



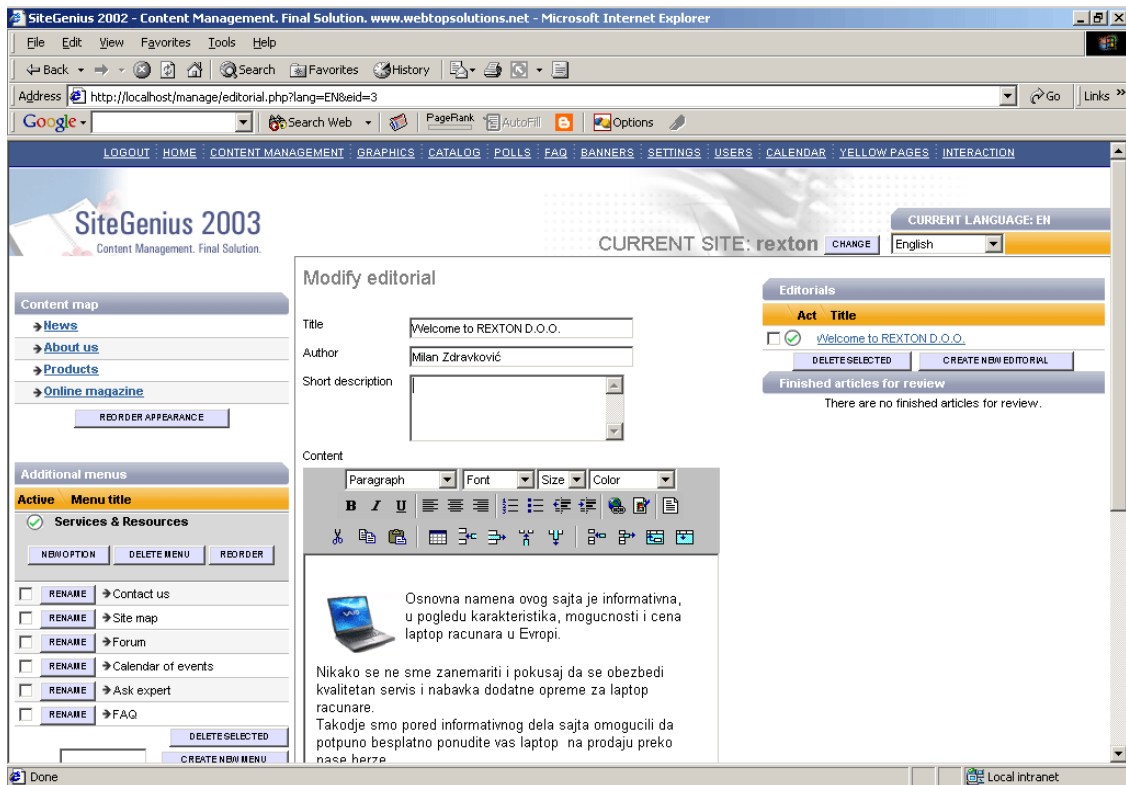
Slika 4. Šeme procesa ručnog kreiranja sadržaja i kreiranja sadržaja primenom CMS alata

Sa druge strane, primenom CMS alata u procesu kreiranja i objavljivanja sadržaja, dostižu se visoki standardi produktivnosti i kvaliteta, pri čemu se dugoročno troškovi značajno smanjuju. Naime, primenom ovih alata, "vlasnici" sadržaja postaju njegovi kreatori, moguća je podela posla, otvara se mogućnost planiranja procesa kreiranja i održavanja sadržaja (publishing workflow), informacije su dostupnije jer su strukturane, sadržaj je razdvojen od njegove vizuelne prezentacije, moguć je rad "iz daleka", i omogućen je brz početak rada, nakon implementacije procesa.

Osnovne funkcije sistema za upravljanje sadržajem su modeliranje sadržaja (Information Design), kreiranje sadržaja (Authoring), objavljivanje sadržaja (Publishing), upravljanje procesom kreiranja sadržaja (Workflow), održavanje životnog veka sadržaja (Versioning), integracija sa poslovnim informacionim sistemom (Integration) i prezentacija sadržaja (Presentation).

Modeliranje sadržaja. Da bi jedan sistem mogao da "razume" jednu informaciju, ona mora biti raščlanjena na jasnu, eksplicitnu strukturu. "Razumevanje" informacije znači da se ona može pretraživati parametarski, da se ona može upoređivati sa drugim, srodnim informacijama, da se ona može koristiti više puta u različitim kontekstima, da je sistemu poznato kako se menjala u vremenu i ko je menjao itd. Strukturiranje informacije je proces koji se sastoji od identifikovanja atributa njene strukture, jasno definisanih tipova i pravila za verifikaciju njihovih vrednosti i definisanju relacija između njih. Sam proces strukturiranja jedne informacije u velikoj meri određuje veličina koju je potrebno postaviti pre njegovog početka – nivo apstrakcije ili nivo detaljnosti. Ona predstavlja kompromis između truda potrebnog da se svim atributima sadržaja dodeljuje vrednost, tačna i ažurna, i predviđenih potreba samih korisnika informacije.

Kreiranje sadržaja. Osim u malom broju slučajeva, sadržaj za web kreiraju ljudi sa ograničenim iskustvom u korišćenju Internet tehnologija. Oni moraju da na raspolaganju imaju front-end alate koji će im omogućiti da svoj posao odrade na najbrži i najjednostavniji mogući način, uz kratku obuku ili bez nje. Čak i ukoliko sadržaj nastaje na desktop aplikacijama, authoring front-end treba da omogući njegovu validaciju i eventualnu tranziciju u oblik razumljiv za web čitače ili njegovo prilagođavanje definisanoj strukturi sadržaja.



Slika 5. Prikaz ekrana jednog sistema za upravljanje sadržajem na webu sa alatom za kreiranje sadržaja

Objavljivanje sadržaja – Publishing. Arhitektura jednog CMS-a se sastoji iz samo dva osnovna podsistema – “staging” okruženje u kojem se sadržaj kreira i menja i produkciono okruženje u kome se sadržaj prezentuje. Sve operacije pripreme, inicijacije i same tranzicije (deployment) iz jednog u drugo okruženje pripadaju ovoj fazi Content Management-a – objavljivanju sadržaja (publishing). Tranzicija sadržaja može biti fizička ili virtualna. Ova dva koncepta podsistema za objavljivanje sadržaja određuju dva moguća pravca razvoja CMS-a od kojih svaki ima svojih prednosti i mana. Fizička tranzicija podrazumeva generisanje i transfer statičkih HTML datoteka na produkcioni server. Velika prednost fizičke tranzicije u odnosu na druge oblike je u tome što se pred produkciono okruženje sistema ne postavljaju složeni zahtevi za instalacijom softvera (samo web server). Generisanje datoteka se može inicirati po zahtevu (on-demand), po zadovoljenju nekog uslova (event-driven) ili periodično - automatizovano, u određenim vremenskim intervalima (scheduled). Transfer generisanih datoteka se obavlja korišćenjem standardnih protokola (FTP, SSH, VPN). Virtualna tranzicija sadržaja se vrši u arhitekturi sistema kod kojih u produkcionom okruženju postoji aplikacioni server koji obrađuje zahteve i ostvaruje direktnu vezu sa centralnim repozitorijumom sadržaja. U slučaju virtualne tranzicije, ne vrši se fizički transfer sadržaja iz “staging” okruženja u produkciono, već su oni u stalnoj vezi.

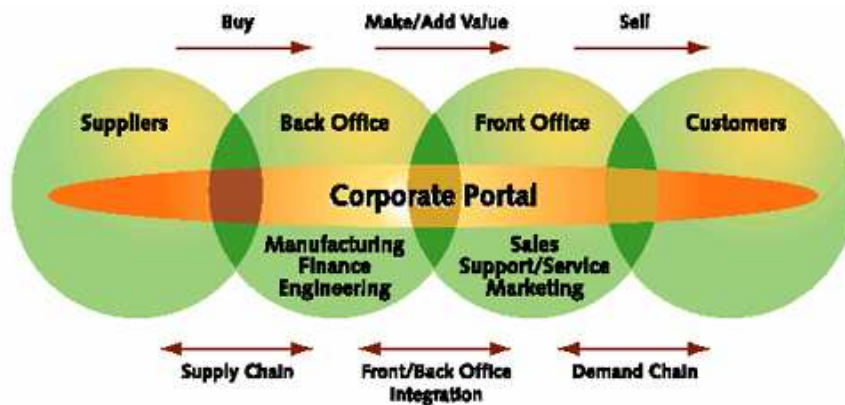
Upravljanje procesom kreiranja i održavanja verzija. Alat koji se koristi za upravljanje projektom kreiranja i održavanja sadržaja na web sajtu je sastavni deo CMS-a i naziva se Workflow alat. Ova aplikacija treba da omogući kreiranje tokova informacija, radnih naloga i upozorenja kroz proces nastajanja sadržaja, koji se odvija u više tačno određenih koraka i u kome učestvuje više poznatih aktera. Ovako definisan tok se naziva modelom procesa. Procesi u kojima sadržaj nastaje mogu biti različiti, u zavisnosti od potrebnog nivoa autorizacije, lokalizacije (prevoda), lektorizacije, definisanja ključnih reči, XML/HTML validacije, itd. Alat treba da omogući kreiranje i održavanje liste svih faza kroz koje može ili mora proći proces izrade sadržaja, zatim izbor pravila koja moraju biti zadovoljena da bi se izvršila tranzicija iz jedne u drugu fazu i definisanje odgovornosti aktera procesa.



Slika 6. Primer procesa kreiranja sadržaja

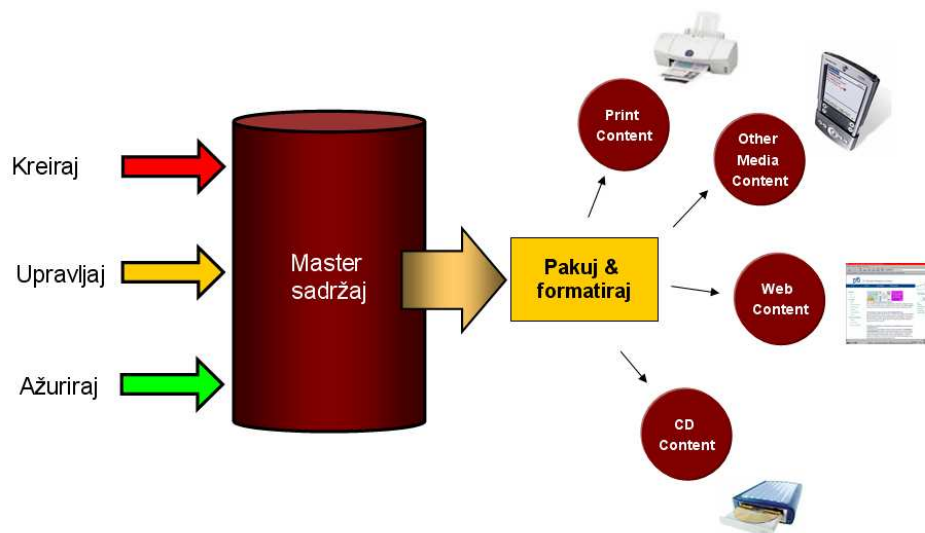
Održavanje životnog veka sadržaja. Svaki sadržaj na web sajtu ima i svoju vremensku dimenziju koja se može sagledavati sa stanovišta trenutka njegovog objavljivanja. Pre objavljivanja sadržaja, on prolazi kroz proces u kome mu određeni akteri iz organizacije dodaju vrednost kreiranjem, oblikovanjem i nadgradnjom informacije. Svaka od ovih operacija mora biti verifikovana od strane aktera koji je zadužen za kontrolu kvaliteta (urednik). Ukoliko njen rezultat nije zadovoljavajući, sadržaj se mora vratiti u prethodno stanje (rollback), da bi se operacija ponovila, od strane istog ili drugog aktera. Da bi rollback bio uopšte moguć, sistem mora da podržava skladištenje prethodnih verzija sadržaja. Nakon završetka mukotrpnog procesa kreiranja sadržaja, on prolazi kroz period adolescencije u kojem se uredništvo zasipa prvim reakcijama auditorijuma. Ovo je deo njegovog životnog veka u kome su izmene česte, jer na videlo izlaze neuočene greške, nejasni iskazi (koji se identifikuju eksplicitnim anketiranjem posetioca web sajta ili analizom pitanja postavljenih u njihovim mailovima), itd. Konačno, u onom trenutku u kome se sirova informacija, upakovana u sadržaj kroz proces njegovog nastajanja izmeni, sadržaj zastareva i neophodno je da se zameni novim. Svaka izmena sadržaja ne podrazumeva njegovu prostu zamenu već kreiranje njegove nove verzije. To što neki sadržaj predstavlja pojavni oblik neke zastarele informacije ne znači da je potpuno bezvredan. Ukoliko je strukturiran, on će biti itekako koristan za relativizaciju informacije predstavljene u novom sadržaju (naravno, samo u kontekstu koji je pozitivan za poslovanje organizacije). Još jedan od veoma važnih aspekata čuvanja prethodnih verzija sadržaja je i mogućnost kreiranja tzv. date-based snapshots web sajta kojima u pravnom procesu možete demantovati objavljivanje netačne informacije, za šta ste eventualno optuženi.

Integracija sa poslovnim informacionim sistemom. Izvor sirovih informacija oko kojih se sadržaj gradi je poslovni informacioni sistem organizacije koja se predstavlja na web-u. Ukoliko sistem za upravljanje sadržajem podržava da se njemu dodeli uloga aktera u procesu kreiranja sadržaja, otvara se mnogo prostora za automatizaciju određenih poslova, ali i brže objavljivanje ažurne informacije na web-u. Dve krajnosti stepena integracije su sinhrona, čvrsta i asinhrona, labava. Kod asinhrona, labave integracije, pretpostavlja se da CMS sadrži interfejs za uvoz datoteke sa podacima u obliku u kojem je generiše poslovni informacioni sistem (npr. CSV – Comma Separated Values datoteka). Sinhrona integracija uobičajeno podrazumeva korišćenje dve egzotične tehnologije, nedovoljno eksploatisane kod nas - data warehousing i web servisi. Data warehousing sistemi omogućavaju kreiranje tačno definisanog pogleda (datamart) na poslovanje firme, koji podrazumeva kalkulaciju različitih informacija iz različitih baza podataka u organizaciji. Ispred data warehousing sistema, treba da postoji interfejs od kojeg sistem za upravljanje sadržajem može u svakom trenutku da potražuje datamart, sa uvek ažurnim podacima i to, u odgovarajućem obliku u kojem se on može ravnopravno uključiti u proces kreiranja sadržaja. Danas, taj oblik je XML, a jezik kojim sistem za upravljanje sadržajem i interfejs data warehousing sistema komuniciraju – SOAP.



Slika 7. Korporativni portali kao sredstvo integracije poslovnih softvera na webu

Prezentacija sadržaja. Primena CMS-a promovise nekoliko važnih koncepata za prezentaciju sadržaja. Osnovni je nezavisnost sirove strukture sadržaja od njegovog pojavnog oblika – web strane. Naime, sistem treba da omogući da se elementi ove strukture, preslikavanjem u određeni template, rasporede na željeni način po strani. Ovaj template takođe treba da strukturu postavi u određeni kontekst, prikazivanjem linkova na drugi sadržaj koji je sa prikazanim u relaciji. Razdvojenost sadržaja od njegovog prikaza omogućava i da ga CMS distribuira u različitim oblicima sa stanovišta tehnologije prikaza (WAP). Mnogo ekskluzivniji koncept za prezentaciju sadržaja se ne odnosi na to KAKO će on biti prikazan, već KOME. Termin “personalizacija” se kao tehnologija prvi put pominje 1993. godine u knjizi “The One to One Future”, autora Don Peppers i Martha Rogers i to u kontekstu primene specifične strategije prodaje svakom klijentu, shodno njegovim individualnim osobenostima. Očigledno je da ova tehnologija otvara vrata automatizovanom direktnom marketingu. S obzirom na važnost personalizacije, u nekom od narednih brojeva će joj biti posvećen ceo članak.



Slika 8. Koncept razdvajanja sadržaja od prezentacije

Osnovne tehnologije za izradu, implementaciju i integraciju poslovnog softvera

U ovom delu, prikazane su osnovne tehnologije za izradu, implementaciju i integraciju poslovnog softvera. Detaljno je objašnjen koncept distribuiranih aplikacija, odnosno klijent – server arhitekture, koju koriste svih savremeni poslovni softveri, odnosno njegove varijante, koje su se razvile kao odgovor na razvoj tehnologija programiranja. Dalje, s obzirom na značaj Interneta, odnosno weba, kao medija za kolaboraciju i integraciju različitih poslovnih softvera, ali i medija za distribuciju poslovnog softvera kao servisa, detaljno je predstavljen način funkcionisanja web aplikacija. Dalje, predstavljen je osnovni standard za modeliranje strukture podataka, koji se koristi u svim savremenim poslovnim softverima – XML. Konačno, dat je uvod u funkcionisanje web servisa kao osnovnih tehnoloških sredstava za integraciju poslovnog softvera, danas.

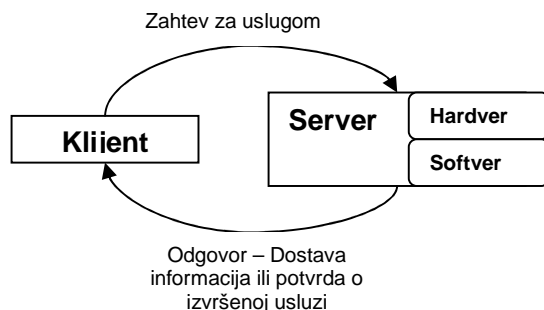
S obzirom na porast značaja kolaboracije i integracije ljudi i računara, odnosno poslovnog softvera, savremeni trendovi razvoja poslovnog softvera podrazumevaju njihov distribuirani dizajn, odnosno, distribuciju funkcija i podataka prema odgovornosti i funkciji određenog odeljenja preduzeća, ili različitih preduzeća. Ovakve aplikacije se nazivaju distribuirane aplikacije, a razvijaju se primenom neke od varijanti klijent – server arhitekture. Osnovna karakteristika distribuiranih aplikacija je korišćenje distribuiranih objekata. Distribuirani objekat je objekat koji se definiše na jednom sistemu a može se koristiti na drugom. U ovoj komunikaciji može učestvovati jedan ili više kompjutera. Cilj je da se obezbedi integracija servisa sa različitim platformi.

Sistem zasnovan na distribuiranim objektima je skup objekata koji razdvaja one koji zahtevaju usluge (klijente) od onih koji pružaju usluge (servera) uz pomoć dobro definisanog interfejsa. Drugim rečima, klijenti su odvojeni od implementacije servisa, npr. zapisa podataka i izvršnog koda. Ovo je jedna od najbitnijih razlika koja izdvaja model zasnovan na distribuiranim objektima od čistog modela klijent – server. Primenom distribuiranih objekata mogu se kreirati klijent/server aplikacije sa troslojnom ili višeslojnom arhitekturom, koje omogućavaju implementaciju sinhronih ili ansihronih rešenja za Internet, intranet i/ili ekstranet.

Klijent - server arhitektura

Klijent – server arhitektura danas predstavlja osnovni model realizacije savremenih poslovnih, ali i drugih kategorija softvera.

Klijent je računarski sistem koji pristupa servisu na drugom računaru (serveru) preko neke vrste telekomunikacione mreže. U oblasti informacionih tehnologija server je računarski sistem koji pruža usluge drugim računarskim sistemima – klijentima. Komunikacija između servera i klijenta odvija se preko računarske mreže. Naziv server najčešće se odnosi na ceo računarski sistem, ali se ponekada koristi i samo za hardver ili softver takvog sistema. Klijent i server zajedno obrazuju klijent-server mrežnu arhitekturu.



Slika 9. Osnovna šema klijent - server arhitekture

Kada se pod pojmom servera podrazumeva računar, to se uglavnom odnosi na računar koji obavlja serverske poslove. Server se može sastojati od standardnih hardverskih komponenti koje se ugrađuju u obične desktop računare (PC – personal computer) u slučaju da programi (aplikacije) koji se izvršavaju na serverima nisu složeni, odnosno hardverski zahtevni. Serveri koji opslužuju složene programe ili veliki broj korisnika zahtevaju specijalizovan hardver koji je optimizovan za upotrebu u serverima. Poseban hardver podrazumeva i hard diskove visokih performansi, prvenstveno brzine i pouzdanosti. Procesorska brzina nije od ključne važnosti za servere pošto se većina servera bavi uzlazno/izlaznim (I/O – input/output) operacijama i ne koristi grafički korisnički interfejs (GUI – graphic user interface).

Pod serverom se, takođe, podrazumeva program koji od klijenta preko mreže prima zahteve, obrađuje ih i opet preko mreže šalje odgovore klijentu. Programi koji se koriste na serverima su posebno razvijani za serverske operativne sisteme i potrebe server/klijent okruženja. Primeri serverskih programa su web server, DHCP server, DNS server, mail server i drugo.

Operativni sistemi koji se koriste na serverima su specijalno dizajnirani za servere. Na serverima se najviše koriste Linux, Solaris i FreeBSD operativni sistemi koji su razvijeni po uzoru na operativni sistem Unix. Koriste se i serveri iz Microsoft Windows porodice: Windows NT, Windows 2000, Server 2003. Za operativne sisteme za servere karakteristično je: bezbednost i pouzdanost, mogućnost rekonfigurisanja softvera i hardvera bez zaustavljanja sistema (ograničeno), fleksibilnost mrežnog povezivanja.

Većina server programa radi u pozadini i ne očekuje ulazne podatke od korisnika niti ispisuje informacije na ekranu. Ovo je posledica toga da je jedina funkcija servera da komuniciraju samo sa klijent-računarima preko računarske mreže. Zbog toga na serverima vrlo često nema potrebe za postojanjem grafičkog radnog okruženja. Nekorišćenje takvog okruženja značajno oslobađa procesor servera za svoje namenske zadatke.

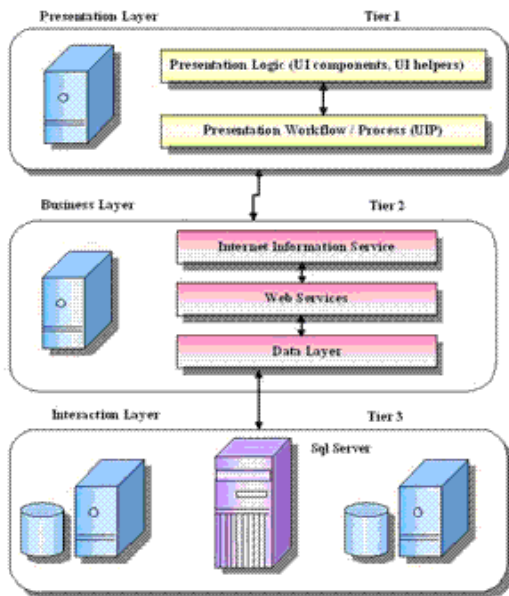
Neki primeri servera su: Mail server, Instant messaging server, Web server, FTP server, News server, Game server, Sound server, Peer-to-peer server, Print server, itd.

Troslojna klijent – server arhitektura

Razvoj personalnih računara, njihovo pojeftinjenje i omasovljenje, ogroman napredak u njihovim performansama, kao i razvoj komunikacionih tehnologija doveli su do nastanka značajnih promena i u postojećim oblicima obrade podataka u većim institucijama i preduzećima. Preteča svega su mainframe-ovi, veliki centralni računari na kojima su se nalazile baze podataka i na koje su bili priključeni “glupi” terminali, koji su predstavljali korisnički interfejs za manipulaciju informacijama iz baza. Ovakvi sistemi bili su neraskidivo vezani, u ekonomskom i eksploatacionom smislu, za firme koje su izrađivale i isporučivale hardver i softver. Proširivanje i nadogradnja sistema bili su skupi i teški, i uvek je postojalo ograničenje na postojeću hardversko-softversku platformu, a često samim tim i na jednu firmu koja je održavala sistem.

Takvi sistemi su nazvani dvoslojni (two-tier), zato što je prezentacioni nivo (sistem terminala) bio neposredno vezan i praktično neodvojiv od nivoa podataka (DBMS/operativni sistem). Razvoj personalnih računara i komunikacionih tehnologija, pre svega internet tehnologija uslovio je potrebu za nadogradnjom postojećih sistema i povezivanjem sa korisnicima sistema, bilo da su to zaposleni u organizaciji, korisnici ili potrošači, drugi delovi organizacije, srodne institucije ili poslovni partneri.

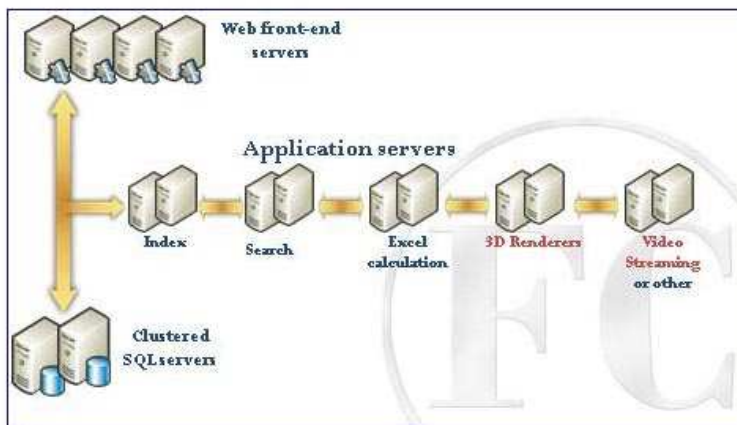
Spas je pronađen u troslojnim (three-tier) i višeslojnim (multi-tier) sistemima. Delovi aplikacije koji su zaduženi za poslovnu logiku instaliraju se na posebnom serveru, PC računari, mobilni uređaji i sl. obuhvataju korisnički interfejs (presentation layer), a baza je izdvojena na poseban server (database-tier). Na taj način su olakšani instalacija i održavanje softvera. Kako su korisnički programi postali mali, pitanje njihove distribucije rešeno je priključenjem na mrežu. Nadogradnja postojećih aplikacija je postala bezbolna – po priključenju na mrežu, aplikacija detektuje da postoji nova verzija, preuzme je sa mreže, instalira je i nastavlja rad.



Slika 10. Primer troslojne klijent-server arhitekture

Aplikacije na srednjem nivou se lako administriraju, a svaka promena na serveru ne zahteva promenu na korisničkim aplikacijama. Praktično, ostvaruje se nezavisnost u smislu odvojenog razvoja klijentskih aplikacija, dodavanja novih operacija unutar poslovne logike ili intervencija na samoj bazi.

Ukoliko na srednjem nivou ima više računara sa kojima korisnik može da komunicira u zavisnosti od tipa obrade, govorimo o višeslojnim (multi-tier) sistemima. Komponente aplikacije su tada distribuirane po različitim računarima: ulazimo u eru arhitekture distribuiranih aplikacija.



Slika 11. Primer višeslojne klijent-server arhitekture

S obzirom da se sistemi zasnovani na distribuiranim objektima razvijaju već decenija unazad, na tržištu postoji veliki broj različitih platformi (tehnologija) za implementaciju višeslojne arhitekture. Pri odabiru distribuirane tehnologije treba obratiti pažnju na sledeće parametre:

- prenosivost (portabilnost) klijentske i serverske platforme
- prenosivost programskog jezika
- performanse pri izvršavanju
- jednostavnost razvoja

- sigurnost

Rezultati testiranja sa različitim komunikacionim modelima za udaljeno pozivanje komponenata do danas su rezultirali sledećim tehnologijama:

COM+ / DCOM. COM+ predstavlja evoluciju starije tehnologije COM (Component Object Model). COM je specifikacija za objekte koja definiše interfejs preko koga različiti objekti mogu da komuniciraju. COM je nezavisan od programskog jezika ukoliko implementira COM interfejs. Teoretski može da se implementira na različitim operativnim sistemima, međutim, ne podržava ih niko osim Microsoft Windows-a. Da bi se omogućilo da COM objekti sa različitih sistema međusobno razmenjuju informacije, COM specifikacija je proširena i nastao je DCOM (Distributed COM). DCOM poseduje znatno kompleksniji model konfiguracije i sigurnosti.

CORBA / IIOP. CORBA (Common Object Request Broker Architecture) je konkurentna specifikacija kreirana od strane Object Management Group, grupe firmi koje razvijaju srednji sloj. CORBA je nezavisna od jezika i implementirana na većem broju platformi nego COM. Međutim, postoje nekompatibilnosti između implementacija različitih proizvođača. CORBA predstavlja object bus koji omogućava klijentu da poziva metode sa objekta na serveru uz nezavisnost programskog jezika i lokacije objekta. Interakcija je omogućena preko ORB (Object Request Brokers) komponenata na klijentu i na serveru a komunikacija se odvija preko IIOP (Internet Inter-ORB Protocol). Mogućnosti CORBA objekata definisane su pomoću IDL (Interface Definition Language).

EJB / RMI over IIOP. EJB (Enterprise JavaBeans) je specifikacija koju je izdao Sun Microsystems za Java Platformu. EJB je nezavisna od platforme, ali ne i od jezika. Svi EJB objekti moraju biti napisani u jeziku Java. Za komunikaciju između različitih sistema, EJB koristi varijantu IIOP nazvanu RMI preko IIOP (Remote Method Invocation over IIOP). RMI je protokol rezervisan samo za Javu.

SOAP. SOAP (Simple Object Access Protocol) je kompletno kreiran na postojećim, proverenim i široko prihvaćenim tehnologijama kao što su HTTP i XML. SOAP koristi XML za prenos podataka između aplikacija, a pošto je XML univerzalni standard, sve platforme mogu da pristupe i obrade informaciju. Pošto koristi HTTP, jednostavno prolazi kroz port 80, tako da firewall-ovi ne predstavljaju problem. Pristup različitim aplikacijama na raznim platformama sa SOAP-om postaje jednostavan, Java aplikacija na Unix-u jednostavno može da poziva metode COM objekta na Windows serveru. Klijentska aplikacija na iMac-u pristupa objektu na mainframe računaru. Sve ovo postaje transparentno i ne zahteva bilo kakvu posebnu administraciju.

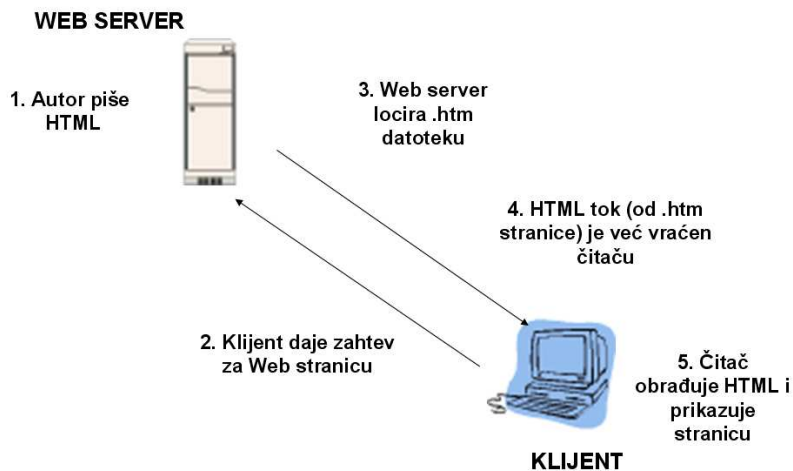
Web aplikacije

Danas, kada web predstavlja sve dominantniji medij za kolaboraciju i integraciju različitih poslovnih softvera unutar lanca snabdevanja, elektronskog tržišta, ali i ljudi, unutar određene interesne zajednice, od suštinske je važnosti da razvoj tehnologija koji će to omogućiti na efikasan način, izlazi u susret narastajućim potrebama tržišta u ovom segmentu.

Osnovne tehnologije za izradu web aplikacija su HTML jezik za izradu statičkih strana i Server-Side Scripting jezici za generisanje HTML koda, odnosno, izradu dinamičkih strana.

Preuzimanje i prikaz statičkih web strane predstavlja jednu realizaciju klijent – server arhitekture, čiji su akteri internet pretraživač, kao klijent i web server, kao server, u ovoj arhitekturi. Web server predstavlja softver, koji se instalira na određenom, dostupnom serverskom računaru i koji upravlja zahtevima za pristup određenoj web stranici, tako što postupa po zahtevu klijenta, isporučujući sadržaj neke web strane. Web server i pretraživač komuniciraju razmenom poruka putem HTTP (Hyper Text Transfer Protocol) protokola, koje mogu biti u vidu zahteva (koji čini odgovarajuće HTTP zaglavlje - header) i odgovora, koji čini odgovarajuće HTTP zaglavlje i sadržaj samog odgovora, odnosno HTML kod koji se interpretira od strane internet pretraživača i prikazuje na njegovoj radnoj površini.

Tok ove komunikacije je prikazan na slici 12.



Slika 12. Tok komunikacije u prikazu statičkih web strana

Osnovna tehnologija za izradu web strana danas predstavlja HTML jezik. On se koristi za definisanje sadržaja ili elemenata interakcije (linkovi ili forme) jedne web strane, tako što definiše komande (odnosno, tagove), koje internet pretraživači, kao što su Internet Explorer ili Firefox, mogu da interpretiraju na pravilan način i, prema toj interpretaciji, korisniku jednog web sajta prikažu njegov sadržaj.

HTML je nastao uproščavanjem SGML (Standard Generalized Markup Language, standardizovani uopšteni jezik za označavanje) standarda sa svrhom opisa dokumenta koji se objavljuju na webu. U početku je bio prilično ograničen što se označavanja sadržaja tiče i pružao je uglavnom elementarne mogućnosti za označavanje i formatiranje teksta (paragrafi, naslovi, citati itd.). Kako je web rastao, tako je rasla i potreba za bogatijim sadržajem te je u tom smeru razvijan i HTML standard. Tada su standardu dodate elementi za opis tabela, slika, slojeva, napredno formatiranje teksta itd.

HTML dokumenti se sastoje iz dva osnovna dela: dela koji opisuje dokument i dela koji predstavlja sadržaj dokumenta. Informacije koje opisuju sam dokument se smeštaju u head tag, dok se sam sadržaj smešta u body tag. Oba ova elementa se nalaze unutar html taga.

```
<html>
  <head>
    <title>Наслов документа</title>
  </head>
  <body>
    <h1>Пример документа</h1>
    <p>Ово је пример једног простог HTML документа.</p>
  </body>
</html>
```

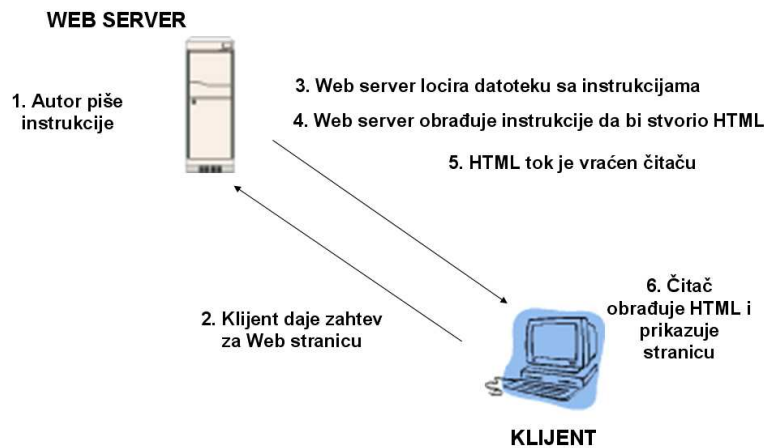
Slika 13. Primer osnovne strukture HTML dokumenta

HTML dokumente, odnosno statičke web strane čini čisti HTML kod, i one predstavljaju datoteke sa ekstenzijom htm ili html, koje se kodiraju i smeštaju na odgovarajuće mesto na web serveru.

Osnovni problem primene statičkih web strana u složenijim web aplikacijama je to što je njihov sadržaj i funkcionalnost unapred određen. Samim tim, statičke web strane se ne mogu koristiti za kreiranje interaktivnih i ažurnih web aplikacija, kod kojih prikazani sadržaj u svakom trenutku zavisi od trenutnih okolnosti. Na primer, kod web aplikacija za komunikaciju između ljudi, kao što su web forumi, sadržaj jedne strane koja prikazuje diskusiju o određenoj temi se stalno menja – svaki put kada neki korisnik te aplikacije da doprinos diskusiji. Dalje, web prikaz kataloga proizvoda se, takođe, kontinuirano menja – svaki put kada se promeni dostupnost nekog proizvoda na lageru, doda novi proizvod ili modifikuju atributi postojećeg – cena ili drugi atributi.

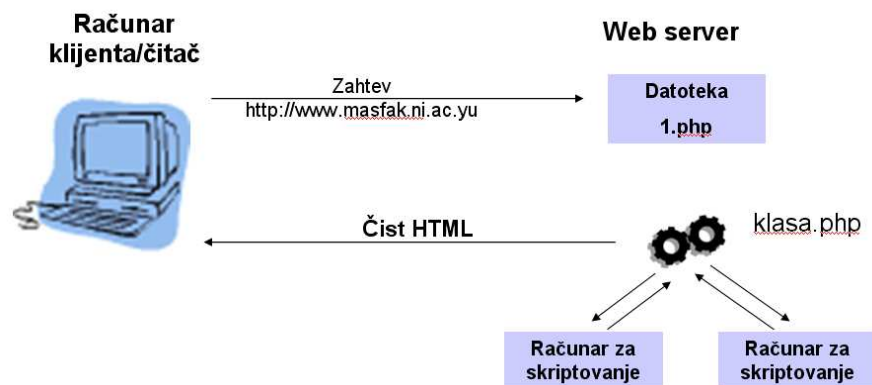
Očigledno, postoji potreba za kreiranjem novog sloja u komunikaciji pretraživača i web servera, koji treba da programski obradi zahtev i parametre zahteva i, prema njihovom sadržaju, automatski generiše odgovarajuću web stranu ili obradi zahtev i skladišti njegove parametre na odgovarajući način (uobičajeno, u neku relacionu bazu podataka). To se radi primenom tzv. dinamičkih web strana – skupa instrukcija koje se po prijemu zahteva izvršavaju i automatski kreiraju HTML kod, koji se, u okviru HTTP odgovora isporučuje pretraživaču.

Na slici dole, prikazana je šema toka komunikacije između pretraživača i web servera u prikazu dinamičkih web strana.



Slika 14. Tok komunikacije u prikazu dinamičkih web strana

Za kreiranje instrukcija čija je namena automatsko generisanje HTML koda, na osnovu parametara zahteva, koriste se različiti programski jezici i njihove varijante, kao što su PHP, JSP (Java), ASP (Visual Basic), itd. Ove instrukcije nazivamo skriptovima na strane servera (Server-Side Scripts), zato što se izvršavaju na web serverima.



Slika 15. Obrada skripta na serveru

Ovi skriptovi uobičajeno ostvaruju vezu sa odgovarajućom bazom podataka u koju se smeštaju obrađeni podaci i zahtevi i koja se, po odgovarajućem zahtevu i interpretira, sa ciljem prikaza određenih podataka.

Pored navedenih, druge prednosti korišćenja tehnologija na strani servera za kreiranje dinamičkih web strana su:

- Ova tehnologija omogućava da se neki program izvršava u programskim jezicima koje web pretraživači ne podržavaju.

- Ona daje mogućnost da se programiraju dinamičke Web aplikacije nezavisno od čitača, bez pribegavanja programiranju na strani klijenta, pomoću Java apleta, DHTML-a i ActiveX kontrola, od kojih svaki zahteva određeni internet pretraživač, odnosno njegove konkretne karakteristike.
- Ona može klijentu (pretraživaču) da dostavi podatke koji su mu inače nedostupni.
- Često ostvaruje brže vreme učitavanja.
- Obezbeđuje poboljšane mere bezbednosti.

Ipak, važno je napomenuti da korišćenje dinamičkih web strana povećava opterećenje web servera, naročito ukoliko njima pristupa veći broj korisnika. Iz ovog razloga, uobičajeno je potrebna veća inicijalna investicija u hardver web servera, koji se koriste za generisanje dinamičkih web strana.

Jezik koji se najčešće koristi za kreiranje dinamičkih web strana je PHP. PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C. Tokom vremena jezik se proširivao i sticao mogućnosti za objektno orijentisano programiranje, naročito od verzije 5.0. Nalikuje jeziku C++ u smislu da dozvoljava i čisto-proceduralno programiranje, ali omogućava i korišćenje klasa i drugih koncepata objektno orijentisanog programiranja (nasleđivanje, apstraktne metode, interfejsi itd.).

Osnovni standardi za razmenu podataka

Danas, kada funkcionalnost poslovnog softvera u sve većoj meri zavisi od njegove sposobnosti da se integriše i razmenjuje informacije sa drugim poslovnim softverima, od vrhunskog značaja je postojanje i kontinuirani razvoj standarda za razmenu podataka. Ovi standardi postoje da bi komunikaciju i integraciju poslovnih softvera učinili efikasnom i prepoznatljivom i smanjili napor u implementaciji odgovarajućih interfejsa.

Može se reći da je tehnološki razvoj, naročito u segmentu razvoja interneta, odnosno weba, danas ustanovio jedinstvenu tehnologiju za modeliranje struktura podataka koje se razmenjuju između poslovnih softvera. Ta jedinstvena tehnologija je XML jezik.

XML jezik za modeliranje strukture podataka

XML (eXtensible Markup Language) jezik predstavlja standardni način za modeliranje struktura podataka u elektronskom poslovanju. Iako je prvobitno namenjen primeni u web aplikacijama, postao je standard za predstavljanje podataka u poslovnim procesima, prihvaćen u potpunosti od strane najvećih proizvođača softvera i razvojne zajednice. Ideja je bila da se stvori jezik koji će i ljudi i računarski programi moći jednostavno da čitaju. XML definiše opštu sintaksu za označavanje podataka pomoću odgovarajućih tagova koje imaju poznato ili lako razumljivo značenje. Format koji obezbeđuje XML za računarske elemente može se prilagoditi najrazličitijim oblastima, kao što su elektronska razmena podataka, čuvanje podataka, odvajanje podataka od prezentacije, vektorska grafika, sistemi glasovne pošte, izrada novih specijalizovanih jezika za označavanje. Osnovna svrha XML-a je da olakša deljenje podataka kroz različite informacione sisteme, posebno kroz one sisteme koji su povezani sa Internetom.

XML je nastao iz potrebe da se same informacije sa HTML strana fizički odvoje od načina na koji se prikazuju unutar web strana (dizajna). Sam HTML je zasnovan na SGML-u (Standard Generalized Markup Language), opširnijem meta jeziku koji je postojao i pre pojave Web-a. SGML definiše gramatiku za sve markup jezike dokumenata. SGML dokumenti nose svoju gramatičku definiciju sa sobom u obliku Document Type Definition-a (DTD) datoteka. DTD datoteke definišu sve oznake (tagove) koje se koriste u jednom dokumentu kao i njihovu sintaksu.

Struktura XML datoteke je hijerarhijska – sastoji se od otvorenih i zatvorenih tagova, unutar kojih su drugi tagovi. Tagovi su opisani u DTD (Document Type Definition) datoteci, koja predstavlja svojevrsnu gramatiku jezika. Za razliku od SGML-a, HTML-a i drugih markup jezika, XML dozvoljava korisniku slobodno kreiranje gramatike jezika. Iz tog razloga se za formiranje strukture

podataka, mogu koristiti i prirodni jezici. Za tumačenje XML strukture i transformaciju u određenu strukturu podataka, koriste se tzv. parseri, programski moduli, napisani u odgovarajućem jeziku.

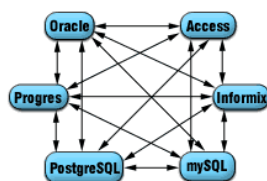
```
<osoba>
  <ime i prezime>
    <ime>Miloš</ime>
    <prezime>Crnjanski</prezime>
  </ime i prezime>
  <zanimanje>književnik</zanimanje>
  <zanimanje>diplomata</zanimanje>
</osoba>
```

Slika 16. Primer XML dokumenta

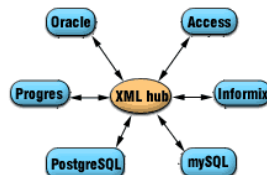
Iako XML ne zahteva striktnu specifikaciju gramatike dokumenta, dobra je praksa da se ona formira, da bi odgovarajući parseri mogli da izvrše validaciju dokumenta opisanog XML datotekom.

Najvažnije prednosti korišćenja XML standarda za modeliranje podataka u poslovnim web aplikacijama su:

- **Samodokumentovanje.** Korišćenje prirodnih jezika za opis strukture podataka. Kriva učenja XML jezika je mnogo strmija od učenja predstavljanja struktura podataka npr. u relacionim bazama. Jedan od osnovnih ciljeva XML specifikacije je integrisanje nove tehnologije za opis podataka u najkraćem mogućem roku.
- **Veliki stepen konzistentnosti sa HTML jezikom – jezikom web-a.** Poznavanje HTML jezika dramatično ubrzava usvajanje XML.
- **XML je danas usvojen kao de facto i de jure standard za EDI (Electronic Data Interchange).** On omogućava očuvanje konzistentnosti dokumenata prilikom razmene i ukida potrebu za prevodiocima u i iz različitih formata.
- **XML reprezentacija kompletne baze ili nekog njenog dela može poslužiti za backup, nezavistan od RDBMS (sistema za upravljanje relacionim bazama podataka),** ili za migraciju podataka sa jednog sistema na drugi (npr. Oracle > SQL Server). Migracija se može izvesti na dva načina, pisanjem koda koji će vršiti svaku pojedinačnu migraciju ili uz pomoć takozvanog centralnog XML hub-a. Prednost centralnog hub-a je u tome da su za dodavanje novog formata baze potrebna samo dva nova konvertora, dok je istovremeno bez hub-a, za šest sistema prikazanih na drugoj slici potrebno 30 filtera.



slika 1



slika 2

Slika 17. Backup podataka iz relacionih baza primenom XML-a

- **Nezavisnost prikaza XML u odnosu na web browser, kod browser-a koji podržavaju XML.** U ovom trenutku, najnovije verzije gotovo svih vodećih browser-a podržavaju XML. Ovo je posebno važno ukoliko je potrebno da se podaci iz baza prezentuju na Internetu,

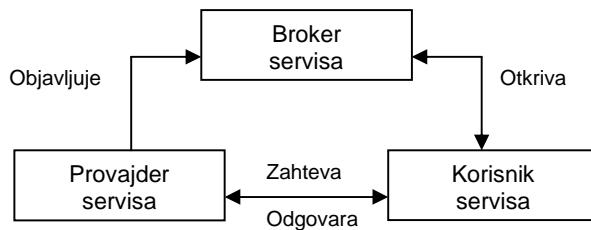
gde postoje različiti browser-i, za razliku od korporativnih mreža (intraneta), gde je browser uglavnom unificiran. Trenutno je najčešći način za prikaz XML-a na Internetu korišćenje XSLT-a (XML Style Language Transformation) za konverziju XML-a u HTML (simultana konverzija).

Uvod u web servise

Web servisi predstavljaju novu primenu standardnih softverskih tehnologija koja omogućava povezivanje raznorodnih informacionih sistema na novi način, korišćenjem internet protokola, i to unutar jednog integrisanog ili više poslovnih procesa. Osnovni motiv razvoja i korišćenja tehnologije web servisa predstavlja mogućnost integracije raznovrsnih informacionih sistema instaliranih u distribuiranim okruženjima. Oni predstavljaju skup aplikacionih funkcija koje se mogu programski pozivati putem interneta.

Jedan web servis je bilo koji skup metoda za obradu i isporuku informacija, dostupan posredstvom HTTP i SMTP protokola, a koji nije vezan za određenu operativnu platformu ili programski jezik i koji koristi XML notaciju za razmenu poruka.

Web servisi predstavljaju tehničku evoluciju koncepta servisno orijentisane arhitekture. U arhitekturi okruženja koje karakterišu web servisi, jedna mrežna komponenta može imati ulogu provajdera servisa, korisnika ili klijenta servisa, i brokera servisa.



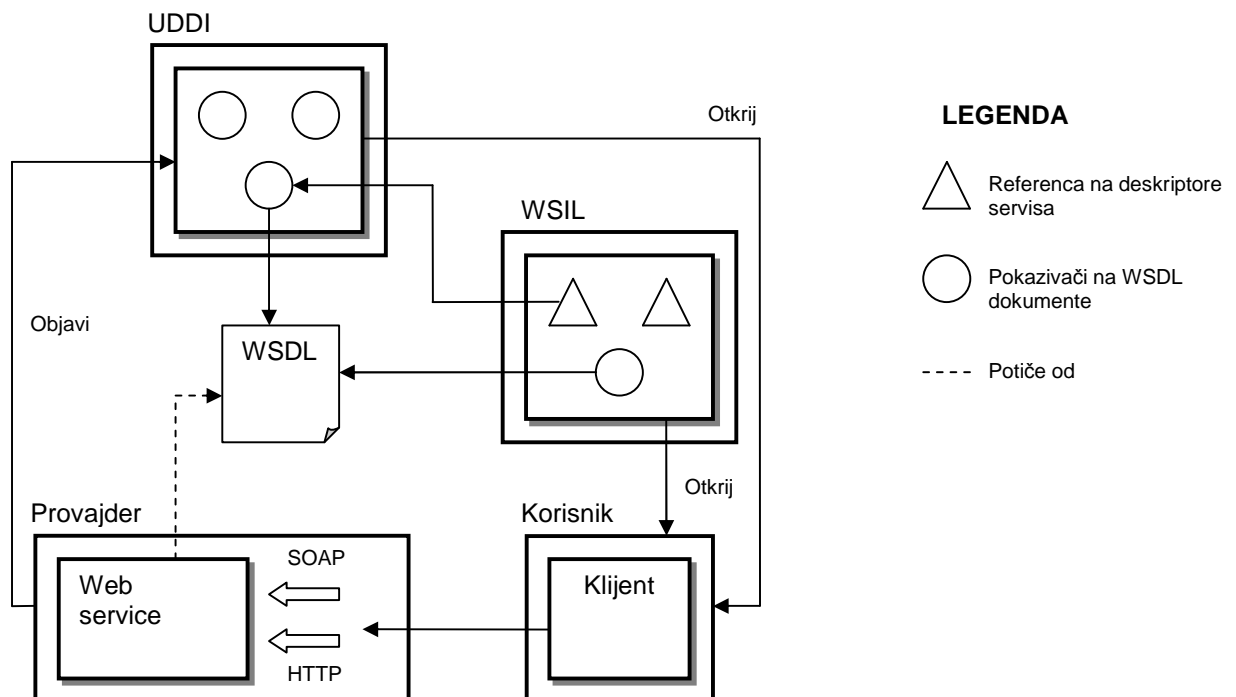
Slika 18. Uloge i interakcije u arhitekturi web servisa

Provajder servisa je odgovoran za kreiranje i aktiviranje web servisa u njegovoj nadležnosti, kao i objavljivanje dostupnosti servisa, opisanog putem WSDL standarda u registar servisa, kao što je UDDI poslovni registar.

Uloga brokera servisa je registracija i kategorizacija objavljenih servisa i obezbeđenje funkcija za njihovo pretraživanje. U arhitekturi web servisa, UDDI ima ulogu registra servisa opisanih putem korišćenjem WSDL standarda.

Osnovna karakteristika internet standarda je da su njihov fokus protokoli, a ne njihove implementacije. S obzirom na to da internet čini niz heterogenih tehnologija, ovaj nivo generičnosti garantuje funkcionisanje u nehomogenom okruženju i visoku interoperabilnost.

Mogućnosti integracije jedinstvenog informacionog sistema, odnosno prevazilaženja nekompatibilnosti njegovih komponenti se ostvaruju primenom skupa softverskih standarda kao što su XML, SOAP (Simple Object Access Protocol), UDDI (Universal Description Discovery and Integration), WSDL (Web Services Description Language), WSIL (Web Services Inspection Language), WS-S (Web Services Security) i WS-I (Web Services interoperability). Ovi standardi omogućuju definisanje, pakovanje, pristupanje i izvršavanje podataka i programa preko Interneta, bez potrebe za vođenjem računa o pojedinačnim implementiranim tehnologijama.



Slika 19. Relacije između SOAP, UDDI, WSDL i WSIL protokola

Provajder servisa obezbeđuje njegovu dostupnost uz pomoć svoje tehničke infrastrukture, koja podržava SOAP/HTTP ili SOAP/JMS (Java Messaging Services) komunikaciju. Web servis je opisan u WSDL dokumentu koji se skladišti na serveru provajdera u posebnom repozitorijumu. Reference na web servis, odnosno podaci koji ukazuju na WSDL dokument, mogu da postoje u UDDI registru ili WSIL dokumentima.

WSDL predstavlja otvorenu specifikaciju, zasnovanu na XML standardu koja opisuje interfejsa ka web servisima na mreži i njihove instance. WSDL sintaksom se opisuju sve vrste aktivnosti koje može da izvrši neki web servis. Ona omogućava da se čvorovi servisno orijentisane arhitekture opišu, bez obzira na formate poruka koje razmenjuju ili mrežne protokole koje koriste za razmenu. Dostupnost WSDL dokumenata se obezbeđuje kroz UDDI ili WSIL, ili javnim objavljivanjem njihovog URL-a nekim drugim konvencionalnim sredstvom.

Primenom njegove sintakse, mogu se definisati sledeće karakteristike web servisa:

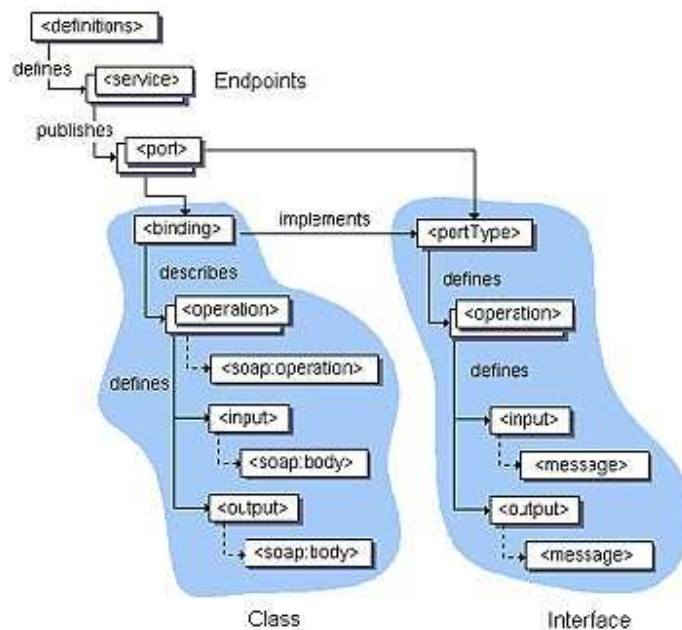
- Naziv web servisa i informacije o njegovom adresiranju
- Protokol i encoding stil koji će se koristiti prilikom pristupanja javnim operacijama web servisa
- Strukturne informacije – meta podaci servisa, kao što su operacije, parametri i tipovi podataka koji čine interfejs web servisa.

Jedan WSDL dokument definiše servise kao skup mrežnih završetaka (network endpoints) ili portova. WSDL omogućava apstraktnu definiciju operacija i poruka i njihovo referenciranje - vezivanje (binding) za tačno određeni mrežni protokol i format poruke. Pritom, definicija operacija i poruka je nezavisna od parametara njihovog vezivanja, odnosno korišćenih protokola i formata poruka. Parametri vezivanja predstavljaju ugovor između logike klijenta i logike servera servisa – posrednika između meta opisa web servisa i njegove konkretne realizacije.

U tradicionalnom distribuiranom programiranju, realizacija poslovne logike, odnosno, logike na serverskoj strani, obuhvata veoma često statičko vezivanje (static binding) - generisanje proxy objekata koji su namenjeni klijentima, za ostvarivanje njihove komunikacije sa serverom, primenom predviđenih protokola, od kojih jedan može da bude i SOAP. Ipak, jedan od osnovnih

ciljeva servisno orijentisane arhitekture je dinamičnost web servisa – oni mogu biti predmet izmene, bez ikakve prethodne najave, ili čak – ponovne distribucije klijentskih proxy-ja. Razlog za to je, između ostalog i činjenica da su njihovi vlasnici veoma često – treće strane, a sadržaj usluga definisanih web servisom – unapred nepoznat.

Očigledno, navedeni princip obezbeđuje karakteristiku proširivosti web servisa, ali i višestrukog korišćenja definisanih formata poruka. Demonstracija ovog principa uz pomoć elemenata strukture WSDL i njihov odnos je prikazana na slici dole.



Slika 20. Elementi strukture WSDL

Za definisanje mrežnih servisa, WSDL koristi sledeće elemente sintakse:

- Tipovi (Types). Kontejneri za definicije tipova podataka, definisanih uz pomoć XML sintakse, odnosno odgovarajuće XSD strukture. XML i XSD definicije tipova se koriste u cilju ostvarivanja maksimalne neutralnosti formata.
- Poruka (Message). Apstraktna definicija podataka koji se razmenjuju u komunikaciji između provajdera i korisnika web servisa. Svaka poruka se može sastojati iz jednog ili više delova. Delovi poruke su analogni atributima poziva jedne operacije.
- Operacija (Operation). Apstraktni opis akcije podržane od strane servisa.
- Tip porta (Port Type). Apstraktni skup operacija podržan od strane jednog ili više mrežnih završetaka.
- Referenciranje - vezivanje (Binding). Specifikacija konkretnog protokola i specifikacija formata podataka za određeni tip porta. Kao protokol se obično koristi SOAP, dok format podataka može biti definisan kao dokument (document) ili nekodirani tekst (literal). Drugi izbor predstavlja HTTP – izborom ovog protokola, format poruke se ograničava na proste tipove (npr. tekst), dok se SOAP koristi za složenije strukture podataka.
- Port. Mrežni završetak definisan kao kombinacija parametara vezivanja i mrežne adrese.
- Servis (Service). Skup mrežnih završetaka.

Prilikom implementacije web servisa uz pomoć Java programskog jezika, poslovna logika web servisa je enkapsulirana u Java bean-ovima ili EJB (Enterprise Java Bean). Savremene Java

razvojne platforme omogućavaju generisanje WSDL dokumenata na osnovu strukture postojećih Java bean-ova ili EJB, i obrnuto.

Sa stanovišta klijenta web servisa, važno je istaći da ove platforme mogu omogućiti i generisanje Java proxy koda, namenjenog upravo klijentu, na osnovu strukture postojećeg WSDL dokumenta i tako obezbediti lak i brzi razvoj klijenta za korišćenje usluga web servisa.

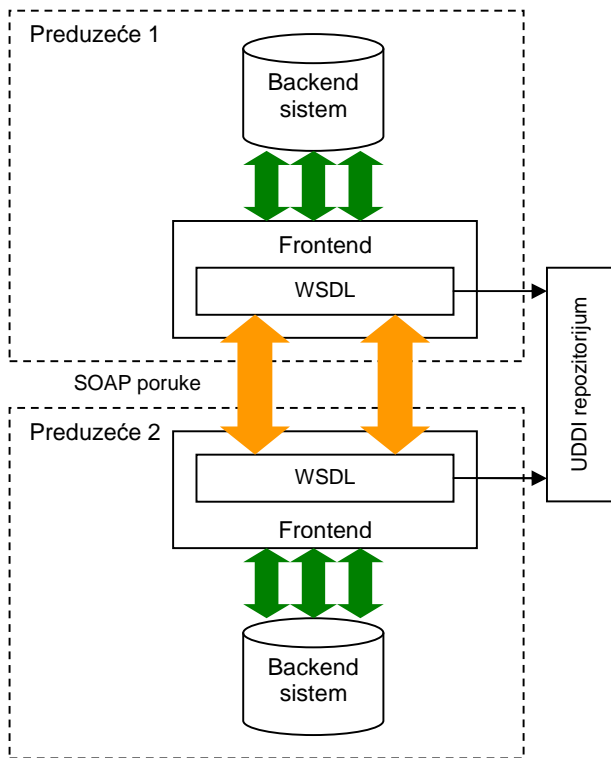
Razvoj web servisa

Razvoj web servisa predstavlja integralni deo razvoja ukupne servisno orijentisane arhitekture. Osnovne aktivnosti procesa razvoja web servisa su:

- Otkrivanje (Discover). Pretraživanje UDDI poslovnih registara ili WSIL dokumenata u cilju lociranja postojećih web servisa sa kojima je potrebno izvršiti integraciju.
- Kreiranje ili transformacija (Create or Transform). Ovaj proces karakterišu dva moguća pristupa – top-down ili bottom-up. Top-down pristup podrazumeva kreiranje WSDL interfejsa uz pomoć odgovarajućeg editora i generisanje skeleta odgovarajućih klasa na osnovu njegove strukture. Bottom-up pristup podrazumeva generisanje WSDL datoteka na osnovu postojećih implementacija, sadržanih u postojećim Java bean-ovima ili EJB.
- Postavljanje (Deploy). Instaliranje web servisa u okruženjima za testiranje.
- Testiranje (Test). Provera kvaliteta funkcionisanja web servisa, lokalno i u distribuiranom okruženju.
- Razvoj (Develop). Razvoj klijentskih aplikacija.
- Objavljivanje (Publish). Objavljivanje informacija o web servisu u UDDI registrima.

Integrisanje poslovnog softvera primenom web servisa

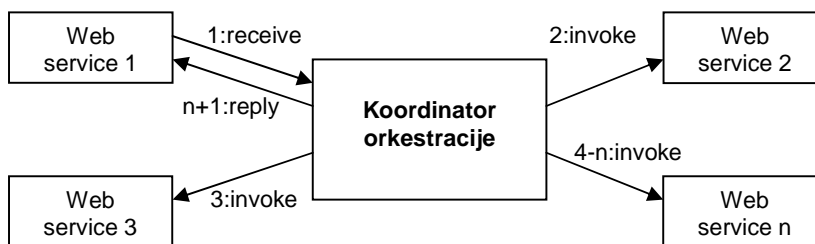
Dijagram sa primenom integracije dva poslovna informaciona sistema primenom web servisa je prikazan na slici dole. WSDL datoteke svakog poslovnog okruženja obuhvataju skup metoda, predstavljenih imenom, argumentima, tipom rezultata i poslovnom logikom, koje su dostupne autorizovanim korisnicima ili su javne.



Slika 21. Primer integracije dva poslovna sistema posredstvom web servisa

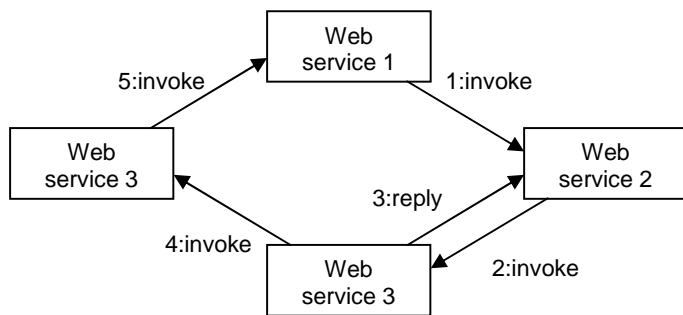
Sa stanovišta integracije izolovanih aplikacija i njihovih funkcija uz pomoć web servisa, oni se mogu organizovati na dva načina: uz pomoć orkestracije (*Orchestration*) ili koreografije (*Choreography*).

Orkestracija se uobičajeno primenjuje u privatnim okruženjima (preduzeća) i podrazumeva postojanje centralni proces (koji takođe može biti web servis) sa ulogom koordinacije i kontrole svih ostalih web servisa. Pritom, nijedan web servis, osim centralnog, nije svestan svoje uloge u orkestriranom okruženju.



Slika 22. Orkestracija web servisa

Sa druge strane, u javnim okruženjima, u okviru kojih postoji veliki broj korisnika usluga i veći nivo kolaboracije u ostvarenju nekog cilja, za integrisanje web servisa se primenjuje metod koreografije. Akteri koreografije su web servisi koji tačno znaju sa kim mogu razmenjivati poruke i kada.



Slika 23. Koreografija web servisa