Servisno orjentisane arhitekture

Branislava Živković 1040/2014
Matematički fakultet
Univerzitet u Beogradu

Sadržaj

- Uvod
- Servisi kao ponovo upotrebljive komponente
- Servisno inžinjerstvo
- Razvijanje softvera pomoću servisa

Uvod

Šta je to servis?

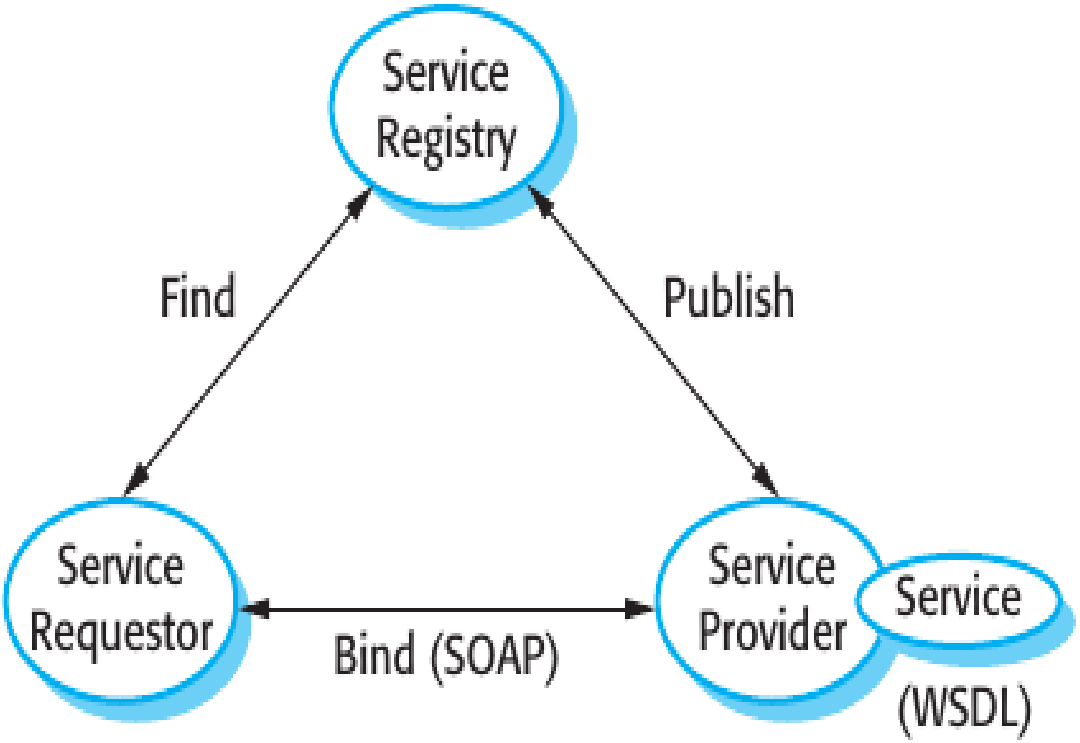
'Servis je usluga ili akcija koju jedna strana nudi drugoj. Iako proces može biti vezan za fizički proizvod, izvršavanje je suštinski sakriveno I ne rezultira u vlasništvu nijednog od učesnika.'

- Pružanje servisa je nezavisno od aplikacije koja ga koristi.

Servisno orjentisane arhitekture

- Servisno orjentisane arhitekture predstavljaju način izgradnje distribuiranih sistema kod kojih su komponente sistema samostalni servisi koji se izvršavaju na različitim računarima
- Servisi ne zavise od platforme I jezika na kom su implementirani
- Razvijani su standardni protokoli za komunikaciju I razmenu informacija

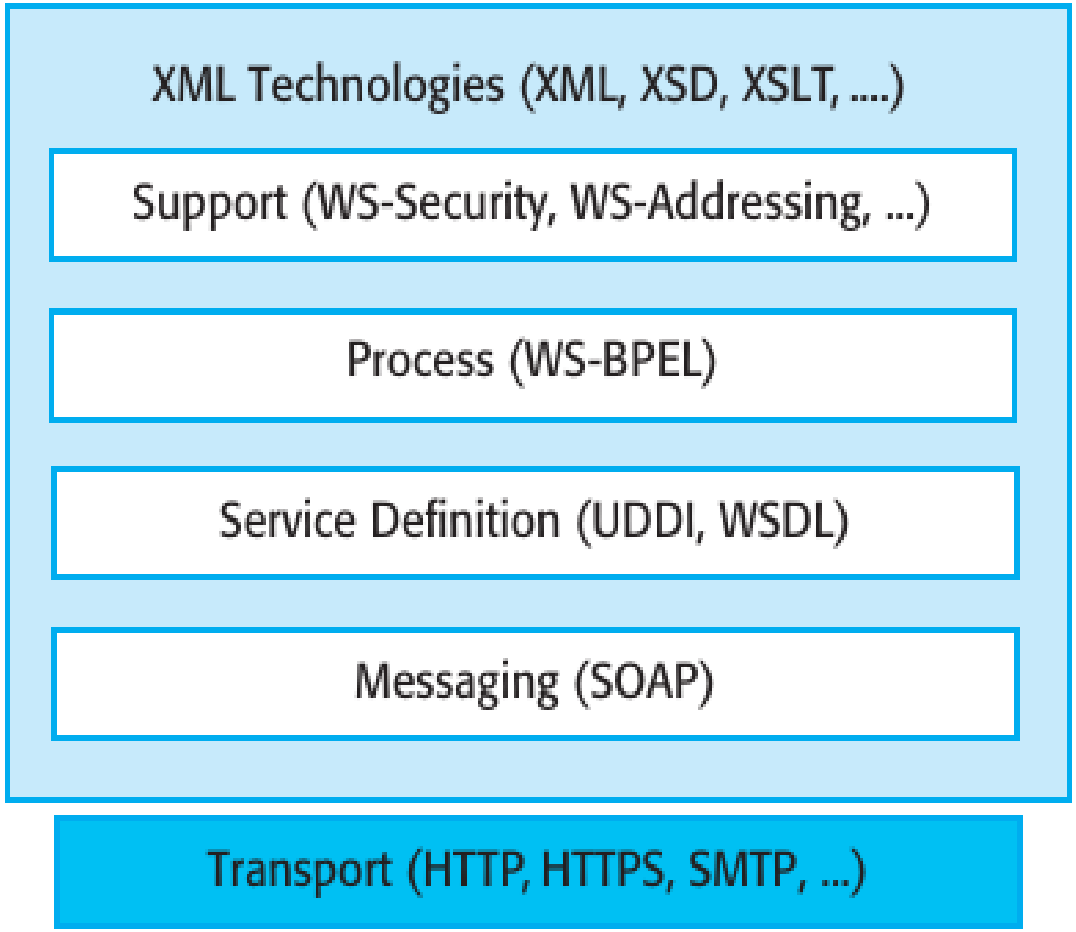
Servisno orjentisane arhitekture



Standardi

- SOAP
  - standard za razmenu poruka i komunikaciju između servisa
  - definiše ključne i opcione komponente poruka koje se razmenjuju
- WSDL (The Web Servise Definition Language)
  - je standard za definisanje interfejsa servisa
  - opisuje način definisanja operacija i povezivanja servisa
- WS-BPEL
  - standardizovan jezik koji se koristi za definisanje procesa koji uključuju više različitih servisa

Standardi


XML Technologies (XML, XSD, XSLT, ...)
Support (WS-Security, WS-Addressing, ...)
Process (WS-BPEL)
Service Definition (UDDI, WSDL)
Messaging (SOAP)
Transport (HTTP, HTTPS, SMTP, ...)

RESTful Web Servisi

- REST (Representational State Transfer)
  - je stil arhitekture koji se zasniva na prenosu reprezentacija resursa od servera ka klijentu
- To je stil koji se koristi na webu jer je jednostavniji za implementaciju servisa od SOAP/WDSL standarda
- RESTful servis je identifikovan svojim URI-em I komunicira koristeći HTML protokol
- RESTful servisi povlače niže troškove I zato ih koriste mnoge organizacije

Servisno orjentisane I komponentno orjentisane arhitekture

- Servisi i komponente očigledno imaju mnogo toga zajedničkog
- I jedni i drugi su ponovo upotrebljivi elementi i možemo posmatrati komponente kao pružaoce servisa
- I pored toga, i dalje postoji velika razlika između servisa i komponenti kao i između servisno i komponentno orjentisanog softverskog inžinjerstva

Servisi kao ponovo upotrebljivi elementi

Servis

- Servis možemo definisati I ovako:

'Oskudno povezani, ponovo upotrebljivi
softverski elementi
koja enkapsuliraju diskretnu funkcionalnost,
mogu biti distribuirani
I programski dostupni.
Web servis je servis kome možemo pristupiti
koristeći standardne
Internet I XML protokole.'

- Servisi treba da budu nezavisni, slabo povezani I da se uvek ponašaju na isti način
- Servisi komuniciraju preko poruka koje su predstavljene XML-om

WSDL – Web Service Definition Language

- Prilikom korišćenja servisa potrebno je da znamo njegovu lokaciju (URI)
  I detalje interfejsa
- Ti podaci se nalaze u opisu servisa koji je predstavljen
  XML zasnovanim jezikom WSDL
- WSLD specifikacija definiše tri stvari:
  - šta servis radi
  - kako servis komunicira
  - gde se on nalazi

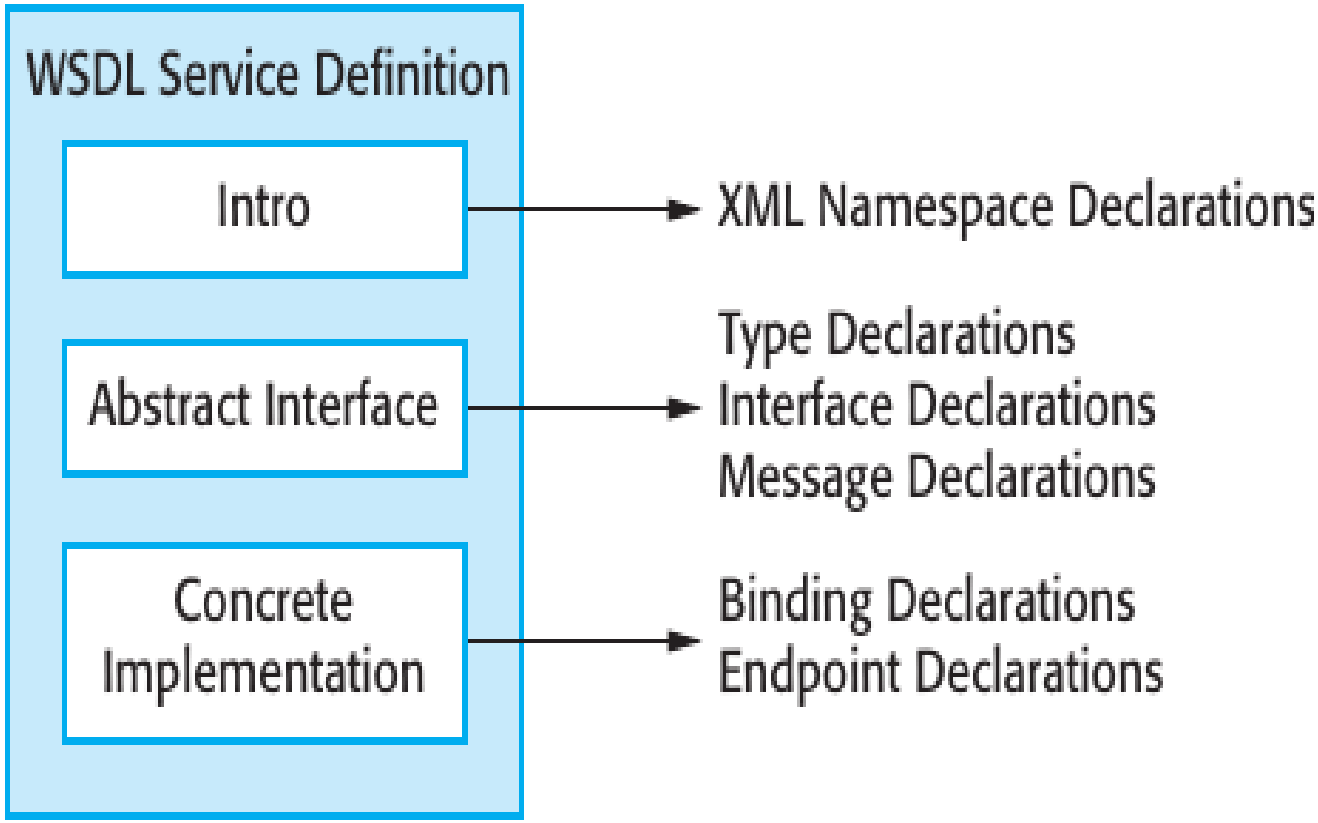
WSDL

- Šta servis radi?
  - Ovaj deo WSDL dokumenta se naziva interfejs
  - Precizira koje operacije servis podržava
  - Definiše format poruka koje servis šalje I prima
- Kako servis komunicira?
  - Ovaj deo WSDL dokumenta se naziva povezivanje
  - Definiše tehničke detalje oko načina komunikacije sa servisom
  - Apstrakcija interfejsa je predstavljena konkretnim skupom protokola
- Gde se servis nalazi?
  - Ovaj deo WSDL dokumenta opisuje lokaciju konkretne implementacije servisa

WSDL – Konceptualni model

- Uvodni deo uglavnom definiše XML prostore imena koji se koriste I koji mogu da uključe delove dokumenta koji pružaju dodatne informacije o servisu
- Opcioni opis tipova
  koji se koriste u porukama koje servis razmenjuje.
- Opis interfejsa servisa
  - operacije koje servis pruža korisnicima.
- Opis ulaznih I izlaznih poruka
- Opis protokola povezivanja, uglavnom je to SOAP ali mogu biti I drugi
- Krajnja specifikacija predstavljena URI adresom fizičke lokacije servisa

WSDL – Konceptualni model

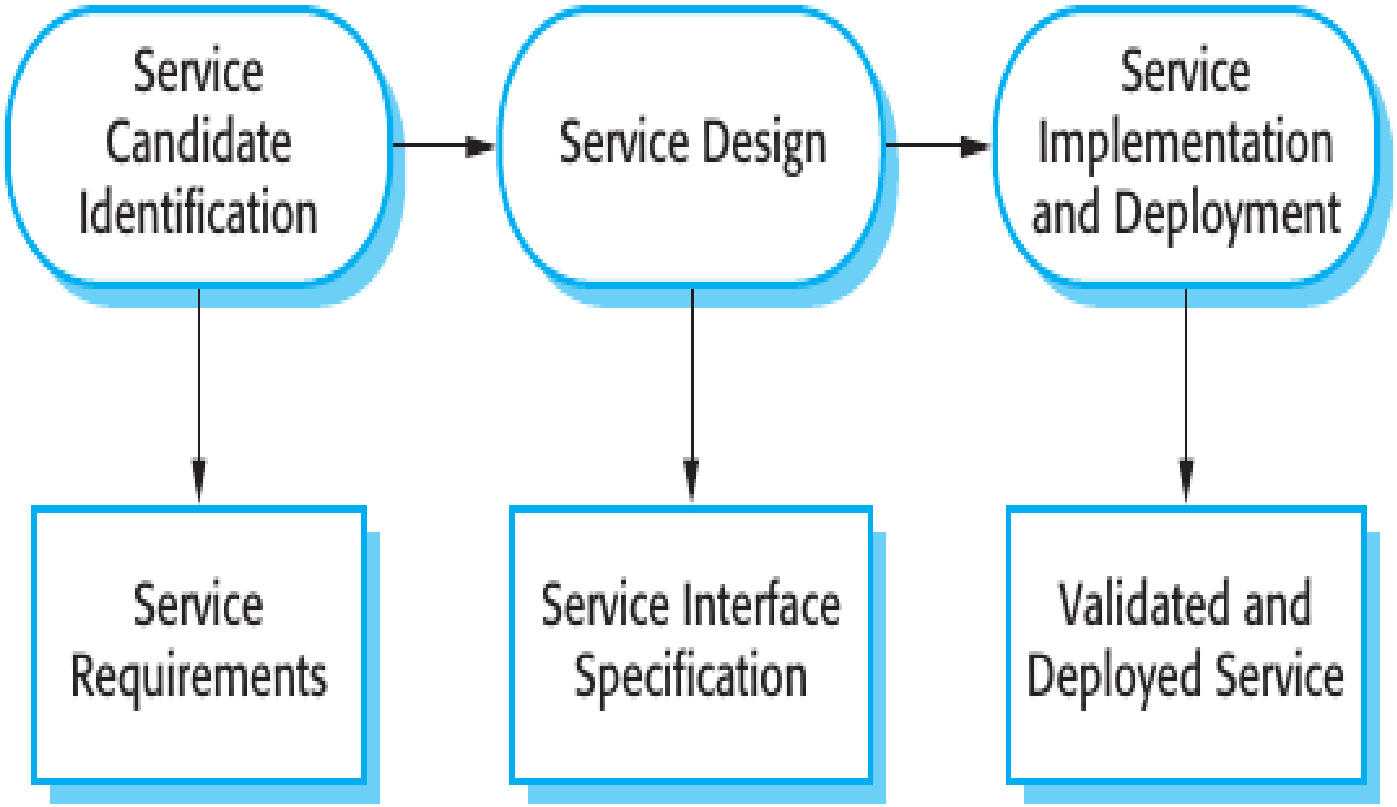


Servisno inžinjerstvo

Servisno inžinjerstvo

- Je proces razvijanja servisa
  za ponovnu upotrebu
  u servisno orjentisanim aplikacijama
- Inžinjeri treba da osiguraju da servis predstavlja
  ponovo upotrebljivu aprstakciju
  koja može biti korišćena u različitim sistemima.
- Postoje tri logičke faze u razvoju servisa:
  - Identifikacija kadidata
  - Dizajn servisa
  - Implementacija servisa

Servisno inžinjerstvo



Identifikacija kandidata

- Osnovna svrha servisno-orjentisanih arhitektura je da podrže poslovne procese
- Identifikacija kandidata podrazumeva razumevanje I analiziranje poslovnih procesa radi odlučivanja koje servise je potrebno implementirati

Tipovi servisa

- Korisni servisi
  - predstavljaju neku generalnu funkcionalnost koja se koristi u više različitih poslovnih procesa
  - Dobar primer je Konverzija valute novca
- Poslovni servisi
  - oni su povezani sa specifičnim poslovnim funkcijama
  - Npr. Upis studenata na fakultet
- Koordinišući servisi
  - ovo su servisi koji podržavaju poslovne procese koji uključuju različite učesnike I aktivnosti
  - Npr. Naručivanje I plaćanje robe

Servisi orjentisani ka zadacima I entitetima

- Servisi orjentisani ka zadacima
  su oni koji se odnose na neku aktivnost
- Servisi orjentisani ka entitetima
  su slični objektima, odnose se
  na poslovni entitet
  npr. Forma za prijavu za konkurs za posao
- Korisni I poslovni servisi mogu biti
  orjentisani I ka zadacima I ka entitetima
- Koordinišući servisi su uvek
  orjentisani ka zadacima

Tipovi servisa

	Utility	Business	Coordination
Task	Currency converter Employee locator	Validate claim form Check credit rating	Process expense claim Pay external supplier
Entity	Document style checker Web form to XML converter	Expenses form Student application form	

Moguća pitanja

- Za servise orjentisane ka entitetima
  - Da li je servis povezan sa jednim logičkim entitetom koji se koristi u više poslovnih procesa?
  - Koje operacije se izvode nad tim entitetom, a servis treba da ih podrži?
- Za servise orjentisane ka zadacima
  - Da li je zadatak predstavlja funkcionalnost koja koristi različitim ljudima u organizaciji?
  - Da li su oni spremni da prihvate novine koje će servis doneti?
- Da li je servis nezavisan?
- Da li servis treba da čuva stanje? Servisi ne čuvaju interna stanja, ukoliko nam je to potrebno, treba koristiti bazu podataka
- Da li servis može da se koristi I van organizacije?
- Da li različiti korisnici imaju različite nefunkcionalne zahteve? Ako je tako, možda je potrebno implementirati više od jedne verzije servisa

Dizajn interfejsa servisa

Dizajn interfejsa

- Nakon što smo izabrali kandidate za servise, naredni korak je dizajniranje interfejsa servisa
- To uključuje definisanje operacija koje servis podržava, njihovih parametara kao I poruka koje razmenjuju
- Cilj nam je da smanjimo broj poruka što više možemo

Faze dizajniranja

- Dizajn logičkog interfejsa
  - Identifikujemo operacije
    kao I njihove ulaze, izlaze i izuzetke
- Dizajn poruka
  - Opisujemo strukturu poruka
    koje servis šalje I prima
- WSDL implementacija
  - Prevodjenje gorenavedenog dizajna
    u apstraktan opis interfejsa pisan u WSDL-u

Implementacija servisa

Implementacija

- Nakon što smo identifikovali kandidate i dizajnirali njihove interfejse, poslednja faza u razvijanju servisa je njihova implementacija
- Implementacija se može realizovati koristeći standardne programske jezike kao sto su Java i C#
- Oba jezika imaju obimne biblioteke za razvijanje servisa

Testiranje

- Pre nego što servise pustimo u rad, moramo ih testirati
- To obuhvata proveru očekivanih izlaza za date ulaze
- Potrebno je generisati izuzetke da bi se proverilo ponašanje servisa sa nevalidnim ulazima
- Postoje mnogi alati koji pomažu programeru prilikom testiranja

Web servisi

- Ukoliko želimo
  da servis bude javno dostupan
  potrebno je obezbediti
  dodatnu dokumentaciju (opis servisa)
- Ova dokumentacija sadrži
  informacije o servisu
  i pomaze korisnicima da odluče
  koji servis im je potreban
  i da li mu mogu verovati

Opis servisa

- Informacije koje treba da sadrži opis servisa su:
  - informacije o vašem poslu, kontakt, itd.
  - neformalni opis funkcionalnosti koju pruža servis
  - detaljan opis interfejsa, tipova i semantike
  - informacije koje omogućavaju korisnicima da se registruju i dobijaju najnovije informacije o servisu

Nadograđeni servisi

- Nasleđeni sistemi su stari softverski sistemi koje organizacija koristi
- Uglavnom su razvijani na zastarelim tehnologijama ali su ključni u poslu
- Zamena ovakvih sistema uglavnom nije isplativa i mnoge organizacije bi želele da ih iskoriste u saradnji sa novim modernim sistemima
- Nadogradjeni sistemi treba da predstavljaju omotač za pristup sistemskim funkcionalnostima i podacima

Razvijanje softvera pomoću servisa

Razvoj softvera

- Osnovna ideja prilikom razvoja softvera pomoću servisa je da pravimo servise koji će nam koristiti pri implementaciji novih, kombinovanih servisa
- Mnoge kompanije svoje poslovne aplikacije prebacuju na servisno orjentisane sisteme kod kojih je osnovna gradivna jedinica aplikacije servis a ne komponenta
- Krajnja realizacija se oslanja na implementaciji 'servise market'-a koji se sastoji od servisa koji su kupljeni od dobavljača

Poces konstrukcije servisa

- Formulacija opšte šeme procesa
- Prepoznavanje servisa
- Izbor mogućih servisa
- Profinjenje toka procesa
- Izrada programa
  koji predstavlja tok procesa
- Testiran servis ili aplikacija

Dizajn toka procesa

- Dizajn obuhvata analizu
  postojećeg i planiranog poslovnog procesa
  kao i razumevanje aktivnosti
  i njihovih interakcija
- Tok poslovnog procesa se uglavnom prikazuje
  koristeći grafičku notaciju
  kao sto su UML dijagrami aktivnosti
  ili BPMN dijagrami
- Obe tehnike su jednostavne
  i lake za razumevanje

Implementacija

- Nakon što smo završili sa dizajnom, on se treba prebaciti u izvršni program. To uključuje dve aktivnosti:
  - Implementacija servisa koji nisu dostupni za ponovno korišćenje
  - Izgradnja izvršne verzije modela toka procesa.
    - Ovo uključuje prebacivanje modela u WS-BPEL, ručno ili automatski
    - Postoje razni alati koji nam pomažu prilikom tog prebacivanja

KRAJ

Hvala na pažnji!

Branislava Živković 1040/2014