

KOMPARATIVNA ANALIZA WEB RAZVOJNIH OKVIRA COMPARATIVE ANALYSIS OF WEB FRAMEWORKS

Elena Milovanović, Slađan Babarogić, Nenad Aničić

REZIME: Osnovni cilj ovog rada jeste prikaz načina funkcionisanja samo nekih od najviše korišćenih razvojnih okvira, kao i njihovo međusobno poređenje. Fokus je na okvirima koji se koriste za razvoj Web aplikacija. U ovom radu su prikazani nedostaci i prednosti svakog od odabranih okvira, udobnost u radu, mogućnost ne samo brzog savladavanja okvira, već i brzog razvoja Web aplikacija. Za evaluaciju je odabran podsistem javnog preduzeća u Beogradu koji je zadužen za javne nabavke. U radu je data specifikacija tog podsistema, a za potrebe ovog istraživanja, fokusirali smo se samo na dva reprezentativna slučaja korišćenja, Evidentiranje ugovora i Evidentiranje zapisnika o prijemu robe, koja su realizovana pomoću odabranih razvojnih okvira. Prvi slučaj je lakši slučaj korišćenja, dok drugi slučaj spada u grupu složenih slučajeva korišćenja, jer sama specifikacija opisuje složenu dinamiku.

KLJUČNE REČI: Razvojni okvir, Web aplikacija, MVC patern, ugovaranje javne nabavke

ABSTRACT: The prime objective of this paper is to show the functionality of only some of the commonly used frameworks as well as their mutual comparison. The focus is on frameworks used for Web developed applications. The paper represents the advantages and disadvantages of each of the selected frameworks, working comfort, and possibility of not only quickly overcoming framework but also a fast Web application development. For the evaluation we have chosen subsystem of public corporation in Belgrade, which is responsible for public procurement. This paper provides the specifications of that subsystem, and for the purposes of this research, we will focus on only two representative use cases, contract proceedings as well as recording of goods received which are implemented using these frameworks. The first case falls under simpler use cases where the second is used for more complex cases as the specification itself describes more complex dynamics.

KEY WORDS: Framework, Web application, MVC pattern, contracting public procurement

1. UVOD

U svetu programiranja, softverski okvir je apstrakcija u kojoj se određene, generičke funkcionalnosti mogu promeniti pomoću korisnikovog koda, pri čemu se na taj način obezbeđuje specifičan softver. Ovaj okvir zapravo predstavlja kolekciju softverskih biblioteka koja definiše Application programming interface, u daljem tekstu API. Ukratko, softverski okvir možemo definisati kao skup biblioteka ili klasa koje se mogu upotrebljavati za neki softverski sistem (ili podsistem). Cilj dizajnera ovih okvira je da pojednostave razvoj softvera tako što će omogućiti projektantima i programerima da svoje vreme posvete ispunjenju specifičnih zahteva, pri tom se ne baveći standardnim detaljima na nižim nivoima koji omogućavaju rad samog sistema, što dovodi do smanjenja ukupnog vremena za razvoj softvera.

Svrha razvojnog okvira je da poboljša efikasnost izrade novog softvera. Oni mogu da poboljšaju produktivnost projekatata kao i kvalitet, pouzdanost i robustnost novog softvera. Produktivnost projekatata se povećava tako što im se omogućava da se fokusiraju isključivo na jedinstvene zahteve aplikacije, a da pri tom ne troše vreme na njenu strukturu [3].

Kada je reč o prednostima i nedostacima korišćenja okvira možemo istaći sledeće. Neke od prednosti korišćenja okvira su: mogućnost ponovnog korišćenja koda, uspostavljanje bolje programerske prakse, kao i razna sredstva koja vam okviri stavljaju na raspolaganje kako biste mogli da nadogradite njegovo ponašanje. Što se tiče nedostataka, veliki problem predstavlja to što je za izgradnju jednog okvira potrebno

puno vremena, truda i novčanih sredstava, kriva učenja može biti jako strma, a veliki problem je i to što okvir vremenom može postati jako komplikovan za korišćenje.

Postoji više tipova softverskih okvira, a u ovom radu stavljen je akcenat na aplikacione okvire, pa ćemo stoga njih i podrobnije objasniti.

1.1 Aplikacioni okviri

Aplikacioni okvir je namenjen programerima kako bi implementirali standardnu strukturu aplikacije za specifično razvojno okruženje, kao što je operativni sistem ili Web aplikacija. Ovi okviri se obično fokusiraju na detalje nižeg nivoa, oslobađajući programera od detaljnog poznavanja operativnog sistema i tehničkih detalja platforme [3].

Aplikacioni okviri su postali popularni sa porastom korišćenja grafičkog korisničkog interfejsa (eng. Graphical user interface, u daljem tekstu GUI), jer su oni težili standardizaciji strukture aplikacija. Programerima je daleko lakše da naprave interfejs ukoliko koriste standardni okvir, koji definiše osnovnu strukturu aplikacije. Programeri obično koriste objektno orijentisane programerske tehnike kako bi implementirali okvir tako da se delovi, koji su specifični za datu aplikaciju, mogu jednostavno naslediti iz već postojećih klasa razvojnog okvira. U daljem tekstu ćemo navesti neke primere aplikacionih okvira:

- Apple je razvio jedan od prvih komercijalnih aplikacionih okvira. To je bio MacApp koji je izdat davne 1985. godine, a bio je dizajniran za Macintosh. Napisan je u

proširenoj, objektno orijentisanoj verziji jezika Pascal, a kasnije se pojavila verzija napisana u jeziku C++. Takođe popularni okviri razvijeni za Mac su: PowerPlant i Mac-Zoop, koji se baziraju na jeziku Carbon. Možemo pomenuti i razvojni okvir Cocoa, koji je razijen za Mac OS X i koji nudi drugačiji pristup aplikacionim okvirima.

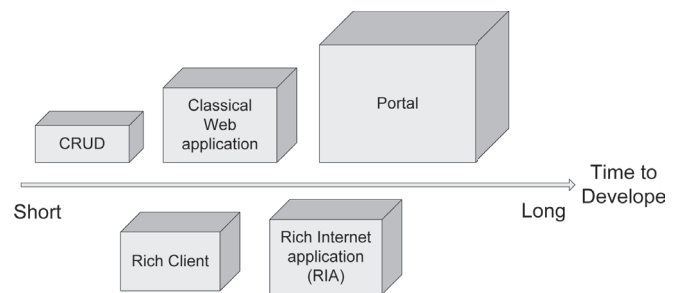
- Besplatni softverski okviri postoje u okviru Mozilla, OpenOffice.org, GNOME, KDE i Eclipse projekata.
- Microsoft je razvio svoj okvir za razvoj Windows aplikacija u jeziku C++, koji je poznat pod nazivom Microsoft Foundation Classes ili MFC.
- Mali broj okvira može omogućiti razvoj aplikacija za više operativnih sistema, na primer za Linux, Macintosh i Windows, i to bez promene izvornog koda. Primeri takvih okvira su: Qt, FOX alat, widget toolkits wxWidgets ili Eclipse RCP.
- Oracle okvir za razvoj aplikacija (Oracle ADF) je okrenut kreiranju Java orijentisanih sistema.

1.2 Web aplikacioni okviri

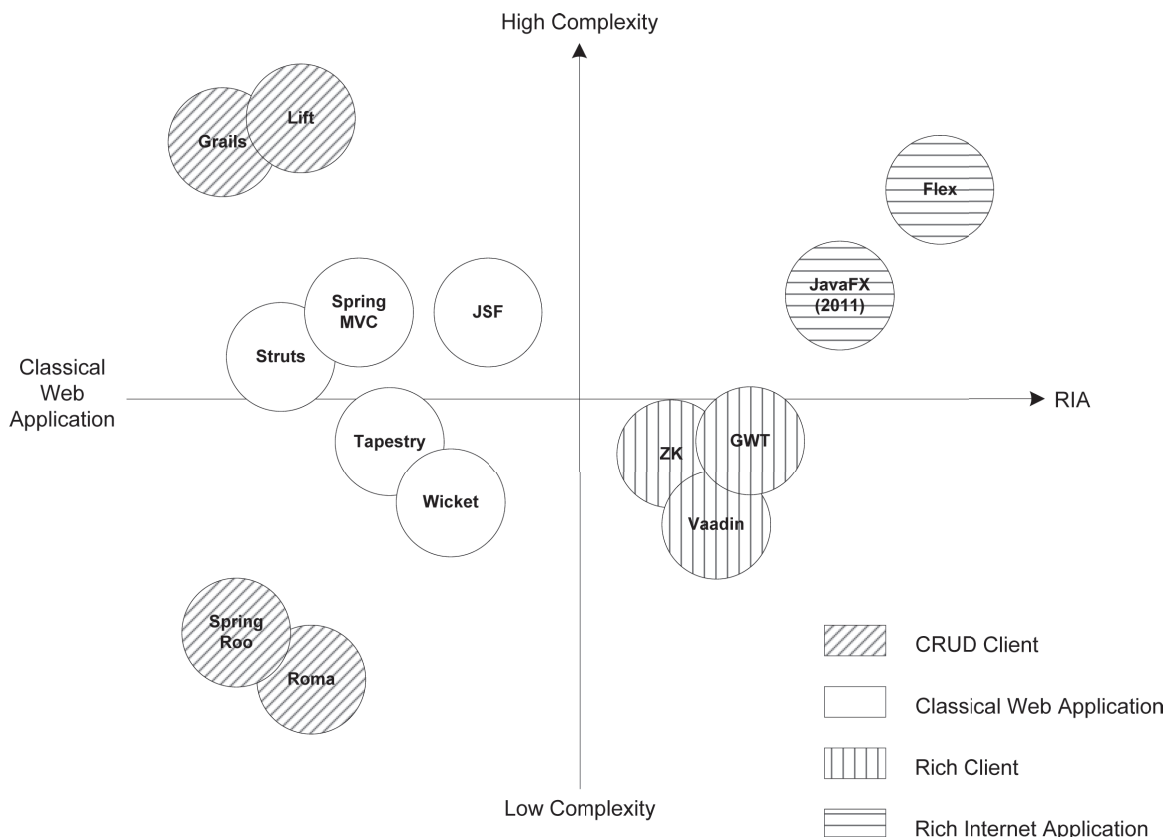
Web aplikacioni okviri nalaze svoje mesto negde između aplikacionih okvira i okvira namenjenih razvoju arhitekture preduzeća. Oni utiču na celokupni dizajn arhitekture, kao što to rade i okviri za arhitekturu preduzeća, ali takođe predstavljaju proširenje aplikacionih okvira, samo što se za razliku od

njih fokusiraju na razvoj aplikacija na Internetu. Kako se ovi okviri koriste za pravljenje Web aplikacija, pre svega ćemo objasniti šta su to Web aplikacije i koje vrste postoje.

Web aplikaciji se pristupa preko mreže, kao što je Internet ili intranet. Termin takođe može da se odnosi na kompjutersku softversku aplikaciju koja je smeštena u okruženju koje kontroliše pretraživač (npr. Java applet) ili je kodirana u jeziku koji podržava pretraživač (npr. JavaScript u kombinaciji sa npr. HTML-om). Web aplikacije su postale popularne zbog zastupljenosti na pretraživačima i zbog pogodnosti da se Web pretraživač upotrebi kao klijent, koji se u ovom slučaju često naziva "tanki" klijent. Ključ njihove popularnosti, jeste sposobnost ažuriranja i održavanja Web aplikacija, čak na hiljadama klijentskih računara, bez distribuiranja i instaliranja softvera, kao i kompatibilnost sa različitim platformama. Na slici 1. (prilagođena prema [7]) prikazani su tipovi Web aplikacija.



Slika 1. – Tipovi Web aplikacija



Slika 2. – Kategorizacija Java / JVM Web okvira

Okvir za razvoj Web aplikacija (u daljem tekstu samo okvir) je softverski okvir koji je dizajniran tako da omogućiti razvoj dinamičkih Web sajtova, Web aplikacija i Web servisa. Okvir ima za cilj da smanji dodatno trošenje resursa za aktivnosti koje su zajedničke u Web razvoju. Zato, na primer, mnogi okviri obezbeđuju biblioteke za pristup bazama podataka, pravljenje templejta i upravljanje sesijama.

Što se tiče arhitekture, najveći broj okvira se zasniva na MVC paternu. Sa aspekta arhitekture, razlikujemo 5 glavnih tipova: tip koji je zasnovan na zahtevu, tip zasnovan na komponenti, hibridni, meta i RIA (rich Internet application) tip. Mnogi okviri primenjuju model-pogled-kontrolor patern (eng. Model-view-controller), u daljem tekstu MVC, kako bi se pomoću poslovnih pravila omogućilo odvajanje modela podataka od korisničkog interfejsa. Možemo reći da je ovo dobra praksa, jer se postiže modularizacija koda, omogućava se njegovo višestruko korišćenje, kao i korišćenje više interfejsa. MVC je softverska arhitektura, odnosno možemo reći da je to arhitekturni patern koji se koristi u softverskom inženjerstvu. Patern izdvaja osnovnu logiku (aplikaciona logika za korisnika) od korisničkog interfejsa i na taj način omogućava nezavisan razvoj, testiranje i održavanje softvera.

2. PRIKAZ I ANALIZA ODABRANIH RAZVOJNIH OKVIRA ZASNOVANIH NA PROGRAMSKOM JEZIKU JAVA

U daljem tekstu sledi obrada odabranih, postojećih okvira koji se zasnivaju na programskom jeziku Java. Na slici 2. (prilagođena prema [7]) je prikazana kategorizacija najrasprostranjenijih i najpopularnijih Web okvira razvijenih u Java okruženju.

U toku istraživačkog rada pronađen je veliki broj pristupa izboru najboljeg okvira, a mi ćemo predstaviti samo jedan od njih. Autori izabranog pristupa definišu sledeće kriterijume:

- Popularnost – Koliko ima pretraga na Google-u? Koliko poslova ima na sajtu Dice.com?
- Aktivnosti za zajednicu – Kada je objavljena poslednja verzija? Koliko je aktivna mejling lista?
- Mogućnost učenja – Koliko postoji dostupnih knjiga? Ko je pisao dokumentaciju?
- Arhitektura – Koliko su rezultati “čisti”? Da li ih dobro skalira?
- Brzi razvoj aplikacija (eng. rapid application development – RAD)
- Bogata Internet aplikacija (eng. rich Internet application – RIA)

Odabir najboljeg okvira je stvar subjektivnog doživljaja, a naravno, moguće je istovremeno koristiti sve gore navedene kriterijume. Tabela na slici 3. dobijena je na osnovu istraživanja autora na Internetu i zasniva se na numeričkom rangiranju okvira [11]. Najniža ocena označava najbolji okvir po određenom kriterijumu.

	Struts	Spring MVC	Tapestry	JSF	Wicket
Popularnost	2	3	4	1	5
Aktivnosti za zajednicu	4	6	5	3	2
Mogućnost učenja	2	3	5	1	4
Arhitektura	1	2	1	2	3
RAD	3	2	1	3	2
RIA	2	1	1	3	2
Ukupno	14	17	17	13	18
Ukupna produktivnost	5	3	2	6	4

Slika 3. – Poređenje okvira

Analizom ukupnog rezultata dolazimo do zaključka da su se u praksi najbolje pokazali sledeći okviri: JSF i Struts, dok su Tapestry i Spring MVC u istom rangu. U daljem radu fokusiraćemo se na tri najbolje plasirana okvira po ovom istraživanju – JSF, Struts i Tapestry.

JavaServer Faces (u daljem tekstu JSF) je novi standardni Java okvir za kreiranje Web aplikacija. On pojednostavljuje razvoj tako što obezbeđuje pristup zasnovan na komponenti za kreiranje Java Web korisničkog interfejsa. Akcenat je na razgraničavanju javnosti i Java Web programera. Programerima i Web dizajnerima razvoj aplikacija uz pomoć JSF okvira neće oduzimati previše vremena, jer se kreiranje interfejsa svodi na obično prevlačenje komponenti, dok će sistemski programeri videti koliko moć i fleksibilnost u programiranju daje bogat i snažan JSF interfejs [3]. Budući da je JSF Java standard razvijen od strane Java Community Process – JCP, proizvođači alata za rad sa ovim okvirom su se pobrinuli za jednostavnost njihovog korišćenja, kao i za vizuelno i produktivno razvojno okruženje. JSF obezbeđuje skup standardnih svojstava koja ga čine moćnim standardom u moru dostupnih tehnologija za razvoj Web aplikacija zasnovanih na jeziku Java. JSF okvir ima višeslojnu arhitekturu koja je dizajnirana tako da podeli sistem na različite funkcionalne jedinice: klijent, prezentacija, poslovna logika, integracija i poslovni informacioni sistem (eng. Enterprise Information System - EIS) odnosno sistem za upravljanje bazom podataka [1].

Struts je jedan elegantan, proširiv okvir za kreiranje Web aplikacija u programskom jeziku Java koje se mogu koristiti za potrebe preduzeća. Okvir je dizajniran tako da unapredi razvojni ciklus, počevši od izgradnje, preko primene i održavanja aplikacija tokom vremena. Struts je vodeći okvir za izgradnju Web aplikacija zasnovanih na programskom jeziku Java. Koristeći MVC patern, Struts rešava mnoge probleme vezane za razvoj Web, poslovnih aplikacija sa visokim performansama, koje koriste Java servlete i JSP (eng. JavaServer Pages) [3]. Kada govorimo o arhitekturi, Struts okvir predstavlja standard za razvoj dobro projektovanih Web aplikacija. Ima sledeće osobine [4]:

- Otvoren kod
- Bazira se na MVC paternu i direktno razdvaja 3 nivoa:
 - o **Model:** stanje aplikacije
 - o **Poged:** predstavljanje podataka (JSP, HTML)
 - o **Kontrolor:** rutira tok aplikacije
- Implementira JSP Model 2 arhitekturu
- Skladišti informacije o rutama aplikacije kao i mapu zahteva u samo jednom XML fajlu koji se zove struts-config.xml

Tapestry je Java okvir otvorenog koda za kreiranje Web aplikacija u programskom jeziku Java. Razvijen je od strane udruženja Howard Lewis i deo je Apache projekta Jakarta. Tapestry se udaljava od striktnih aplikacija zasnovanih na Webu i dozvoljava vam da kreirate aplikaciju koja će više ličiti na tradicionalnu. Osnovni cilj Tapestry programera jeste da kreiraju okvir sa visokom produktivnošću, gde nećete morati da pišete nepotrebni, ponavljajući i mehanički kod. Iako aplikacija napravljena pomoću okvira Tapestry izgleda isto iz ugla korisnika, u suštini ona se dosta razlikuje od pravog Modela 2 ili od verzija okvira Struts. Neke od karakteristika ovog razvojnog okvira su: moć programskog jezika Java i visoka produktivnost. Tapestry aplikacija pokreće se onda kada joj korisnik pristupi preko pretraživača, ukazujući na Tapestry aplikacioni servlet. Servlet se ponaša kao univerzalni kontrolor i kreira mehanizam aplikacije, koji zapravo predstavlja objekat okvira koji upravlja korisnikovom interakcijom sa aplikacijom. Za svakog korisnika se kreira instanca mehanizma, koja zastupa tog korisnika. Mehanizam pronalazi specifikaciju i šemu zahtevane strane kako bi utvrdio sadržaj, a zatim te informacije koristi da bi prikazao stranu korisniku. Najveći broj konfiguracionih dokumenata je uskladišten u memoriji, tako da ovaj proces ne zahteva puno resursa, kao što se može učiniti na prvi pogled [3].

3. SPECIFIKACIJA DELA PODSISTEMA KORIŠĆENOG ZA EVALUACIJU

3.1. Opis problema

U ovom delu ćemo detaljnije analizirati okvire odabrane na osnovu poređenja u prethodnom poglavlju. Kroz praktične primere ćemo pokušati da dokažemo zašto su oni najbolji, kakvi su u praksi, koje su njihove prednosti, kao i da otkrijemo da li postoje problemi u radu sa njima i ako postoje koji su i kako se mogu prevazići. Kao primer će nam poslužiti proces javne nabavke u beogradskom javnom preduzeću.

Organizaciona jedinica zadužena za sprovođenje procesa javne nabavke u JP "Elektromreža Srbije" svoju funkciju vrši tako što komunicira sa organizacionom jedinicom zaduženom za planiranje, sa računovodstvom, sa pogonima, kao i sa direktorom. Van organizacije ostvaruje interakciju sa oglašivačima kao što su "Službeni glasnik Srbije" i Portal javnih nabavki, zatim sa drugim pravnim licim koja se javljaju kao ponuđači, kao i sa državnim organom zaduženim za zaštitu ponuđača.

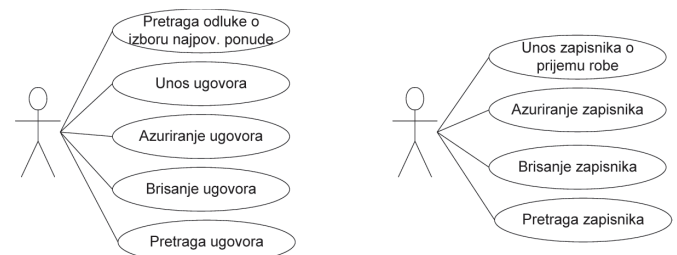
Proces javne nabavke počinje kada u organizacionu jedinicu stigne interna narudžbenica iz nekog od pogona preduzeća. Nakon toga proverava se usklađenost narudžbenice sa planom nabavke koji se dobija iz sektora za planiranje. Ukoliko je narudžbenica u skladu sa planom, šalje se zahtev

za pokretanje postupka javne nabavke generalnom direktoru koji odgovara odlukom o pokretanju postupka javne nabavke i rešenjem kojim imenuje tročlanu komisiju koja je zadužena za sprovođenje konkretne nabavke.

U našem radu pažnju ćemo obratiti samo na neke procese javne nabavke, to su Ugovaranje javne nabavke, tačnije Potpisivanje ugovora, Realizacija i Reklamacija. Sa ponuđačem čija je ponuda najbolja sklapa se ugovor, a oglašivačima se upućuje obaveštenje o dodeli ugovora. Nakon sklapanja ugovora sledi realizacija i eventualna reklamacija ili tužba ukoliko nisu ispunjeni uslovi ugovora od strane ponuđača.

3.2. Specifikacija slučajeva korišćenja Evidentiranje ugovora i Evidentiranje zapisnika o prijemu

Na slici 4. dat je dijagram slučajeva korišćenja podsistema za sprovođenje procesa javne nabavke. Za potrebe evaluacije Web razvojnih okvira izabrana su dva reprezentativna slučaja korišćenja: Evidentiranje ugovora i Evidentiranje zapisnika o prijemu. Prvi slučaj predstavlja po težini lakši slučaj korišćenja i po tom šablonu može se raditi specifikacija aplikacija nad šifarnicima. Drugi slučaj spada u grupu složenih slučajeva korišćenja, jer sama specifikacija opisuje složenu dinamiku.



Slika 4. – Dijagram slučajeva korišćenja

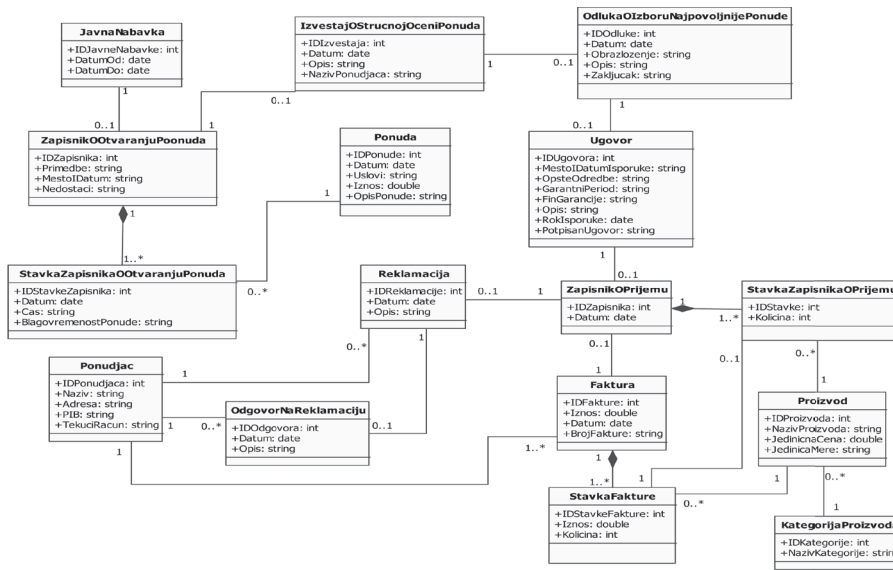
Prvi, lakši slučaj korišćenja služi za evidentiranje ugovora koji su sklopljeni sa ponuđačima u procesu javne nabavke. Na osnovu ponude, na koju se ugovor odnosi, korisnik dalje unosi podatke o ugovoru kao što su: šifra, datum, garantni period, rok isporuke i ostalo. Takođe, postoji mogućnost pretrage ugovora, njihovog brisanja i ažuriranja.

Drugi, teži slučaj korišćenja omogućava evidentiranje zapisnika o prijemu robe. Nakon izbora ugovora na koji se zapisnik odnosi, dalje se unose podaci o samom zapisniku, kao što su: šifra zapisnika, datum, stavke zapisnika i ostalo. I u ovom slučaju korišćenja imamo mogućnost pretrage zapisnika, kao i mogućnost njihovog brisanja i eventualnog ažuriranja. Ono što je bitno da naglasimo, jeste to da su ovo specifični slučajevi korišćenja i da se mogu koristiti za realizaciju drugih, sličnih slučajeva. Na slici 5. dat je konceptualni dijagram klasa za dva izabrana slučaja korišćenja.

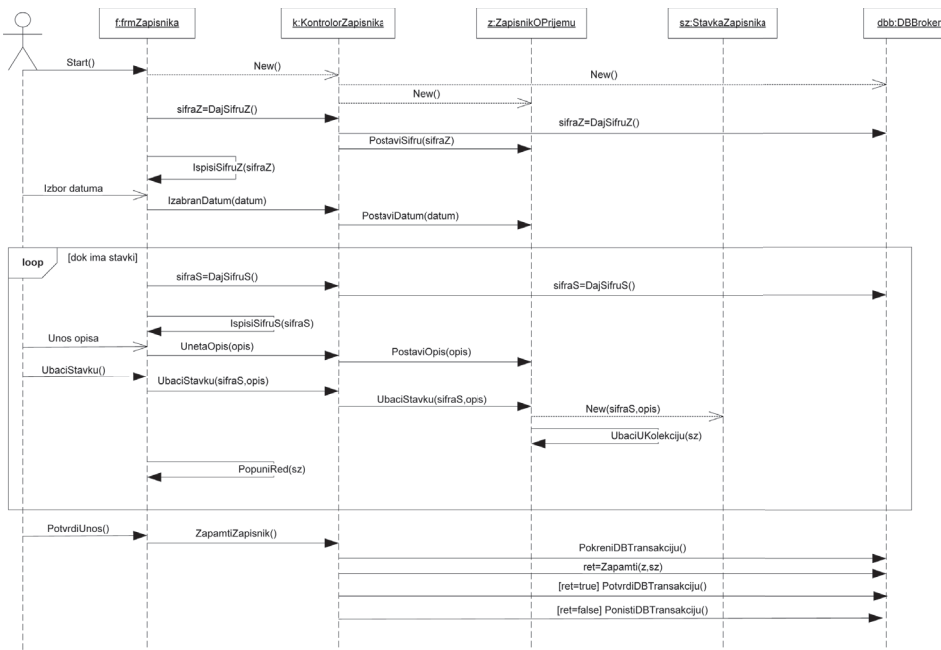
Na slici 6. prikazan je dijagram sekvenci za prvi slučaj korišćenja, a na slici 7. dijagram sekvenci za drugi slučaj korišćenja.

4. ZAKLJUČNA RAZMATRANJA

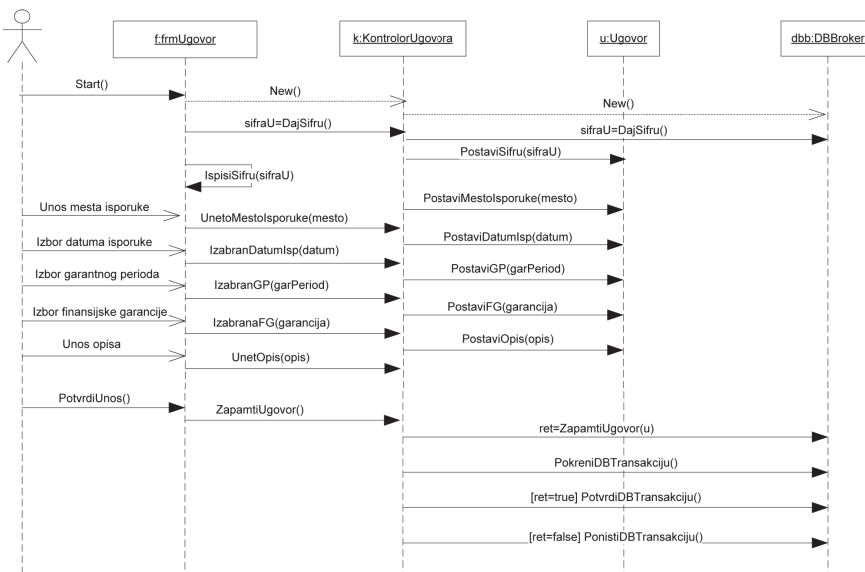
U ovom odeljku ćemo izneti opšti utisak i zaključke do kojih smo došli koristeći ova tri razvojna okvira. Sva tri okvira



Slika 5. – Konceptualni dijagram klasa za izabrane slučajeve korišćenja



Slika 6. – Dijagram sekvenci za prvi slučaj korišćenja – Unos ugovora



Slika 7. – Dijagram sekvenci za drugi slučaj korišćenja – Unos zapisknika o prijemu

služe za kreiranje klasičnih Web aplikacija. Na slici 2. je prikazana kategorizacija Java Web razvojnih okvira. Po toj kategorizaciji JSF razvojni okvir spada u one koji su komplikovaniji za korišćenje, dok su Struts i Tapestry jednostavniji. Međutim pitanja koja se nameću jesu, da li je sve baš tako kako kaže statistika i kakvi su zapravo oni u praktičnoj primeni. Odgovore na ta pitanja pokušali smo da otkrijemo realizacijom odabranih slučajeva korišćenja pomoću tri odabrana okvira.

Što se tiče JSF razvojnog okvira, možemo potvrditi da je komplikovaniji za rad. Potrebno je dosta kodiranja, i morate dobro poznavati kreiranje HTML strana, pokretanje i praćenje sesija i ostale stvari karakteristične za Web programiranje. Međutim da li nam to predstavlja problem? Kada imate dosta dokumentacije, tutorijala i dobro objašnjenih primera, što je slučaj sa ovim okvirom, ništa nije problem. Jako ga je lako savladati zahvaljujući moru tutorijala. Posедуje i paletu za brzo kreiranje strana, tako da vas oslobađa bespotrebnog kodiranja. Tu je i grafička prezentacija faces-config.xml i web.xml fajlova za kreiranje navigacionih pravila, definisanje validacije i kreiranje sesije [8], [9].

Ako se sada osvrnemo na kategorizaciju okvira na slici 2. i na tabelu na slici 3, koja predstavlja početnu tačku ovog istraživanja, možemo zaključiti da su polazne pretpostavke u potpunosti tačne. JSF zaista spada u okvire koji su komplikovaniji za rad, ali isto tako možemo reći da je opravdao prvo mesto koje je zauzeo po popularnosti i po mogućnosti učenja. Takođe, ako obratimo pažnju na karakteristiku aktivnost za zajednicu i uporedimo ocene koje je svaki od odabranih okvira dobio za ispunjenje te karakteristike, vidimo da je JSF dobio najvišu ocenu, što je sasvim opravdano s obzirom da je poslednja stabilna verzija ovog okvira izdata 9. septembra 2011. godine.

Što se tiče Struts razvojnog okvira, jednostavniji je od JSF okvira. Postoji mogućnost instaliranja Struts konzole, što znatno olakšava korišćenje, a tu je i dodavanje blanko projekta u vaš projekat, koji pravi strukturu i ubacuje sve potrebne biblioteke koje su potrebne za korišćenje Struts okvira. Postoji dosta primera i kvalitetnih tutorijala koji olakšavaju savladavanje ovog razvojnog okvira [5], [6].

Poređenjem zaključaka, do kojih smo došli u radu sa ovim okvirom, sa polazni podacima, možemo potvrditi sve pretpostavke. Struts je zaista jednostavniji za korišćenje od JSF okvira, a što se tiče ispunjenja karakteristika, možemo reći da je iskustvo u radu sa ovim okvirom u skladu sa ocenama koje je dobio.

Opšti utisak o Tapestry razvojnog okviru nije baš najsajjniji. Okvir je jako jednostavan za korišćenje. Zahteva jako malo kodiranja i osnovno poznavanje HTML strana. Međutim korišćenje ovog okvira je najmanji problem. Najveći problem predstavlja uključivanje mogućnosti Tapestry okvira u odabrano razvojno okruženje, nalaženje odgovarajućih i dobro objašnjenih primera, kao i kvalitetnih tutorijala. Materijala za Tapestry ima jako malo, tako da ako uspete da savladate taj početni problem, uživaćete u radu sa ovim razvojnim okvirom [10].

Što se tiče Tapestry okvira i iskustva u radu sa njim, dolazimo do zaključka da su ocene koje je dobio za svaku od karakteristika u potpunosti opravdane i da je s razlogom najslabije rangiran od tri okvira koja smo odabrali.

Iz prethodnog možemo zaključiti da ćemo nailaziti na probleme koji god razvojni okvir odabrali, a na nama je da odlučimo

da li želimo lako savladavanje sa više kodiranja, ili teže savladavanje sa minimumom kodiranja i minimumom poznavanja Web programiranja.

Cilj ovog rada je bio da objasnimo princip rada samo nekih od mnoštva razvojnih okvira. Da bismo otkrili sve funkcionalnosti koje nam pružaju razvojni okviri potrebno je dosta vremena. Ovim radom mi smo samo zagrebali po površini, a pravo obilje mogućnosti nas tek očekuje.

5. LITERATURA

- [1] Zambon G., Sekler M., "Beginning JSPTM, JSFTM, and Tomcat Web Development: From Novice to Professional", Apress Inc., 2007.
- [2] Raible M., "Comparing Java Web frameworks", Raible Designs Inc., 2007.
- [3] Ford N., "Art of Java Web Development", Manning, 2004.
- [4] Holmes J., "Struts: The Complete Reference, Second Edition", McGraw-Hill/Osborne, 2007.
- [5] Doray A., "Beginning Apache Struts From Novice to Professional", Apress, Inc., 2006.
- [6] Spielman S., "The Struts Framework Practical Guide for Java Programmers", Morgan Kaufmann Publishers, 2003.
- [7] "Categorization and Comparison of Popular Web Frameworks in the Java / JVM Environment", članak sa Interneta, java.dzone.com, preuzeto juna 2011.
- [8] "First Struts Tutorial (free tools, Struts Console, JBoss IDE)", članak sa Interneta, www.laliluna.de, preuzeto juna 2011.
- [9] "JSF (JavaServer Faces) Tutorial", članak sa Interneta, www.vogella.de, preuzeto jula 2011.
- [10] "Step-by-Step Jakarta Tapestry", članak sa Interneta, www.sitepoint.com, preuzeto jula 2011.
- [11] "Choosing the Right Java Web Development Framework", članak sa Interneta, olex.openlogic.com, preuzeto juna 2011.
- [12] Nešković S., Živković S., "Implementacija Web orijentisanog korisničkog interfejsa sistema za upravljanje dokumentima primenom modelom vođene arhitekture", časopis INFO M god 3. sv. 10, str. 32-37, Beograd, 2004.



Elena Milovanović, student master studija, Smer za informacione sisteme, Fakultet organizacionih nauka Univerziteta u Beogradu
Kontakt: milovanovic.elena@yahoo.com
Oblasti interesovanja: baze podataka, projektovanje IS, BPMN, Web razvojni okviri, XML



Slađan Babarogić, Fakultet organizacionih nauka Univerziteta u Beogradu
Kontakt: babarogic.sladjan@fon.bg.ac.rs
Oblasti interesovanja: Projektovanje informacionih sistema, Modelovanje poslovnih procesa, Workflow i sistemi za upravljanje dokumentima



Nenad Aničić, Fakultet organizacionih nauka Univerziteta u Beogradu
Kontakt: anicic.nenad@fon.bg.ac.rs
Oblasti interesovanja: Razvoj informacionih sistema, baze podataka, integracija i interoperabilnost poslovnih sistema