

AGILNI POSLOVNI SISTEMI U SERVISNO ORIJENTISANOJ ARHITEKTURI AGILE BUSINESS SYSTEMS IN SOA

Ardeljan Emanuel

REZIME: Opisana je servisno orijentisana arhitektura za projektovanje agilnih informacionih sistema, kao i standardi koji utiču na arhitekturu i strategije implementacije. Prikazane su dominantne platforme za razvoj. Dat je i jedan ilustrativni primer, koji je implementiran na Microsoft-ovoj platformi.

KLJUČNE REČI: Servisno orijentisana arhitektura, komunikaciona fondacija, procesna fondacija, arhitektura sistema, veb servisi, ekstenzije web servisa, .NET, JAVA.

ABSTRACT: It is described service oriented architecture for implementing agile enterprise systems and standards which influence on implementation strategies. There are shown two major platforms for developing service oriented architecture (JAVA and .NET). At the end it is shown case study which is developed on Microsoft .NET platform.

KEY WORDS: Service oriented architecture, communication foundation, workflow foundation, system architecture, web services web service extensions, .NET, JAVA

1. UVOD

Tokom prethodnih decenija potrebe tržišta su diktirale tempo razvoja različitih standarda u okviru razvoja SOA (*Service Oriented Architecture*) tehnologija ali najbitnije su bile meta jezici i web servisi. Danas SOA predstavlja ključnu ulogu u dizajniranju kompleksnih i agilnih informacionih sistema [1-9]. SOA je baziran na standardima i iza tih standarda stoje najuglednije softverske kompanije kao što su IBM, Microsoft, SAP, ORACLE, BEA, EMC kao i mnoge druge.

Posle definisanja specifikacija za web servise [1,10-19], organizacije su shvatile da umesto akomodacije postojećih distribuiranih aplikacija, web servisi mogu postati osnova za arhitekturu distribuiranih aplikacija u organizacijama. SOA je često klasifikovan na različite načine, najčešće zbog tehnologije na kojoj su servisi implementirani [1-9]. Rani model, koji je bio inspirisan standardima za web servise, definiše SOA kao arhitekturu oko tri osnovne komponente: servisni klijent, servisni provajder i registar servisa. Ovaj SOA model je zadržan do danas i svi proizvođači softvera su ga implementirali u njihovim platformama. Mnogi proizvođači već implementiraju nove ideje u njihovim platformama i poznate su kao web servisne specifikacije WS-* (*Web Service Extension*). Ove specifikacije implementiraju različite funkcionalnosti za web servise.

Osnovni cilj ovog rada je prikaz osnovnih temelja SOA arhitekture. Opisani su svi delovi koji sačinjavaju datu arhitekturu, od paterna za razmenu poruka među servisima, ekstenzije koje omogućavaju dalje proširenje arhitekture sa novim funkcionalnostima, kao i njegove strategije za efikasnu implementaciju date arhitekture u poslovnim sistemima. Prikazan je jedan realan poslovni primer, koji je implementiran na .NET platformi. Ovim primerom su predstavljene najbitnije komponente koje sačinjavaju SOA arhitekturu, kao i prednosti Microsoft-ove platforme za razvoj informacionih sistema baziranih na SOA.

2. SERVISNO ORIJENTISANA ARHITEKTURA

Današnji informacioni sistemi su uglavnom izolovani sistemi, ali poslovni procesi su veoma kompleksni sa velikom međuzavisnošću prostirući se čak preko više različitih organizacija. Zbog ovih razloga, sve poslovne organizacije trebalo

bi polako da razmišljaju o novim sistemima koji su podložni brzim promenama kao i mogućnost komunikacije i integracije sa drugim sistemima interno u organizaciji ili van nje.

Da bi se bolje razumeli problemi poslovnih informacionih sistema, potrebno je sagledati njihove ključne karakteristike koje ih razlikuju od običnih desktop aplikacija, naučnih aplikacija ili čak video igrica. Poslovni informacioni sistemi su strogo vezani za interne poslovne procese i modele jedne poslovne organizacije, gde različiti sektori u organizaciji koriste informacije iz njenih drugih sektora. Da bi se povećala efikasnost i brzo reagovanje na promene sistema, arhitektura informacionih sistema treba da zadovoljava sledeće karakteristike [2]:

- Jednostavnost (arhitektura treba da bude jednostavna da bi se postigla efikasna komunikacija između različitih radnika koji rade na implementaciji i održavanju sistema, kao što su analitičari, programeri, menadžeri).
- Fleksibilnost i lakoća održavanja (svaki sistem je podložan izmenama i zato treba brzo da reaguje na sve izmene u poslovanju).
- Iskorišćenost (delovi sistema treba da se iskoriste u različitim drugim sistemima gde je potrebna ista funkcionalnost).
- Razdvojenost funkcionalnosti i tehnologije (arhitektura sistema treba da razdvoji poslovne procese od tehnologije koja se stalno menja).

SOA je arhitektura informacionih sistema bazirana na servisima gde svaki servis enkapsulira neku poslovnu logiku. Servis predstavlja individualnu logičku jedinicu, koja enkapsulira neki poslovni proces koja je samostalna ali nije izolovana od ostalih servisa. SOA se implementira pomoću web servisa.

SOA ima više prednosti u odnosu na tradicionalno dizajnirane aplikacije. Sve prednosti su zasnovane na realnim zaključcima iz različitih realnih informacionih sistema kao što je SAP, BEA, IBM i drugi. Neke od prednosti SOA arhitekture su [8]:

- Bolja iskorišćenost gotovih servisa – SOA promovise da se svaki servis kreira za mogućnost iskorišćenja na što više mesta u nekom informacionom sistemu. Ovakav način iskorišćenja servisa rezultira u bržem razvoju sistema.

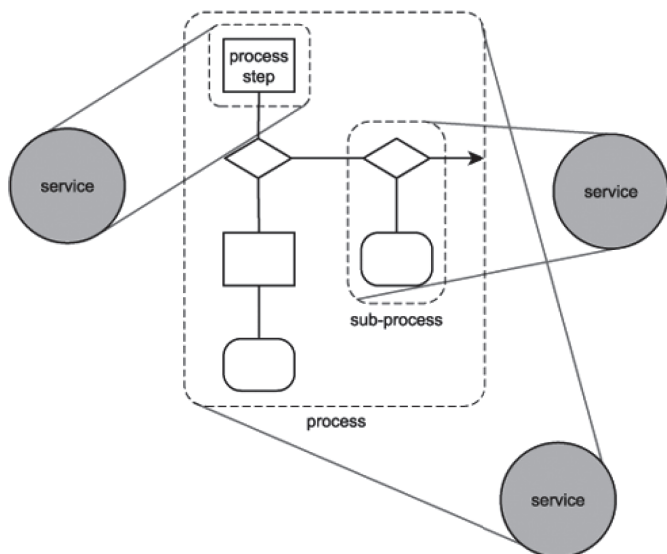
- Bolja automatizacija poslovnih procesa.
- Bolja isplativost na duge staza.
- XML standardizacija podataka – svi podaci se mogu predstavljati preko XML-a čime se smanjuje kompleksnost aplikacija.
- Bolje skaliranje sistema – postojeći sistem može se brže proširiti i rasti.
- Bolja agilnost sistema – informacijski sistem može brže da se adaptira na izmene u postojećim procesima.
- Bolja integracija i interoperabilnost – SOA može kreirati informacijske sisteme koji su nezavisni od platforme.

SOA je bazirana na standardima. Da bi ovi standardi ostali nezavisni od platforme, proizvođači kao što su Microsoft, IBM, Sun Microsystems, SAP počeli su da ih ugrađuju u njihove platforme za razvoj informacijskih sistema. Glavne institucije koje utiču na SOA su: W3C [16], OASIS [20], Web Service Interoperability Organization (WS-I) [15]. Ovi standardi su razvijeni tako da nijedna organizacija nema kontrolu nad razvojem. Oni pružaju otvorene standarde od kojih nastaju nezavisni protokoli za komunikaciju.

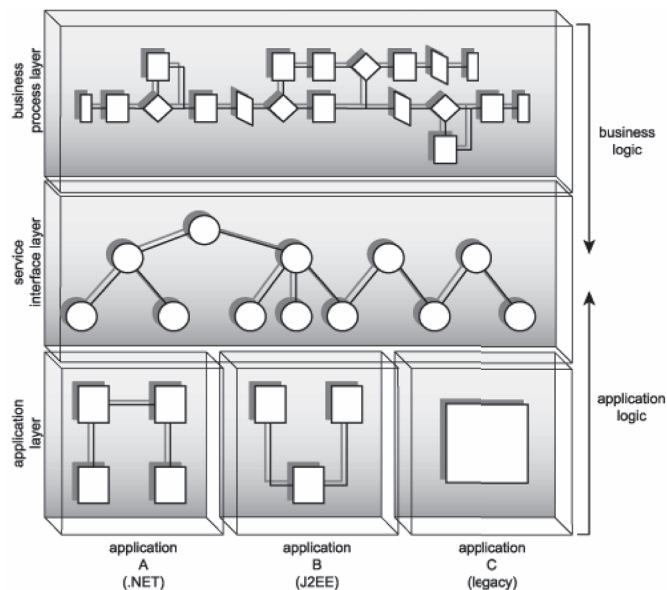
3. LOGIČKA STRUKTURA I PRINCIPI SOA

Da bi servisi održavali svoju nezavisnost, oni enkapsuliraju logiku u posebnom kontekstu. Ovaj kontekst, može biti specifičan za neki poslovni proces, poslovni entitet ili neko drugo logičko grupisanje. Veličina logike koju servis enkapsulira može varirati, neki servisi mogu koristiti i logiku drugih servisa kao krajnji rezultat njihove logike. Na slici 1 je prikazan jedan poslovni proces koji sadrži više koraka. Kompletna logika se može razdvojiti u seriju koraka, koji se obrađuju po redosledu. Svaki korak može biti jedan servis koji obrađuje jedan zadatak i neki potproces od nekoliko manjih koraka. Servis može sadržati čak i kompletan poslovni proces, što se prikazuje na slici 1.

U SOA arhitekturi, servis može biti korišćen od strane drugih servisa. Da bi servisi komunicirali, moraju biti svesni jedni drugih i da znaju kakve operacije sadrže, zato svaki



Slika 1. – Poslovni proces definisan pomoću servisa – izvor [7]



Slika 2. – SOA slojevi – izvor [7]

servis ima opis servisa. Opis servisa definiše ime servisa i koje podatke on prima i vraća, odnosno operacije. Sva komunikacija među servisima se obavlja na osnovu poruka koji oni međusobno razmenjuju. Svaki informacijski sistem sadrži logiku koja se može deliti na dva dela: poslovna logika (*business logic*) koja predstavlja poslovne procese u organizaciji i aplikaciona logika (*application logic*) koja je implementacija poslovne logike preko različitih tehnologija.

Servisno orijentisana arhitektura unosi nove koncepte kako će se poslovna i aplikaciona logika predstavljati. Ovi koncepti se realizuju uvođenjem servisa koji predstavlja apstrakciju koja enkapsulira kako poslovnu logiku tako i aplikacionu logiku. Slika 2 prikazuje sloj servisa koji je postavljen između poslovne i aplikacione logike. Svaki servis predstavlja neku logičku celinu koje se može koristiti u svim aplikacijama bez obzira na njegovu implementaciju i platformu.

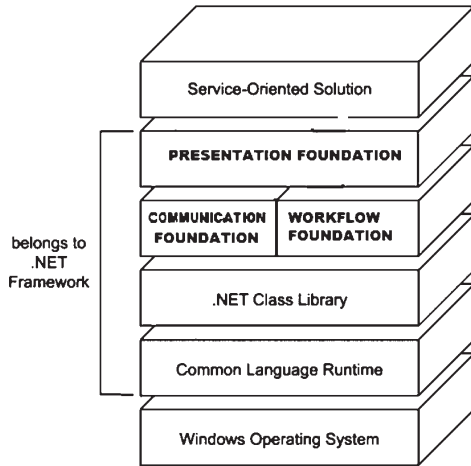
Najčešći skup osobina koje servisi u SOA zadovoljavaju su: iskorišćenost servisa (*reusability*), slaba povezanost (*loose coupled*), servis deli formalni ugovor (*contract*), apstrakcija logike servisa (*abstract underlying logic*), sastavljivost (*composable*), servisi ne čuvaju stanje (*stateless*), servisi su otkrivajući (*discoverable*).

U SOA arhitekturi web servise možemo klasifikovati u sledeće grupe: osnovni servisi (*basic services*), posrednički servisi (*intermediary services*), procesno orijentisani servisi (*process centric services*) i javni servisi (*public enterprise services*).

Detaljniji pregled SAO tehnologija dat je i u radu [21].

4. SAO PLATFORME

Svi koncepti vezani za SOA arhitekturu treba da budu podržani od neke tehnološke platforme da bi omogućili njenu implementaciju i korišćenje. Dve značajnije platforme su J2EE čiji proizvođač je Sun Microsystems [8] i .NET čiji proizvođač je Microsoft [22-26]. Obe platforme omogućavaju kreiranje servisno orijentisane arhitekture pomoću web servisa i podržavaju *WS-I basic profile* standard.



Slika 3. – Windows foundation arhitektura u .NET

Na slici 3 prikazana je .NET platforma. Osnovni segmenti ove platforme su:

- Windows Presentation Foundation (WPF) - bazira se na kreiranju klijentskog interfejsa za desktop ili web aplikaciju. *Presentation Foundation* omogućava kreiranje korisničkog interfejsa koji se bazira na XML za opisivanje komponenti.
- Windows Communication Foundation (WCF) - platforma za kreiranje distribuiranih i konektovanih sistema. WCF je programski model koji omogućava programerima da kreiraju servisna rešenja koja su pouzdana, sigurna i transakciona. WCF omogućava unificiran, simplificiran i održiv pristup razvoju distribuiranih sistemima.
- Windows Workflow Foundation (WWF) - platforma razvijena od strane Microsoft-a da bi omogućila programerima jedinstvenu platformu za razvoj procesno orijentisanih softverskih rešenja. WWF je izgrađen na .NET 3 kao jedna od glavnih komponenti. WWF podržava kreiranje sekvencijalnog i događajnog procesa.

5. STUDIJSKI PRIMER

Primer će biti zasnovan na prikazivanju integracije između *windows communication* što predstavlja sloj osnovnih servisa i *windows workflow*-a odnosno sloja procesnih servisa. Ovakav primer služi za prikazivanje realnih slučajeva SOA arhitekture gde postoje konkretni servisi i konkretan poslovni proces koji se zasniva na datim servisima.

Prikazan je poslovni proces za naručivanje nekog proizvoda za nekog klijenta. Ovakvu vrstu poslovnog procesa neka organizacija može imati na Internetu ili kao internu varijantu kada neki klijent vrši naručivanje telefonom. Kreiranje poslovnog procesa je fokusirano na sloj za osnovne servise i na sloj za procesne servise u SOA.

Proces sadrži sledeće web servise:

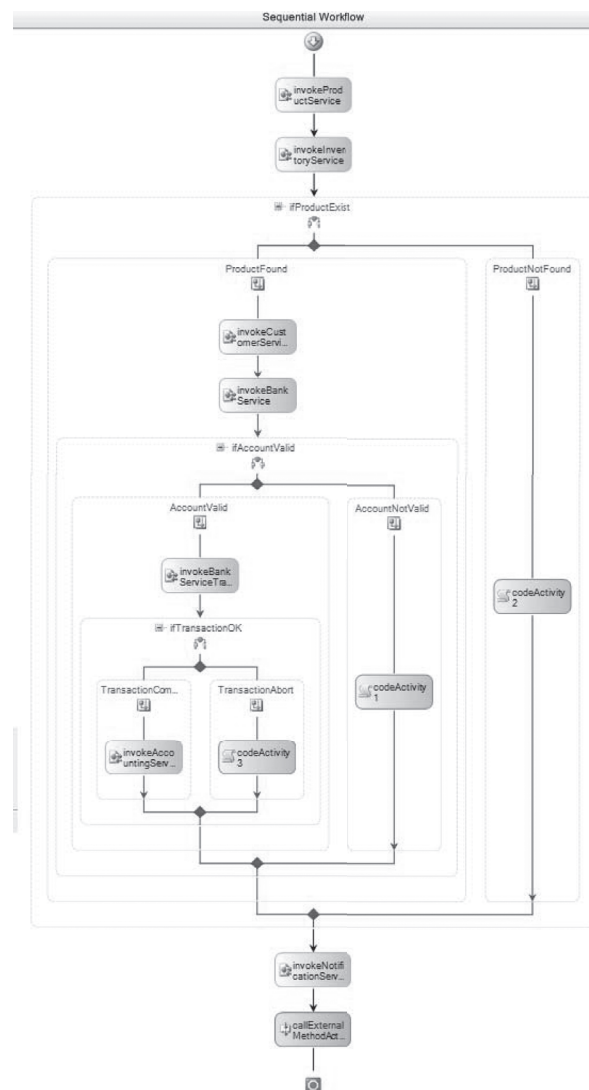
- Product Service – servis služi za manipulaciju proizvodima kao što su: dodavanje novog proizvoda, editovanje, pretraživanje proizvoda i druge operacije koje su vezane za proizvode.
- Customer Service – servis služi za manipulaciju klijentima.
- Bank Service – predstavlja eksterni servis koji služi organizacijama da obave transakcije vezane za njihove račune, proveru validnosti klijenata, e-banking i druge operacije koje banka pruža svojim korisnicima.

- Inventory Service – servis je deo organizacije i služi za vođenje evidencije inventara proizvoda i usluga. Isto omogućava naručivanje novih proizvoda od dobavljača ili obaveštava ako nekog proizvoda nema u zalihama, kao i neke statističke podatke o zalihama.
- Accounting Service – servis služi za dodavanje novih zapisa u glavnu knjigovodstvenu knjigu, kao i druge operacije koje su vezane za knjigovodstvo.
- Notification Service – servis služi za notifikaciju odgovornih lica putem sms-a, e-mail-a ili drugih metoda.

Kreiranje datih web servisa omogućava nam Visual Studio koji služi kao razvojni alat za kreiranje aplikacija. Hostovanje ovih servisa omogućuje IIS 2007 Microsoftov web server.

Slika 4 prikazuje proces za naručivanje proizvoda. Za kreiranje takvog procesa u Visual Studio-u potrebno je napraviti biblioteku za sekvencijalne radne tokove (*workflows*).

Na slici 4 vidi se *workflow* koji ima puno aktivnosti a u njemu postoji grananje u zavisnosti od rezultata. Početak procesa je označen gornjom strelicom odakle počinje a kraj označen donjim krugom. Windows *workflow* ima predefinisane aktivnosti koje omogućavaju kreiranje raznih procesa. Sve ove aktivnosti se mogu naći u alatima Visual Studio-ja.



Slika 4. – Proces za naručivanje proizvoda

Kreiranje orkestracije među web servisima u datom procesu se implementira pomoću *InvokeWebService* aktivnosti. Ova aktivnost se automatski dodaje jednostavnim prevlačenjem na *workflow* i posle traži dodatna podešavanja u parametrima za tu aktivnost kao što su: metoda koja će se pozivati, parametri metode, rezultat i, naravno, url za web servis. Sve ostale aktivnosti se dodaju po redosledu kojim treba sa se izvršava da bi kreirali dati proces.

Scenario procesa za naručivanje proizvoda je sledeće. Proces kao ulazne parametre prima *id* proizvoda koji se naručuje i *id* klijenta koji naručuje dati proizvod. Posle unosa inicijalnih parametara, proces počinje tako što poziva servis *ProductService*, koji uzima detaljnije podatke o datom proizvodu, a zatim zatraži od *InventoryService* da li dati proizvod postoji u magacinu. Ako ne postoji proces će se završiti sa određenom porukom, a ako postoji proces nastavlja dalje. Ovde postoji aktivnost grananja gde u zavisnosti od rezultata *InventoryService*-a se izabere dalja putanja. Ako je operacija bila uspešna onda će *CustomerService* zatražiti detaljnije podatke o klijentu. Kada se dobiju podaci, proverava se validnost kreditne kartice klijenta pomoću *BankService*. *BankService* bio bi u realnosti na serverima banke gde bi postojala SSL komunikacija sa njima. Takođe, postoji grananje u zavisnosti od validnosti kreditne kartice klijenta. Ako kreditna kartica nije validna onda se proces završava sa određenom porukom, a u slučaju da je kreditna kartica validna, onda se proces nastavlja tako što se napravi transakcija u banci gde se sa klijentskog računa skine određena suma novca koja je potrebna da se naplati kupljeni proizvod i stavi na račun organizacije koja prodaje dati proizvod. Ovde postoji grananje u zavisnosti da li je transakcija bila uspešna ili ne. Ako je transakcija uspešno okončana, poziva se *AccountingService* da bi se napravio novi ulaz u glavnu knjigovodstvenu knjigu, i poziva se *NotificationService* koji obaveštava odgovorno lice za naručivanje proizvoda, bilo putem elektronske pošte ili sms mobilne poruke. Pre završetka procesa poziva se klijentska aplikacija i prosleđuju se rezultati izvršenog procesa.

Sekvencijalni proces je kreiran kao biblioteka i može se iskoristiti i u drugim aplikacijama, ili kao potproces drugog procesa.

6. ZAKLJUČAK

Prikazana je servisno orijentisana arhitektura i njene prednosti u razvoju velikih informacionih sistema. Razmatrani su standardi koji omogućavaju pravilnu implementaciju ovakve arhitekture, kao i glavne platforme za razvoj servisno orijentisane arhitekture. Servisno orijentisana arhitektura daje veliku agilnost pri implementaciji poslovnih procesa, kao i mogućnost ilustracije poslovnih procesa što omogućava značajnu komunikaciju među ljudima, koji dizajniraju poslovne procese u organizacijama ili implementiraju date procese. Strategije razvoja informacionih sistema baziranih na servisno orijentisanoj arhitekturi su veoma bitne pri razvoju i implementaciji sistema što doprinosi krajnjem efikasnosti sistema.

Dati ilustrativni primer na .NET platformi prikazuje fleksibilnost i lakoću kreiranje SOA informacionih sistema. Uvođenjem *windows foundation* tehnologije Microsoft otvara vrata novim naprednim mogućnostima u komunikaciji među različitim sistemima.

LITERATURA

- [1] *XSL Transformations (XSTL)* - <http://www.w3.org/TR/xslt>, Decembar 2007.
- [2] Dirk Krafzig, Karl Blanke, Dirk Slama – *Enterprise SOA: Service Oriented Architecture Best Practices*, idavač - Prentice Hall PTR, godina 2004, ISBN 0-13-146575-9.
- [3] James McGover, Oliver Sims, Ashish Jain, Mark Little – *Enterprise Service Oriented Architecture: Concepts, Challenges, Recommendations*, indavač – Springer, godina 2007, ISBN 1-4020-3704.
- [4] Thomas Mattern, Dan Woods – *Enterprise SOA: Design IT for Business Innovation*, izdavač - O'Reilly, godina 2006, ISBN 0-596-10238-0.
- [5] *Service Oriented Architecture* - http://en.wikipedia.org/wiki/Service-oriented_architecture, Decembar 2007.
- [6] *Simple Object Access Protocol (SOAP)* - <http://www.w3.org/TR/soap/>, Decembar 2007.
- [7] Tomas Earl – *Service Oriented Architecture: Concepts, Technologies and Design*, izdavač – Prentice Hall PTR, godina 2005, ISBN 0-13-185858-0.
- [8] Tomas Earl – *Service Oriented Architecture: A Fied Guide to Integrating XML and Web Service*, izdavač – Prentice Hall PTR, godina 2004, ISBN 0-13-142898-5.
- [9] Tomas Earl - *Service Oriented Architecture: Principles of Service Design*, izdavač – Prentice Hall PTR, godina 2007, ISBN 0-13-234482-3.
- [10] *HyperText Markup Language (HTML)* - <http://www.w3.org/TR/REC-html40/>, Decembar 2007.
- [11] *Secuity Assertion Markup Language (SAML)* - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=securitym, Decembar 2007.
- [12] *Standardized Generalized Markup Language (SGLM)* - <http://www.w3.org/MarkUp/SGML/>, Decembar 2007.
- [13] *Universal Description, Discovery and Integration (UDDI)* - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec, Decembar 2007.
- [14] *Web Service Description Language (WSDL)* - <http://www.w3.org/TR/wsdl>, Decembar 2007.
- [15] *Web Service Interoperability Organization (WS-I)* - <http://www.ws-i.org/>, Decembar 2007.
- [16] *World Wide Web Consortium (W3C)* - <http://www.w3.org/>, Decembar 2007.
- [17] *Extensible Access Control Markup Language (XACML)* - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, Decembar 2007.
- [18] *Extedible Markup Language (XML)* - <http://www.w3.org/XML/>, Decembar 2007.
- [19] *XML Schema (XSD)* - <http://www.w3.org/XML/Schema>, Decembar 2007.
- [20] *Organization for the Advancement of Structured Information Standards (OASIS)* - <http://www.oasis-open.org/home/index.php>, Decembar 2007.
- [21] Ardeljan Emanuel, *Projektovanje informacionih sistema baziranih na servisno opijentisanoj arhitekturi*, master rad, Prirodno-matematički fakultet, Novi Sad, 2008.
- [22] Ken Scribner – *Microsoft Windows Workflow Foundation Step by Step*, izdavač – Microsoft Press, godina 2007, ISBN 2006940677. *Microsoft .NET Passport* – www.live.com, Decembar 2007.
- [23] Todd Kitta – *Professional Windows Workflow Foundation*, izdavač – Wrox, godina 2007, ISBN 9780470053867.
- [24] Craig McMurty, Marc Mercuri, Nigel Watling – *Windows Communication Foundation: Hands On*, izdavač – Sams, godina 2006, ISBN 0-672-32877-1.
- [25] Craig McMurty, Marc Mercuri, Nigel Watling, Matt Winkler – *Windows Communication Foundation: Unleashed*, izdavač – Sams, godina 2007, ISBN 0-672-32948-4.



Ardeljan Emanuel
Execom, Kralja Petra I 89 Novi Sad
Oblast: Servisno orijentisana arhitektura,
web servisi, informacioni sistemi, Microsoft
.NET, elektronsko poslovanje.