

Sekcija 3

INFORMATIKA

Koncept sistema u inženjerstvu

Concept of system in engineering

Milenko Heleta, Radoslav Avlijaš, Goran Avlijaš, Tehnički fakultet, Univerzitet Singidunum

Apstrakt - U radu se prikazuje koncept sistema koji se koristi u inženjerstvu za razvoj i realizaciju složenih proizvoda. Složeni proizvodi se posmatraju kao sistemi. Pošto je za njihovu realizaciju neophodno multidisciplinarno znanje neophodno je obezbediti primenu sistemskih metoda, alata, tehnika i softvera u tehničkim procesima u kojima se oni realizuju. Prikazana je uloga standardizacije menadžment sistema koji treba da omoguće koordinaciju tehničkih i upravljačkih procesa primenom projektnog menadžmenta i da obezbede održivi razvoj novih proizvoda i održivu uspešnost organizacije.

Ključne reči –inženjerstvo, sistem, održivi razvoj.

Abstract - This paper presents the concept of system that is used in engineering for the development and implementation of complex products. Complex products are observed as systems. Since their implementation requires multidisciplinary knowledge, it is necessary to ensure the implementation of systematic method, tools, techniques and software in technical processes in which they are delivered. Paper illustrate the role of standardization of management system which facilitate coordination of technical and management processes, using the project management and ensure the sustainable development of new products and the sustainable success of the organization.

Index terms - engineering, system, sustainable development.

1. UVOD

Održiv ekonomski oporavak jedne države moguć je jedino uz njen privredni oporavak. Dugoročan privredni oporavak može se ostvariti orijentacijom na sopstveni razvoj, koji podrazumeva ovladavanje savremenim tehnologijama u oblastima u kojima postoje raspoloživi resursi. Strana ulaganja, motivisana činjenicom da, pored razvijene infrastrukture, imamo talentovane inženjere i razvijenu industrijsku kulturu radnika, ni izbliza nisu dovoljna za reindustrijalizaciju.

Za prelazak iz pasivne uloge čekanja ulaganja kapitala i tehnologija u izgradnju proizvodnih pogona u cilju zapošljavanja, sve više treba prelaziti u aktivnu ulogu u kome ćemo biti nosioci ideja i tehnologija za nove poduhvate. Postoje veliki problemi za otpočiranje i vođenje biznisa koji istovremene ispunjava zahteve tržišta, strogih zakonskih propisa, ekonomskog rasta, očuvanja životne sredine i društveno odgovornog poslovanja. Pored preduzetničkog duha i veština neophodno je poznavati karakteristike i strukturu samih proizvoda, tehnologiju njihove izrade i kompletnu organizaciju kako internu tako i prema okruženju.

Inženjerstvo, disciplina koja je kod nas skoro zaboravljena, ima ulogu da rešava navedne praktične probleme. Zadatak inženjerstva je da kreativno primenjuje naučna, matematička i tehnička znanja i principe za razvoj, projektovanje, izradu i primenu inventivnih, upotrebljivih i ekonomičnih proizvoda, objekata ili procesa [1].

Milenko Heleta - Univerzitet Singidunum, Danijelova 32, 11000 Beograd, Srbija (e-mail: mheleta@singidunum.ac.rs).

Radoslav Avlijaš - Univerzitet Singidunum, Danijelova 32, 11000 Beograd, Srbija (e-mail: ravlijas@singidunum.ac.rs).

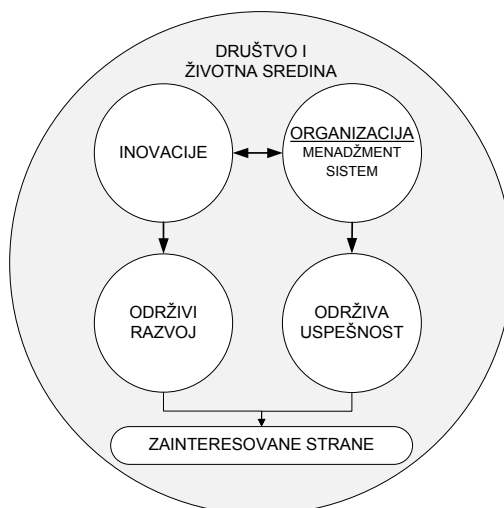
Goran Avlijaš – Univerzitet Singidunum, Danijelova 32, 11000 Beograd, Srbija (e-mail: gavlijas@singidunum.ac.rs).

Sve što je ljudska ruka stvorila oko nas plod je inženjerstva, odnosno jedne ili više inženjerskih disciplina. Danas su industrijski, ali i drugi proizvodi i usluge do te mere složeni, sa integrisanim funkcijama više proizvoda u jedan, da se moraju posmatrati kao složeni sistemi. Ključna reč za razvoj i realizaciju složenih proizvoda je sistem. Koncept sistema omogućuje da se koncipiraju i projektuju složeni proizvodi i procesi kao sistemi, korišćenjem još složenijih sistemskih alata, metoda i tehnologija, uz koordinaciju i upravljanje još složenijim menadžment sistemima.

2. KREIRANJE PREDUZETNIČKOG PODUHVA TA I ODRŽIVOST BIZNISA

Ključni faktori za savlađivanje ekonomske krize i prosperitet svake organizacije su inovacije i dobra organizacija. Potrošači su spremni kupiti samo inovativne proizvode atraktivnog kvaliteta koji ispunjavaju njihove potrebe i skrivena očekivanja. Za deceniju ili dve nijedan industrijski proizvod neće ličiti na današnje. Ciklus razvoja i lansiranja novih proizvoda dramatično se smanjuje, stalno se traže novi i novi proizvodi.

Na slici 1 prikazan je odnos ključnih faktora održivog biznisa i njihove međusobne veze. Od organizacije se zahteva da bude sistemski i inovativno orijentisana. To znači da se generisanje ideja i njihovo pretvaranje u inovacije vrši na organizovan način, a da se strategija njihovog razvoja vrši kroz održivi razvoj koji mora da vodi računa o životnoj sredini.



Slika 1: Ključni faktori održivog biznisa

Sa druge strane menadžment sistem mora da omogući održiv poslovni uspeh zasnovan na uspunjenju zahteva svih zainteresovanih strana – korisnika, vlasnika, zaposlenih, partnera i društva u celini.

Za kreiranje i ostvarenje preduzetničkog poduhvata potrebno je imati ideju i tražiti način na koji se ona može ostvariti. Pet načina da se kreira jedinstven, profitabilan preduzetnički poduhvat na bazi postojeće ideje su [2]:

- razvijanje ideja koje predstavljaju proširenje ili redizajn već postojećeg proizvoda ili usluge;
- resegmentovanje i kreiranje poboljšane proizvoda ili usluge;
- redefinisane i lansiranje proizvoda po nižoj ceni;
- dodavanje vrednosti postojećem proizvodu ili usluzi;
- razvoj ili redizajn nove verzije postojećeg proizvoda.

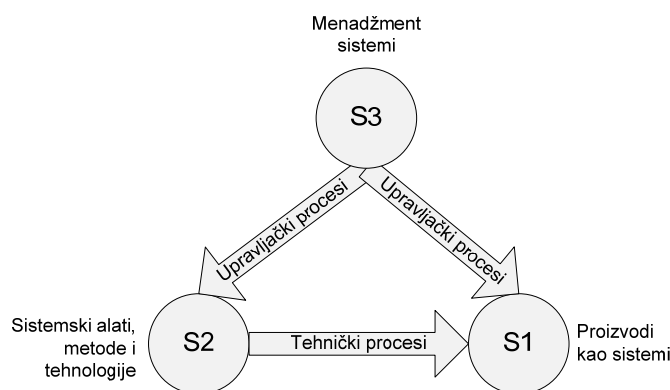
Nakon što preduzetnik dođe na ideju, mora proceniti dostupne mogućnosti za sproviđenje te ideje u praksu. Procenjuje se da oko 85% biznisa u svetu propadne u roku od 5 godina od osnivanja. Određenom broju preduzeća nedostaju investicije koje bi im omogućile da nastave započeti proces (obično 6 meseci ili godinu dana između analize šanse i pokretanja biznisa). Ostala preduzeća se gase zbog lošeg planiranja biznisa u početnim fazama razvoja. Veliki broj preduzeća nestaje zbog nedostatka poslovnih resursa, stručnosti menadžera i jednostavno nedostatka iskustva [2].

Smatra se da veliki broj firmi propadne pri prelazu iz male u srednju organizaciju ako nisu izgradile efektivan menadžment sistem. To je trenutak kada vlasnik firme više nije u stanju da sve konce biznisa

drži u svojoj glavi i postaje usko grlo, što vodi stagnaciji i propadanju firme. Distribucijom ovlašćenja na saradnike, bez izgrađenog sistema, gubi se kontrola nad procesima i resursima i firma takođe vremenom propada.

3. KONCEPT SISTEMA

Koncept sistema, prikazan na slici 2, čini srce mogućnosti organizacije da zamisli, koncipira, i projektuje sve složenije proizvode. Svi složeni proizvodi se moraju strukturirati kao sistemi, kako bi se mogla definisati njihova arhitektura podsistema (podsklopova, delova ili komponenti) i konfiguracija funkcija celog sistema i podfunkcija svakog podsistema [1].



Slika 2: Koncept sistema u inženjerstvu

Da bi se uspešno realizovali složeni proizvodi, moraju biti razvijene i primenjene još složenije systemske metode, alati, tehnologije i softveri koji koriste inženjerske tehničke procese za projektovanje, izgradnju i održavanje složenih proizvoda kao sistema. Za koordinaciju ljudima, aktivnostima i drugim resursima koji koriste složena systemska znanja i iskustva u realizaciji složenih proizvoda kao sistema moraju biti uspostavljeni još složeniji menadžment sistemi koji koriste upravljačke i tehničke procese.

4. SLOŽENI PROIZVODI KAO SISTEMI

Sistem u najširem smislu predstavlja skup povezanih i međusobno delujućih elemenata [3]. Složeni proizvod kao sistem se sastoji od podsistema koji su povezani sa međudejstvom kroz njihove funkcije. Funkcije svakog dela su potčinjenje funkcijama sistema, pošto proizvod kao celina treba da zadovolji određene potrebe.

U tabeli 1 dat je ilustrativno uporedni pregled nekih funkcija, karakteristika i performansi određenih proizvoda kao sistema. Važno je utrditi njihov međusobni odnos kako sledi:

- funkcije sistema određuju koje potrebe sistem zadovoljava i šta može da radi,
- karakteristike sistema predstavljaju osobine ili svojstva njegovih funkcija,
- performanse sistema predstavljaju rezultate koje mogu postići njegove karakteristike.

Prvi korak kod projekata kreiranja i realizacije novih proizvoda je rasčlanjivanje proizvoda na komponente (arhitektura sistema) primenom metoda WBS (Work Break Structure) [4]. Nakon toga odvijaju se paralelne aktivnosti, kako sledi:

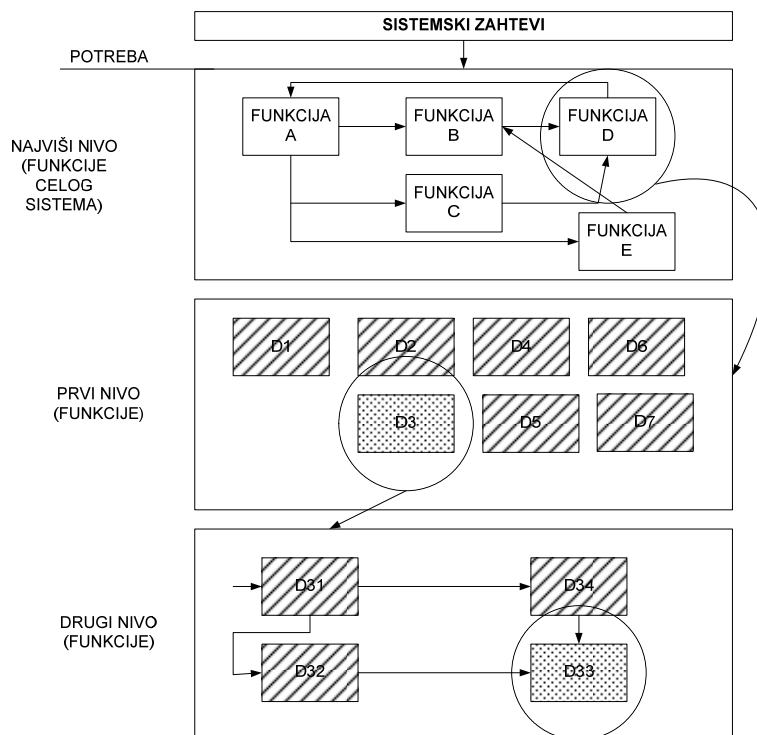
- koncipiranje proizvoda kao sistema, primenom funkcionalnog pristupa i drugih alata i tehnika systemskog inženjeringa,
- izrada detaljnih planova, primenom alata i tehnika projektnog menadžmenta za realizaciju samog poduhvata kao projekta.

TABELA 1

Ilustrativni pregled nekih funkcija, karakteristika i performansi proizvoda kao sistema [1]

Red broj	Sistem / Proizvod	Potreba koju treba zadovoljiti	Funkcija sistema	Karakteristika	Performansa
1.	Hidroelektrana	Snabdevanje električnom energijom	Proizvodnja električne energije	Snaga elektrane	1.000 MW
2.	Autobus	Kolektivni prevoz na srednje daljine	Prevoz putnika	Broj sedišta	80 putnika
3.	Avion	Brzi prevoz na veće daljine	Letenje	Brzina	950 km/h
4.	Foto aparat	Snimanje fotografija koje se mogu memorisati	Digitalno fotografisanje	Rezolucija fotografije	9,0 Mega Pixela
5.	Klima uređaj	Regulisanje temperature u prostorijama	Grejanje i hlađenje prosorija	Površina prostorije koju greje / hladi	60 m ²

Funkcionalni pristup pomaže da se, kod realizacije složenih proizvoda kao sistema, dosledno izvrši hijerarhijski raspored funkcionalnih zahteva. Funkcije su razvijene od najvišeg nivoa, preko podfunkcija do najnižih nivoa, kako je prikazano na slici 3. One mogu biti zavisne ili nezavisne, i predstavljene kao serijske, paralelne ili kombinovane. Na svakom nivou definisane su potrebama koje treba da zadovoljavaju, sa utvrđenim karakteristikama i performansama.



Slika 3: Nivoi funkcija u sistemu [4]

Ilustrativni funkcionalni tok na slici 3 ima najviši nivo funkcija celog sistema i dva niža nivoa. Na najvišem nivou prikazano je pet funkcija A, B, C, D i E, koje se odnose na ceo sistem, sa progresivnim širenjem funkcija D i D3. Funkcija D sa najvišeg nivoa se dekonponuje i širi na sedam funkcija, D1 do D7, na prvom nivou. Funkcija D3 sa prvog nivoa se dekonponuje i širi na četiri funkcije, D31 do D34, na drugom nivou.

Funkcije sistema sadrže karakteristike i performanse za rad sistema u celini, za rad svakog njegovog dela, za održavanje, za uticaj na životnu sredinu i za uticaj na zdravlje i bezbednost na radu. Prednosti funkcionalnog dijagrama toka su:

- mogućnost da se projektovanje, uključivo i izmene, odvija na logičan i efikasan način,
- redosled i odnosi između različitih projektnih grupa u projektovanju su uspostavljeni,
- integriše i koordinira funkcionalne veze koje moraju biti ostvarene u toku projektovanja i
- hijerarhijsko pridruživanje sistemskih funkcija podelementima sistema.

5. SISTEMI KOJI KORISTE TEHNIČKE PROCESSE

Inženjerstvo je disciplina koja koristi tehničke procese za razvoj i primenu vrlo složenih sistemskih metoda, alata, tehnologija i softvera za razvoj, projektovanje, realizaciju i održavanje složenih proizvoda kao sistema. Inženjerske discipline se primenjuje, pored klasičnih inženjering organizacija, i u svim drugim organizacijama koje razvijaju i projektuju sopstvene proizvode ili usluge. Inženjerstvo se primenjuje praktično u svim delatnostima i oblastima (elektrotehnika, mašinstvo, građevinarstvo, razvoj softvera, medicina, finansije, nanotehnologije, ekologija, itd).

Da bi zadovoljilo potrebe razvoja i realizacije složenih sistema, inženjerstvo se razvijalo u dva međusobno komplementarna pravca:

- razvoj pojedinačnih inženjerskih disciplina sa težištem na produbljivanje specijalističkih tehnoloških znanja, metoda, alata i procedura, u cilju njihove primene kod rešavanja pojedinih problema vezanih za sistem i
- razvoj sistemskog inženjeringa koji integriše sistemske metode, alate i procedure, kako u tehničkom smislu vezanih za sam sistem, tako i u upravljačkom smislu, sa ciljem da se izvrši optimalna integracija svih inženjerskih disciplina koji učestvuju u realizaciji projekta.

Sistemski inženjering je koncipiran da bude interdisciplinaran samo u metodološkom, a ne i sadržajnom smislu. Proces sistemskog inženjeringa je vertikalni, iterativan proces za rešavanje problema, prvenstveno u procesu razvoja i projektovanja proizvoda kod bilo koje inženjering discipline. Ovaj proces se koristi se za:

- transformacije potreba i zahteva korisnika u komplet sistemskih specifikacija organizacije po kojima se razvija proizvod,
- generisanje informacija za donošenje odluka i
- obezbeđenje ulaza za sledeći nivo razvoja.

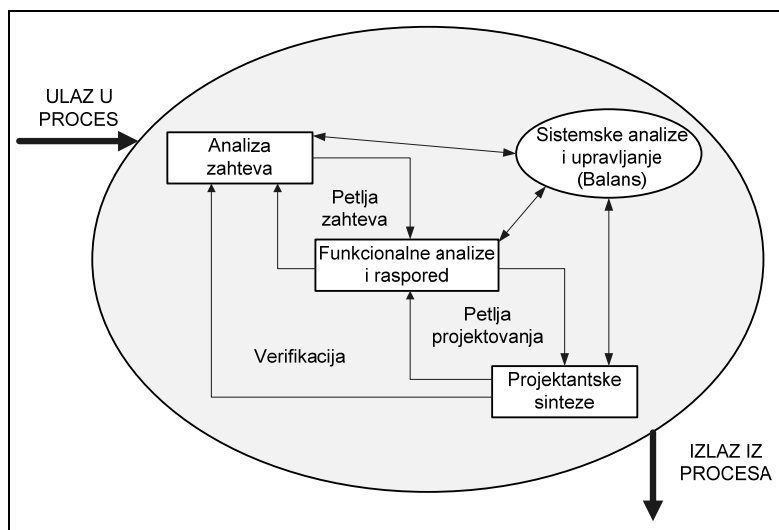
Kao što je prikazano na slici 4 fundamentalne aktivnosti sistemskog inženjeringa su :

Ulazi u proces, koji mogu da budu zahtevi, potrebe i ciljevi kupaca i drugih zainteresovanih strana, tehnološke baze, izlazni zahtevi prethodnog procesa, zahtevi propisa, standarda i drugih specifikacija, i drugo.

Analiza zahteva obuhvata analizu okruženja, identifikaciju funkcionalnih zahteva za proizvod kao sistem, definisanje ili redefinisane performansi pojedinih funkcija.

Funkcionalne analize i alokacije (raspoređivanje) obuhvataju:

- rasčlanjivanje funkcionalnih zahteva na niže funkcionalne nivoe,
- raspoređivanje performansi i drugih graničnih zahteva na sve funkcionalne nivoe,
- definisanje i redefinisane funkcionalnih međuveza (internih i eksternih),
- definisanje, redefinisane i integrisanje arhitekture podsistema odnosno komponenti.



Slika 4: Proces sistemskog inženjeringa [5]

Projektantske sisteze obuhvataju:

- transformaciju arhitekture podsistema (funkcionalne u fizičku),
- definisanje alternativa koncepta sistema, konfiguracije performansi i dokumenata podsistema,
- izbor preporučenih rešenja za sistem i proces u kome će se realizovati,
- definisanje, redefinisanje i integracija fizičkih međuveza (internih i eksternih).

Sistemske analize i upravljanja, čiji je cilj uspostavljanje ravnoteže (balansa) između zahteva, funkcionalnih performansi i izlaznih rešenja sistema i podsistema, obuhvataju studije usaglašavanja zahteva, performansi i rešenja, analize efektivnosti, menadžment rizikom, konfiguracijom, međuvezama, podacima, performansama, itd. Za ove analize koriste se mnogobrojni alati, tehnike i softveri.

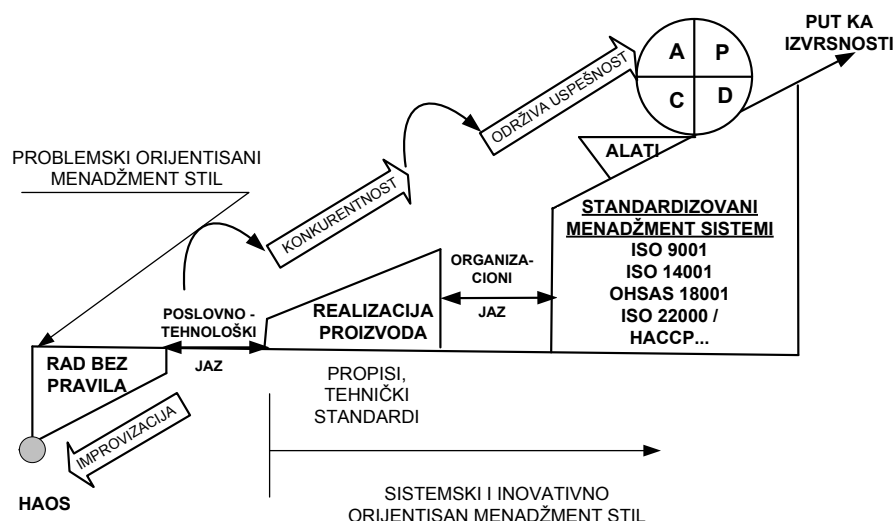
Izlaz iz procesa, zavisno od nivoa procesa razvoja, obuhvata podatke o odlukama, konfiguraciju sistema - strukturu karakteristika sistema i podsistema, arhitekturu podsistema- raspored podsistema u sistemu sa međusobnim vezama i odnosima, specifikacije i druge podatke o rezultatima razvoja (proračuni, analize, crteži, softveri, modeli, prototipovi, itd.).

6. MENADŽMENT SISTEM ZA ODRŽIVI USPEH I DRUŠTVENU ODGOVORNOST

Organizacija koja želi da realizuje složene proizvode strukturane kao sisteme i koja koristi sistemske tehničke procese za njihovu realizaciju mora da izgradi menadžment sistem koji je složeniji od njih. Takav menadžment sistem mora da obezbedi projektni menadžment i održivu uspešnost u realizaciji i plasmanu ovih proizvoda uz poštovanje društvene odgovornosti.

Na slici 5 prikazana je elovucija organizacije čiji se put odvija između krajnjih tačaka; haosa i izvrsnosti. Problemski orijentisani menadžment stil, koji radi praktično “bez pravila”, čija je osnovna karakteristika improvizacija, nije u stanju da kreira i isporučuje proizvode i usluge koji zadovoljavaju utvrđene propise, standarde i zahteve korisnika.

Ključna promena koju treba da preduzme menadžment je gradnja sistemski i inovativno orijentisanog menadžment stila. Prvi korak u tom pravcu, pored gradnje menadžment sistema koji poštuje sve propise u poslovanju, je usklađivanje proizvoda i usluga sa međunarodnim propisima i standardima, kao i zahtevima korisnika. Time se ostvruje prvi korak u konkurentnosti proizvoda na međunarodnom tržištu. Preskakanje ovog “poslovno-tehnološkog jaza” zahteva podizanje tehnološkog i profesionalnog nivoa i primenu savršenih inženjerskih metoda, alata i tehnika u tehničkim procesima.



Slika 5: Evolucija organizacije [6, modifikovano]

Za uspešnu koordinaciju upravljačkih procesa i tehničkih procesa, pri realizaciji složenih proizvoda, neophodno je savladati “organizacioni jaz” što znači izvršiti standardizaciju menadžment sistema prema međunarodnim standardima. U osnovi svih ovih standard je menadžment stil zasnovan na poznatom Demingovom PDCA krugu. U tabeli 2 prikazani su standardi za menadžment sisteme i standardi koji služe kao podrška za efektivnost standardizovanih menadžment sistema, kako pojedinačnih, tako i integriranih.

TABELA 2

Standardi za menadžment sisteme i za podršku njihovoj efektivnosti

Menadžment sistem / podrška za efektivnost		Standard	Zainteresovana strana
Oznaka	Naziv		
Standardi za menadžment sisteme koji se sertifikuju			
QMS	Menadžment sistem kvaliteta	ISO 9001:2008	Korisnik
EMS	Menadžment sistem životne sredine	ISO 14001:2004	Zajednica
OHSMS	Menadžment sistem zdravlja i bezbednosti na radu	OHSAS 18001:2007	Zaposleni Zajednica
FSMS	Menadžment sistem bezbednosti hrane	HACCP/ISO 22000:2005	Korisnik
FMS	Finansijski menadžment sistem	Sarbanes – Oxley Act	Sve zainteresovane strane
ISMS	Menadžment sistem bezbednosti informacija	ISO/IEC 27001:2005	Sve zainteresovane strane
SMS	Menadžment sistem bezbednosti	ISO/PAS 28000:2005	Sve zainteresovane strane
CTCL	Opšti zahtevi za kompetentnost ispitinih i kalibracionih laboratorija	ISO / IEC 17025:2005	Korisnik
ENMS	Energetski menadžment sistem	ISO 50001:2011	Korisnik Zajednica
RTS	Menadžment sistem bezbednosti drumskog saobraćaja	ISO 39001:2012	Korisnik Zajednica

Menadžment sistem / podrška za efektivnost		Standard	Zainteresovana strana
Oznaka	Naziv		
Standardi za podršku zajedničkoj efektivnosti integrisanih menadžment sistema			
GSR	Vodič za društvenu odgovornost	ISO 26000: 2010	Sve zainteresovane strane
MSSO	Menadžment za održivu uspešnost organizacije – pristup preko menadžmenta kvaliteta	ISO 9004:2009	Sve zainteresovane strane
RM	Menadžment rizika	ISO 31000:2009	Sve zainteresovane strane

Standardizacija menadžment sistema postala je uslov za prístup međunarodnom tržištu, kao vid poverenja da će organizacija biti u stanju da ispuni zahteve korisnika. Standardizacijom se stvaraju uslovi da organizacija ispunjava zahteve različitih zainteresovanih strana čime se osposobljava za drugoročan održivi uspeh i društveno odgovorno poslovanje.

7. ZAKLJUČAK

Izlazak iz ekonomske krize moguć je uz održiv privredni oporovak, zasnovan na sopstvenom razvoju i ovladavanju visokim tehnologijama, primenom inženjerskih disciplina. Organizacija koja je sistemski i inovativno orijentisana u stanju je da generiše ideje i da ih pretvara u primenjene inovacije. Menadžment sistem mora da omogućiti održivi razvoj novih proizvoda i održivu uspešnost poslovnih rezultata.

Za kreiranje i realizaciju složenih proizvoda mora da se primeni koncept sistema koji podrazumeva da se kao sistem posmatraju sam proizvod koji se razvija, metode, alati i tehnike koje se koriste u tehničkim procesima kod njegove realizacije, kao i menadžment sistem koji koordinira sve aktivnosti u upravljačkim i tehničkim procesima.

Svaka organizacija koja želi da razvija sopstvene proizvode, usluge ili procese odnosno tehnologije mora da primenjuje koncept sistemskog inženjeringa i da standardizuje svoje menadžment sisteme. Važno je da sistemski pristup, zasnovan na “tvrđim merama i disciplini”, sadrži u sebi i inovativnost zasnovanu na “mekim merama i kreativnosti”.

LITERATURA

- [1.] M. Heleta, D. Cvetković, *Osnove inženjerstva i savremena metode u inženjeringu*, Univerzitet Singidunum, Beograd, 2009.
- [2.] R. Avlijaš, G. Avlijaš, *Preduzetništvo*, Univerzitet Singidunum, Beograd, 2013.
- [3.] Standard ISO 9000:2005 – Menadžment sistem kvaliteta, Termini i definicije
- [4.] M. Heleta, “Koncept sistema u procesu razvoja i projektovanja”, Naučni skup Sinergija 2010, Univerzitet Sinergija, Bijeljina, BiH
- [5.] *System engineering fundamentals*, supplementary text, Defense acquisition University press Fort Belvoir, Virginia, 22060-5565, Januar 2001.
- [6.] M. Heleta, *TQM – Model izvrsnosti i integrisani menadžment sistemi*, Zavod za izdavanje udžbenika, Beograd, 2010
- [7.] M. Heleta, G. Grubor, S. Veljković, “Model for integrated management of process, objectives, risks and performances”, *International Journal of Scientific and Research Publications*, Volume 3, Issue 10 Oktobar 2013.

Kontrola i nadzor inteligentne kuće sa pametnih telefona i tableta

Control and monitoring of intelligent home with smart phones and tablets

Željko Stanković, Gordana Radić, Dragan Cvetković, Mirza Ramičić, Zoran Ilić

Abstrakt - Ovaj rad predstavlja pregled novog koncepta za moguću primenu pametnih mobilnih uređaja (smart telefona, tableta) u definisanim uslovima . Praktičan rad koji je urađen (kontrola pametne kuće) je predstavljen u dve ravni . Prvi plan je opservacija u definisanom ambijentu , pametna kuća sa svim neophodnim funkcionalnim zahtevima , hardverski elementi i komunikacioni moduli. Drugi plan je Android aplikacija koja je 3D interfejs za komunikaciju sa mobilnim uređajima i inteligentnim komponentama kuće. Rad predstavlja odličnu osnovu za eksperimentalnu nastavu programiranja i robotike . Predložena rešenja su zasnovana na radu i istraživanja pomoću najnovije pametnih mobilnih uređaja, komunikacionih modula i softvera.

Ključne reči: Inteligentna kuća, pametni telefoni 3D interfejs

Abstract - This paper presents an overview of the new possible concept for the application of smart mobile devices (smartphones, tablets) in defined environments. Practical work that has been done (control of smart solar house) is presented in two planes. The first plane is observe a defined environment, smart solar house with all the necessary functional requirements, hardware elements and communication modules. The second plane is the Android application that is a 3D interface to communicate with mobile devices and smart components of solar home. The work represents an excellent experimental basis in the teaching of programming and robotics. The proposed solutions are based on the work and research using the latest smart mobile devices, communication modules and software.

Key words: Intelligent house, smart phone 3D interface

1. UVOD

Poslednjih godina svedoci smo velikog tehnološkog razvoja smart telefona i tablet uređaja. Navedeni uređaji povećanjem snage procesora, dodatnim senzorima, kamerama i GPS modulima, postali su respektabilni hardver, tako da sada imamo pravu multimedijalnu platformu na svom dlanu.

¹ Željko Stanković, Fakultet za informatiku i računarstvo, Univerzitet Apeiron, Pere Krece 13, Banja Luka, Republika Srpska (e-mail: zeljko.s@apeiron-uni.eu)¹

Gordana Radić, Fakultet za informatiku i računarstvo, Univerzitet Apeiron, Pere Krece 13, Banja Luka, Republika Srpska (e-mail: gordana.r@apeiron-uni.eu)¹

Dragan Cvetković, Fakultet za informatiku i računarstvo, Univerzitet Singidunum, Danijelova 32, Beograd, Srbija (e-mail: devetkovic@singidunum.ac.rs)

Mirza Ramičić, , Fakultet za informatiku i računarstvo, Univerzitet Apeiron, Pere Krece 13, Banja Luka, Republika Srpska (e-mail: mirza.ramicic@hotmail.com)¹

Zoran Ilić, Fakultet za informatiku i računarstvo, Univerzitet Apeiron, Pere Krece 13, Banja Luka, Republika Srpska (e-mail: zorni@pd.elta-kabel.com)¹

Još jedna oblast koja je doživela buran razvoj su rešenja za inteligentne kuće i korišćenje zelenih tehnologija.

Integrisanjem tabletu ili pametnog telefona u sistem inteligentne solarne kuće proširuje se polje korišćenje pametnih mobilnih uređaja.

Ovaj rad predstavlja prikaz integracije tableta (mobilnih uređaja) u sistem kontrole i video nadzora pametnih kuća, zgrada i objekata u potpuno novom okruženju. Preko dinamičkog 3D interfejsa mobilnog uređaja (virtuelan svet) može da se kontroliše željeni deo u kući (paljenje svetla, klima uređaj, veš mašina grejanje objekta, video nadzor prostorija) realan svet.

3D interfejs, koji je dizajniran u ovom radu predstavlja novu viziju pametnih telefona i tableta u oblasti inteligentnih solarnih kuća.

Inteligentni mehatronski sistem solarne kuće veoma je složen. Stoga je i dizajn kontrolnog polja samog menija 3D interfejsa zahtevao definisanje svih elementa koji će biti uključeni u sistem kontrole pametne solarne kuće. Složenost sistema podeljena je na sledeći način:

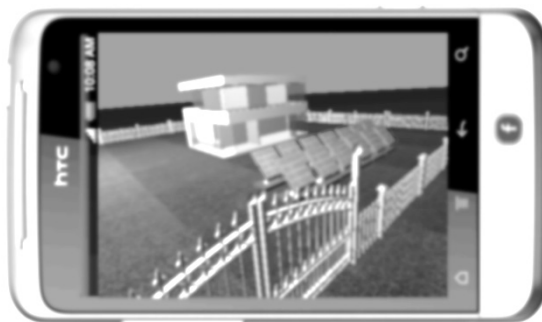
- a) kuća,
- b) solarno polje,
- c) okolina (zemljište, dvorište, bašta)

Sl. 1, pokazuje prvi ulazni interfejs dizajniran u klasičnom stilu.



Slika 1. Ulazni interfejs klasični stil

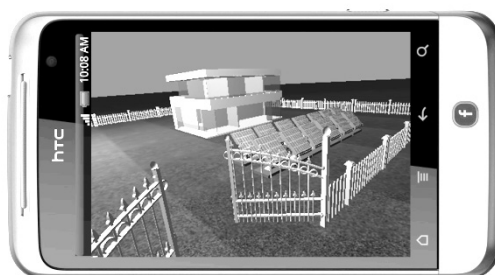
Kućni sistemi, u literaturi poznato i kao "kućna automatika", sve su to termini koji se koriste za opisivanje novih aplikacija [1]



Slika 2. Ulazni interfejs dinamički model 3D kućni kontrolni panel, u stanju je da kontroliše osvetljenje, termostat, bezbednost, brave i multimedijalne uređaje.

Proširena funkcionalnost dobija se 3D modelom dinamičkog ulaznog interfejsa, sl. 2. Definisanjem radne pozicije može se direktno pristupiti etažama kuće, svakoj prostoriji posebno, dvorišnim elementima, ili ulaznoj kapiji. Sam pristup prostorijama preko 3Dinterfejsa vraća trenutno stanje pokretnih i stacionarnih elemenata integrisanih u datom prostoru, inteligentne solarne kuće.

Primer koji eksplicitno prikazuje funkcionisanje 3D interfejsa prikazan je na slici. 2. Može se videti da je kapija bila zatvorena. Kada kliknete na kapiji dobijate opciju "gate open" (otvori kapiju). Kapija će se otvoriti i novo stanje biće odmah ažurirano i prikazano na ekranu mobilnog uređaja. To predstavlja prednost jer se sve odvija u realnom vremenu na dinamičkom 3D interfejsu smart telefona ili tableta.



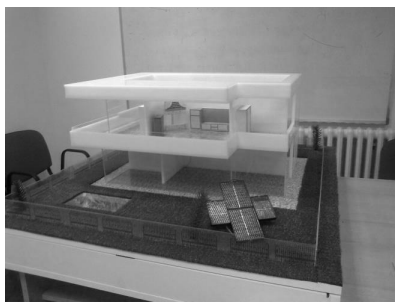
Slika 3. Primer "gate open"

2. RADNI MODEL

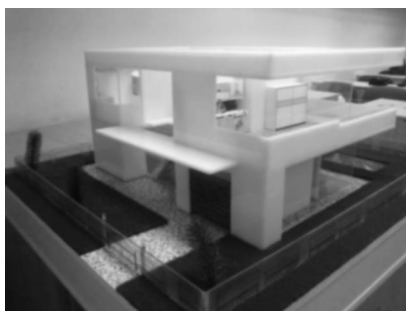
Kuća - eksperimentalni funkcionalni model ima dva sprata, prizemlje i prvi sprat, sl. 4. i sl. 5.

Na meniju 3D interfejsu definisan je taster "HOME" (kuća). To je skup sa najsloženijim elementima 3D interfejs dizajna, tako da je proces programirawa počeo od ovog prvog segmenta.

Prilikom definisanja kontrolnih komande uziman je u obzir i broj soba. U projektu koji je predstavljen bilo je potrebno da se kontrolišu 14 soba. U isto vreme dobijaju se informacije od 24 senzora (temperature i vlažnost vazduha u svim sobama).



Slika 4. Radni 3D model urađen od providnih plastičnih panela



Slika 5. Radni 3D model urađen od providnih plastičnih panela

3D model kuće podeljen je po spratovima. Svaki sprat predstavlja zaseban entitet koji se sastoji od sedam soba. Za svaku sobu napravljen je dinamički 3D model sa svim srodnim komandama. Radni model je prikazan na slici. 6.

Pošto se radi o dinamičkom 3D interfejsu sve informacije od senzori i prekidača za kontrolu treba da se postavi na mesto, a da ekran i dalje bude pregledan. U prvoj verziji instalirani su samo senzori temperature i prekidači za svetla i klima uređaje.



Slika 6. 3D Interfejs dinamičkog modela jedne od soba

Video nadzor omogućen je preko Wi-Fi kolor video kamere. Na slici 7 prikazana je kamera koju koristi projekat inteligentne kuće.



Slika 7. Wi-Fi kamera

3. TEHNOLOGIJA KOJA STOJI IZA PROJEKTA INTELIGENTNE KUĆE

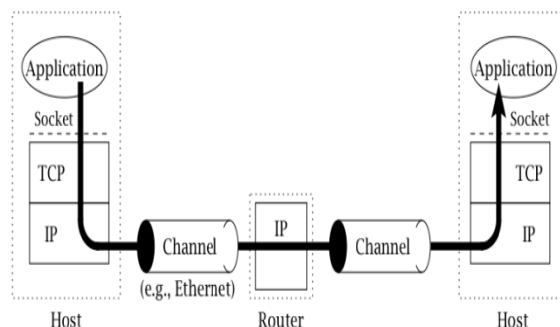
Postoje dve glavne komponente koje rade zajedno u sinergiji kako bi projekat bio funkcionalan.

Softverske komponente razvijane su za operativni sistem Android [2] koji je prilagođen mobilnim platformama u cilju da se obezbedi korisnički 3D interfejs koji treba da oponaša realni svet na modelu kuće – predstavljenog projekta.

Hardverska komponenta koja kontroliše realan model realizovan je korišćenjem “open source” (programski kod koji je dostupan svima) a zasnovan je jednim delom na Arduino [3] prototip platformi dok je deo za komunikaciju sa tabletom ili pametnim telefonom urađen kao fakultetski prototip.

Suštinski ovim hardverskim komponentama ostvaruje se mogućnost da koristimo Internet za prenos podataka i protokole koji osiguravaju standardizovan i široko dostupni platformu za prenos podataka posebno kada se koristi i IP kamera. Mi smo koristili peer-to-peer mrežnu arhitekturu koja eliminiše potrebu za centralizovanim serverom radi prenosa mrežnog prometa sa klijentima. U ovoj arhitekturi obe strane mogu da deluju kao klijent i kao server. Umesto da se koristi asinhroni UDP protokol, komunikacija između komponenti je osnovana neposredno i koristi TSP protokol, koji obezbeđuje siguran i bez grešaka prenos podataka, slika 8.

TSP je dizajniran da otkrije i ispravi greške, gubitke, dupliranja i druge greške koje se mogu pojaviti u host-to-host IP kanalu. TSP obezbeđuje pouzdan protok po kanalu, tako da aplikacije ne moraju da se bave ovim problemima.



Slika 8. TCP/IP mreža

Arduino hardverska komponenta povezuje se na Wi-Fi mrežu, dobija se IP adresa i “slušaju” dolazne veze na određenom portu. Projektovani softver zatim šalje zahtev za uspostavljanje veze sa Arduino komponentom, a podaci se šalju upakovani. Paketni sadržaji ima kodiranu komandu za Arduino komponentu koja se dekodira i izvršava na hardverskoj strani. Posle uspešnog izvršenja komande povratna poruka šalje se sinhrono Android kontrolnoj aplikaciji, kako bi se korigovalo (ažuriralo) novo stanje na softverskoj strani (3D interfejsu).

`status = WiFi.begin(ssid); // Attempting to connect to a Wi-Fi network` PLC hardverska komponenta [4] koristi sledeće komande za povezivanje na bežičnu mrežu i sluša dolazeće veze.

`WiFiServer server(port); // Setting the listening port`
`server.begin(); // Start the server mode`

Nakon toga Arduino komponenta ulazi u petlju koja stalno proverava da li svaki klijent uspostavlja vezu.

`client.connected(); // Returns from a loop when the client is connected`
`charRead = client.read(); // Reads the command`
`client.write(command); // Returns the data to the client`

Aplikaciona komponenta koristi Microsoft.Net Framework TCPClient koji je definisan u `system.networks.sockets`.

TCPClient klasa pruža jednostavne metode za povezivanje, slanje i primanje toka podataka preko mreže u sinhronom blok režimu.

The typical TCP client's communication involves four basic steps:

1. Create a TCP socket using `socket()`.
2. Establish a connection to the server using `connect()`.
3. Communicate using `send` and `recv()`.
4. Close the connection with `close()`.

Ispod je primer metoda i klasa naših aplikacija koje se koriste da pošalju paket podataka za hardverske komponente.

```
TcpClient client = new TcpClient(server, port); // Make a new client
Stream stream = client.GetStream(); // Get a client stream for reading and writing.
stream.Write(data, 0, data.Length); // Send the message to the connected TcpServer.
Int32 bytes = stream.Read(data, 0, data.Length); // Read the first batch of the TcpServer response bytes.

// Close everything.
stream.Close();
client.Close();
```

4. ZAKLJUČAK

Inteligentni model solarne kuće sa 3D dinamičkim interfejsom za pametne telefone i tablet uređaje sigurno će biti dobra osnova i eksperimentalna platforma. Sa dodatnim senzorima može se koristiti za širok spektar nastavnih modula u oblasti tehničkog obrazovanja, mehatronika, elektronike i informacione tehnologije. Bez obzira da li želite da rešavate zadatke dizajna inteligentne kuće ili realizovati zadato kretanje solarnih panela, raspravljati sa svojim studentima na temu programiranja i funkcije senzora pomoću praktičnih primera, ili želite da razvijete nov simulacioni program u nekom od programskih jezika, predložena rešenja su prvi korak i idealno polazište za savremenu eksperimentalnu nastavu.

5. REFERENCE

- [1] "Home systems overview" Garrett, Simon Home Systems - Information, Entertainment and Control, Publication Year: 1990 , Page(s): 1/1- 1/3
- [2] <http://en.wikipedia.org/wiki/Android>
- [3] <http://arduino.cc/en/Main/ArduinoBoardDue>
- [4] "Univerzalni programibilni logički kontroler", Cvrtković D., Stanković Ž, Miljković Z.: Konferencija 36 Jupiter, Beograd, pp. , 2010.
- [5] <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>

Menadžment znanjem i informacione tehnologije

Knowledge management and information technologies

Biljana Tešić, Marko Marković, Katarina Plečić, Ivan Pantelić,
Poslovni fakultet Valjevo, Univerzitet Singidunum

Ovaj rad je delimično finansiran od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije (TR32054).

Apstrakt - Upravljanje znanjem pomoću primene savremenih koncepata informacionih tehnologija je neizostavni element strategije uspešnih globalnih poslovnih sistema, a isto tako i državnih institucija razvijenih zemalja. Na taj način se strategijske potrebe organizacija za inženjeringom i reinženjeringom novog znanja dostižu sinergijom pomoću procesiranja informacija uz pomoć naprednih informaciono-komunikacionih tehnologija (ICT) i korišćenjem inovativnih i kreativnih kapaciteta koji su skriveni u ljudskom društvu. Primena koncepta definisanja procesa upravljanja znanjem, projektovanja sistema za menadžment znanjem, kao i primena alata i tehnologija poslovne inteligencije omogućava efikasnije i efektivnije rešavanje upravljačkih problema na svim nivoima odlučivanja.

Današnje vreme svakodnevno donosi nove i brze promene okoline kojima se treba što pre, a često i u hodu prilagođavati. Velike mogućnosti i širok spektar ponude softverskih alata koje nudi tržište informacione tehnologije za pomoć pri obavljanju važnih poslova planiranja, a naročito donošenja poslovnih odluka, među kojima se nalaze i alati poslovne inteligencije, velik broj poslovnih sistema, a i državne institucije u našoj zemlji, ne prepoznaju i ne koriste još u zadovoljavajućoj meri. Iz tog razloga obrada ove teme ima značajnu težinu, a istovremeno predstavlja izazov i privlačnost.

Ključne reči - Menadžment znanjem, informacione tehnologije, poslovna inteligencija

Abstract - Using modern information technology concepts for knowledge management is an essential element of successful global business strategy system, and developed countries government institutions also. That way, organization strategic needs for engineering and reengineering of new knowledge are reached by the synergy of the information processing using advanced information and communication technologies (ICT) and using innovative and creative capacities hidden in society. Applying the concept of defining knowledge management process, the design of the knowledge management, and the application of tools and business intelligence technologies results in efficiently and effectively solving management problems at all decision making levels.

dr Biljana Tešić, - Poslovni fakultet Valjevo, Univerzitet Singidunum, Železnička 5, 14000 Valjevo, Srbija (e-mail: btesic@singidunum.ac.rs).

Marko Marković, MSc - Poslovni fakultet Valjevo, Univerzitet Singidunum, Železnička 5, 14000 Valjevo, Srbija (e-mail: mmarkovic@singidunum.ac.rs).

Katarina Plečić, MA - Poslovni fakultet Valjevo, Univerzitet Singidunum, Železnička 5, 14000 Valjevo, Srbija (e-mail: kplecic@singidunum.ac.rs).

Ivan Pantelić, MSc- Poslovni fakultet Valjevo, Univerzitet Singidunum, Železnička 5, 14000 Valjevo, Srbija (e-mail: ipantelic@singidunum.ac.rs).

Nowadays, every day brings new and rapidly changing surroundings which need to be adapted as soon as possible, and often along the way. There are great possibilities and a wide range of software tools that provide information technology market for assisting in performing important tasks of planning. Particularly business decision making, which includes business intelligence tools, has not been recognized and used sufficiently in many business systems, and government institutions in our country also. For this reason, this topic has great significance and a challenge and attraction as well.

***Index terms* - Knowledge management, Information technologies, Business intelligence**

1. UVOD

Radikalne promene u poslovnom okruženju traže novu koncepciju upravljanja u odnosu na dosadašnju praksu. Tako se nametnula potreba za korišćenjem savremenih upravljačkih alata i tehnologija koje omogućavaju sveobuhvatno, brzo i efikasno korišćenje svih dostupnih podataka i informacija, unutar i izvan poslovnih sistema važnih za uspešno upravljanje poslovnom organizacijom. Tako, na primer, potreba za informacijama jedne industrijske kompanije javlja se radi potrebe izrade tačne analize pojedinih tržišnih segmenata, analiziranja ponašanja kupaca i dobavljača, lakšeg pregovaranja s kupcima i dobavljačima, odabira ključnih kupaca i dobavljača, zbog pridošlica u industriju, zamenskih proizvoda i naravno postojeće konkurencije. Nadalje potrebno je uočiti osnovne probleme i uzroke njihove pojave, da bi se pravovremeno i što uspešnije mogle otkloniti neželjene posledice.

Upravljanje znanjem primenom koncepta i korišćenjem savremenih alata ICT, posebno metodologije poslovne inteligencije od strane svih poslovnih subjekata danas je neophodno za sticanje konkurentske prednosti i opstanak na tržištima. To ne može izbeći ni jedna poslovna organizacija koja u globalnom okruženju više ne može dozvoliti kampanjski i ad-hoc način rešavanja poslovnih problema. Široko prihvatanje tog koncepta od strane proizvodnih poslovnih sistema, a isto tako i na svim nivoima državne uprave sigurno je jedan od puteva koji može približiti poslovno okruženje u Srbiji krugu uspešnih i prosperitetnih zemalja.

U poslednje vreme veći broj istraživanja se bavi konceptom znanja i upravljanja znanjem te se sve više naglašava uloga znanja u mogućem rešavanju poslovnih problema današnjice, u kojima se obrađuju teme vezane za savremeni menadžment i rešavanje pitanja opstanka preduzeća u nesigurnom poslovnom okruženju. Uz navedeno nadovezuju se i pojmovi intelektualni kapital i poslovna inteligencija. Poslovna inteligencija predstavlja korišćenje svih potencijala podataka i informacija u organizaciji radi donošenja boljih poslovnih odluka, kao i u cilju identifikacije novih poslovnih mogućnosti [1].

Istovremeno sve češće se govori o prelazu iz informacionog društva u društvo znanja. Pri tom se retko definiše sam pojam znanje, dok je pojam upravljanje znanjem (engl. knowledge management), krajem devedesetih godina prošlog veka bio dosta popularan i obrađivan u stranoj stručnoj literaturi kao jedan od mnogih "alata" koji stoje na raspolaganju savremenom menadžmentu u svakodnevnoj praksi.

Naglašavanje uloge znanja u novije vreme posledica je značajnih strukturnih promena u razvoju ekonomije. Ona se krajem prošlog veka potpuno i suštinski menja prelaskom iz industrijske u savremenu informacionu ekonomiju današnjice, glavni resursi više nisu materijalna dobra, već novi proces ekonomskih aktivnosti i produktivnosti postaje ekonomija znanja podržana savremenim IT u Internet okruženju.

Osnovna orijentacija rada je na savremene informacione tehnologije sa posebnim osvrtom na njihovu primenu u procesu projektovanja sistema menadžmenta znanjem i reinženjeringu poslovnih procesa, istraživanje čiji su elementi bitni za uspešnu implementaciju poslovne inteligencije u poslovne sisteme.

2. MENADŽMENT ZNANJEM

Sam pojam znanje može se definisati kao nematerijalni resurs, slika stvarnosti iskazana zamislama čoveka dok posmatra svet koji ga okružuje: prostor, objekte, odnose i događaje u toj stvarnosti. Ono se

sastoji od intuicije, skupa ideja, iskustva, veština i učenja i ima potencijal stvaranja nove vrednosti. Znanje nastaje u umovima ljudi, a prema autoru koji se među prvima bavio pitanjima kreiranja i korišćenja znanja, Michaelu Polanyiu [2], znanje možemo podeliti na eksplicitno (engl. explicit knowledge) i tiho ili iskustveno znanje (engl. tacit knowledge).

Eksplicitno znanje artikulirano je formalnim jezikom i može se lako prenositi u obliku informacije s pojedinca na pojedinca. Ono je uskladišteno u knjigama i može se jednostavno "dohvatiti" procesom obrazovanja, upotrebljavati, preuzeti i ugraditi u proizvode u oblicima kao što su tehničke specifikacije, nacrti ili standardizovani dizajn. Suprotno tome, tiho ili iskustveno znanje je znanje ugrađeno ili stvoreno ličnim iskustvom pojedinca i uključuje u sebi nedodirljive elemente kao što su lično uverenje, instinkt, vrednosti i stečene veštine. Ono je personalizovano i do njega nije lako doći pa često ostaje neotkriveno i neiskorišćeno. Ovaj oblik znanja bogatiji je od opšteg, eksplicitnog tipa, međutim, ono nema vrednost ako nije upotrebljeno. Posmatrano iz perspektive poslovne kompanije, iskustveno znanje svih njenih zaposlenih nije lako prikupiti, deliti i upotrebljavati. Međutim ta vrsta znanja ima veliku vrednost ako se upotrebi i ono po pravilu donosi novu vrednost kompaniji i predstavlja značajan element konkurentske prednosti.

Premda je znanje nematerijalan resurs, može se skupljati, skladištiti i prenositi i to govorom, tekstom i slikom. Iako je ono izvan medija može se kreirati i prenositi korišćenjem istih ili kombinacijom različitih medija.

Zapravo, jedina održiva prednost savremene organizacije proizilazi iz onoga što firma zna, koliko efikasno koristi ono što zna i koliko brzo stiče i koristi novo znanje. Savremena organizacija u eri znanja je ona koja uči, pamti i deluje na osnovu informacija i znanja dostupnog na najbolji mogući način.[3]

Sa aspekta savremenih ICT menadžment znanjem se može definisati kao:

- Interdisciplinarni, poslovni model koji se bavi svim aspektima znanja u okviru organizacije, uključujući stvaranje znanja, kodifikaciju znanja, deljenje znanja i kako ove aktivnosti podstiču učenje i inovaciju, potpomognuti informaciono-komunikacionom tehnologijom i organizacionim rutinama.
- Proces prikupljanja kolektivne ekspertize organizacije, gde god se ona nalazila, i distribucija te ekspertize tamo gde ona može najviše da doprinese.
- Proces koji predstavlja stalnu, svrsishodnu interakciju između donosilaca odluke kroz koju on ima za cilj da upravlja drugim donosiocima odluka, komponentama i aktivnostima koje učestvuju u osnovnim procesima znanja u planiranu, usmerenu celinu koja proizvodi, održava, proširuje, usvaja i prenosi bazu znanja organizacije.

Razmatrajući pojam znanja u okviru ovog rada koji daje naglasak na poslovni proces, znanje se može vezati uz čoveka i organizaciju, a njegova realizacija u organizaciji odvija se kroz uspostavljanje sistema upravljanja znanjem. Danas u savremenom poslovnom svetu postoji razvijena svest da je ukupno znanje u organizaciji značajno veće od onog koje se koristi u procesima organizacije. U tom smislu interesantna je izjava Lew Platta, bivšeg izvršnog direktora Hewlet Packarda: "Da je HP znao što HP zna, mi bi bili tri puta više profitabilniji" [2].

Pored prethodno navedene klasifikacije, znanje se prema supstanci može podeliti na činjenično znanje, proceduralno znanje i znanje za prosuđivanje. Činjenično znanje čine podaci i informacije, proceduralno znanje predstavljaju tehnike kao što su algoritamski i heuristički postupci i koriste se u ekspertnim sistemima ili od strane stručnjaka za rešavanje određenih problema. Znanje za prosuđivanje omogućava vrednovanje ograničenja i bitno je za menadžment prilikom procesa planiranja i postavljanja ostvarivih ciljeva u planovima. Na slici 1. dat je grafički prikaz klasifikacije znanja prema supstanci.



Slika 1. Klasifikacija znanja prema supstanci

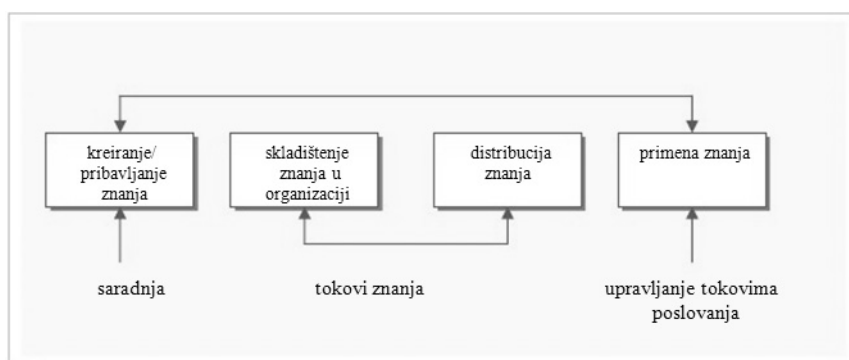
3. PROCES UPRAVLJANJA ZNANJEM

U uvodnom delu navodi se značaj znanja u smislu poslovnog upravljanja. U modernoj ekonomiji najvrednijim resursom današnjeg poslovanja smatra se ljudsko znanje, stoga upravljanje znanjem predstavlja nezaobilaznu komponentu savremenog menadžmenta.

Kao pojam i definiciju, u smislu poslovnog upravljanja, možemo reći da upravljanje znanjem predstavlja proces koji se odnosi na identifikovanje-utvrđivanje znanja, organizovanje, skladištenje i distribuciju znanja i iskustva individualnih grupa unutar organizacije. Opšti cilj upravljanja znanjem je povezati one kojima je znanje potrebno sa izvorima znanja u organizaciji, kao i usklađivanje transfera tog znanja. Konkretni cilj upravljanja znanjem predstavlja proces utvrđivanja i analize raspoloživog znanja kako bi se ispunili zadati ciljevi poslovanja organizacije.

Nadalje, upravljanje znanjem podrazumeva konstantni proces obnove znanja u organizaciji i time se ono u osnovi razlikuje od drugih pristupa, kao što je reinženjering poslovnih procesa ili sistema upravljanja kvalitetom. Osnovni cilj reinženjeringa aktivnost je uspešno rešavanje određenog problema u definisanom, po mogućnosti što kraćem vremenskom periodu, a cilj procesa upravljanja znanjem je kontinuirana inovacija i modifikacija postojećeg znanja kako bi se organizacija prilagodila dinamičnom tržišnom okruženju.

Proces upravljanja znanjem sastoji se od nekoliko faza, i to: prikupljanje/kreiranje znanja, skladištenje znanja, distribucija znanja i na kraju primena znanja. U organizaciji kojoj je cilj kontinuirana obnova znanja, tzv. "learning organization" taj proces je stalan i cikličan. Slika 2. prikazuje ključne faze procesa upravljanja znanjem.



Slika 2. Ključne faze procesa upravljanja znanjem

4. ZNAČAJ UPRAVLJANJA ZNANJEM

Znanje je osnov za donošenje poslovnih odluka i kritični faktor uspeha za poslovanje tako da nije potrebno posebno naglašavati da danas konkurentska prednost kompanije zavisi od toga koliko uspešno ona upotrebljava znanje na svim menadžment nivoima (operativni, taktički, strategijski) i naročito kako brzo može naučiti i usvojiti nešto novo. Prema istraživanjima, najveći deo znanja, između 50 i 95% i to eksplicitnog i iskustvenog, prenosi se usmenom, direktnom komunikacijom. Pritom se veliki deo znanja izgubi, pa uglavnom kompanije upotrebljavaju oko 20% svog organizacionog znanja. Naravno da je danas važno pronaći i zadržati ostatak neiskorišćenog znanja u kompaniji i pretvoriti ga u intelektualni kapital. Sistemsko prikupljanje i upotreba znanja u cilju

generisanja novih ideja postiže se stvaranjem odgovarajućeg okruženja u kojem upotreba novih informaciono-komunikacionih tehnologija (ICT) znatno olakšava rešavanje tog problema.

Sposobnost kontinuiranog generisanja novih ideja i stvaranje inovativnih proizvoda karakteristika je uspešnih kompanija. Za njih inovacija predstavlja deo strategije poslovanja. One poseduju odgovarajuću organizacionu strukturu i menadžment koji podstiče inovativnost. Osnov kreativnosti i inovativnosti postiže se i stvaranjem timova koji se sastoje od ljudi različitih profila, što je preduslov za kreiranje različitih pitanja i generisanje novih ideja.

S obzirom da se do inovacije po pravilu ne dolazi za kratko vreme važno je da menadžment ne očekuje brze rezultate, već da sistematski usmerava i razvija definisanu strategiju i podstiče aktivnosti značajne za praktičnu primenu upravljanja znanjem kao što su:

1. podsticanje kreativnosti, timskog rada i istraživačkog pristupa poslovanju,
2. primena benchmarking tehnika, brainstorminga i sl. metoda,
3. ulaganje u istraživačke studije i razvoj,
4. redovno analiziranje poslovnih procesa,
5. samostalno razvijanje novog znanja ali i kupovanje znanja,
6. organiziranje treninga i ostalih vidova obrazovanja,
7. stvaranje projektne dokumentacije,
8. učenje iz završenih projekata,
9. sprovođenje "cost-benefit" analize projekata,
10. upotreba znanja izvan organizacije (knjige, studije, obrazovne institucije),
11. prikupljanje, skladištenje i analiza informacija o kupcima, dobavljačima i partnerima,
12. razvijanje preduzetničke organizacione kulture,
13. razvijanje odgovarajuće ICT infrastrukture - informacionog sistema,
14. znanja uz upotrebu intraneta, interneta i Web-a,
15. upotreba tehnologija poslovne inteligencije.

5. USKLAĐIVANJE SISTEMA UPRAVLJANJA ZNANJEM I IT-A

Delotvorno upravljanje znanjem zahteva kombinaciju ljudi i tehnologije. Dok su ljudi potrebni radi razumevanja, sinteze i interpretacije raznih vrsta nestrukturiranog znanja, kompjuterski i komunikacioni sistemi obezbeđuju njegovo prikupljanje, transformaciju i distribuciju. Kompjuterski i mrežni sistemi i njihovo korišćenje (e-mail, groupware, Internet, intranet) omogućavaju jednostavnu i brzu razmenu znanja bez obzira na udaljenost. Videokonferencije i multimedija mogu preneti zvuk i sliku na način koji je neuporedivo efikasniji i sadržajniji nego što se to postiže personalnom komunikacijom.

Svakodnevno smo u mogućnosti da u realnom vremenu radimo sa drugima, razmenjujemo ideje i koordiniramo poslovnim procesima i aktivnostima. Nove tehnologije nude značajno efektivniji model saradnje koji omogućava interakciju bez obzira na vremensku i prostornu dimenziju koji nazivamo asinhrona komunikacija. Oblici asinhrona komunikacije su e-mail, videokonferencije, indeksirane baze podataka, web portali, Internet, intraneti i sl.

Pravi izazov IT-a predstavlja razvoj informacionih sistema koji će biti korišćen za propagiranje, deljenje i korišćenje organizacionog znanja. Za mnoge organizacije upravljanje znanjem znači postavljanje intraneta, mreže nalik na Internet, ali samo u okviru organizacije. Ovde se može govoriti i o ekstranetu, mreži sa ograničenim pristupom za kupce, dobavljače i druge partnere, kao i o raznim on-line forumima i diskusionim bazama uz pomoć kojih se mogu dobiti odgovori na razna pitanja od onih koji ih znaju.

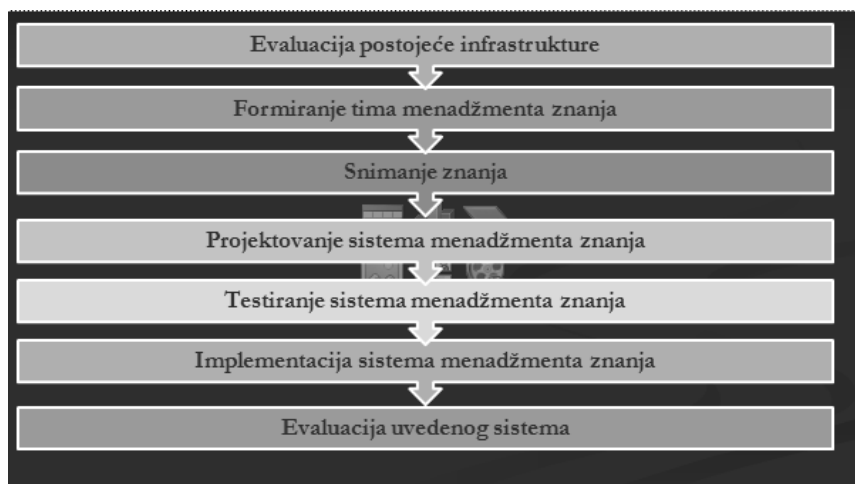
Uspeh upravljanja organizacijom zavisi od veštine obavljanja funkcija upravljanja kao što su planiranje, organizovanje, rukovođenje i kontrola. Da bi se izvršile ove funkcije, menadžeri učestvuju u stalnom procesu donošenja odluka. [4] Zato se kao izuzetan alat za potrebe menadžmenta znanja mogu koristiti algoritmi poslovnog odlučivanja, tako da je moguće od postojećih podataka koje organizacija poseduje doći do nekih potpuno novih informacija koje prethodno nisu bile poznate, a mogu biti od izuzetnog strateškog značaja u donošenju poslovnih odluka. [5]

6. PROJEKTOVANJE SISTEMA MENADŽMENTA ZNANJEM

Savremeno poslovanje zasniva se na procesima, a ne na funkcijama, zadacima, strukturama i proizvodima. Proces predstavlja skup aktivnosti koje se sprovode u svrhu stvaranja izlazne vrednosti za pojedinog kupca na tržištu. Za sistem menadžmenta znanjem neophodno je definisati ključne poslove procese i procese menadžmenta koji su usmereni na interakcije sa procesima iz okruženja, planiranje, kontrolu, merenje, praćenje, analiziranje, poboljšanje, unapređenje, upravljanje sistemom i odlučivanje. Na taj način tokovi informacija su usmereni ka procesima upravljanja kako bi menadžment raspolagao pravovremenim i kvalitetnim informacijama neophodnim za donošenje validnih poslovnih odluka.

Sa aspekta vlasništva nad procesima, menadžment je zadužen za definisanje procesa koje treba unaprediti, zbog čega je potrebno definisati uloge i odgovornosti menadžmenta za funkcionisanje procesa u okviru sistema. Međutim, procesu je potrebna tehnologija, pre svega informaciona tehnologija, a organizaciju treba osposobiti da može primeniti savremene tehnologije i da bude spremna na stalne promene u internom i eksternom okruženju.

Projektovanje sistema menadžmenta znanjem obuhvata sve faze životnog ciklusa razvoja informacionih sistema: dekompoziciju poslovnog procesa, projektovanje organizacione strukture, analizu ključnih aktivnosti, analizu zahteva korisnika, relacionu analizu, dizajn baze podataka i projektovanje aplikativno-sofverske arhitekture informacionog sistema. Faze projektovanja sistema menadžmenta znanjem prikazane su na slici 3.



Slika 3. Faze projektovanja sistema menadžmenta znanjem

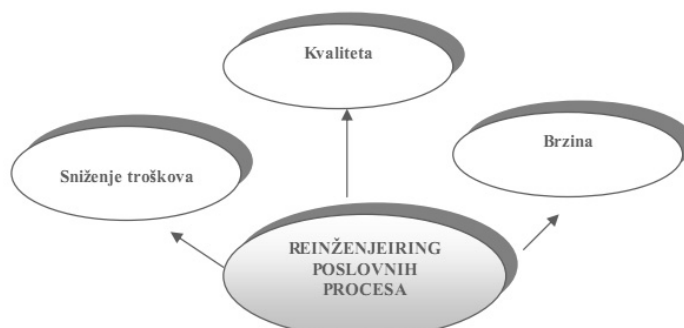
7. REINŽENJERING POSLOVNIH PROCESA SISTEMA MENADŽMENTA ZNANJEM

Novi skup ideja, postupaka i načela, početkom devedesetih godina prošlog veka obuhvaćen je pojmom reinženjeringa poslovnog procesa, a podrazumeva takve zahvate u organizaciji i dizajniranju poslovnih procesa koji donose suštinske, kvalitativne promene. Reinženjering poslovnog procesa podrazumeva sistematski pristup u cilju radikalnog unapređenja ključnih poslovnih procesa organizacije (engl. core business) kao i ključnih procesa podrške. [6]

Reinženjeing se obično fokusira na nekoliko ključnih poslovnih procesa među mnogim procesima organizacije koji su bitni za uspeh kompanije u okviru sektora u kojem kompanija posluje. To su po

pravilu oni procesi koje je kompanija u svojoj strategiji identifikovala kao kritične za postizanje poslovne efektivnosti i vodeće pozicije na tržištu.

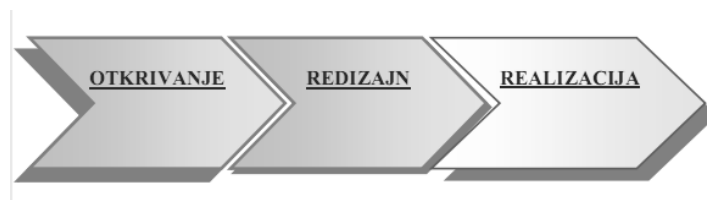
Sušтина reinženjeringa nije samo srediti, modifikovati ili manjim parcijalnim zahvatima reorganizovati postojeće procese već temeljno redefinisati postojeće procese s ciljem postizanja drastičnih poboljšanja ključnih parametara poslovanja i to prvenstveno: troškova, kvalitete i brzine (slika 4).



Slika 4. Ključni parametri na koje se fokusira reinženjering

Primenu reinženjeringa poslovnog procesa karakterišu tri faze. U prvoj fazi kompanija kreira stratešku viziju kojom otkriva i utvrđuje procese koje treba redizajnirati npr. u cilju dominacije ili obnove konkurentnosti na tržištu, sprovodi determinaciju aktivnosti na pojedinim procesima da bi se ostvarila definisana strategija.

Faza "otkrivanja" je esencijalna u ispitivanju poslovanja radi identifikovanja mogućnosti i rangiranja ključnih procesa nad kojima će se sprovesti reinženjering. Na početku ove faze rukovodstvo kompanije vrši selekciju procesa, određuje multidisciplinarni tim koji će sprovesti reinženjering i donosi odluku kojim će se alatima i tehnikama koristiti u sprovođenju reinženjeringa. Druga faza odnosi se na aktivnosti samog redizajna odabranih procesa tokom koje se donose detaljni planovi izvršenja postupaka reinženjeringa. Treća faza predstavlja fazu realizacije i implementacije. (slika 5)



Slika 5. Osnovne faze u provedbi reinženjeringa

Zavisno od kompanije, one se odlučuju za primenu reinženjeringa iz različitih razloga. Kompanije lideri, premda dobro i uspešno posluju, odlučuju se za unapređenje poslovnih parametara s ciljem da još više učvrste postojeću vodeću poziciju, dok to druge rade jer žele dostići svoju konkurenciju. Drugi razlog za reinženjering može biti uvođenje novih tehnologija, naročito informacionih tehnologija, koje se sve više ugrađuju u proizvode čime se procesi automatizuju i ubrzavaju. IT obezbeđuje potrebne informacije sve većem broju korisnika uključenih u procese proizvodnje, olakšava i ubrzava donošenje efektivnih odluka, ne samo menadžmentu, već svim uključenim u navedene procese. Na taj način omogućeno je skraćivanje vremena razvoja, proizvodnje i distribucije proizvoda, povećanje kvaliteta, čime oni postaju konkurentniji na tržištu i omogućavaju kompaniji održavanje liderske pozicije.

8. ZAKLJUČAK

Iako se ozbiljniji naponi na konstituisanju i teorijskom fundiranju discipline upravljanja znanjem uočavaju tek devedesetih godina prošlog veka, značaj ove discipline neprestano raste. Objektivno govoreći, sama disciplina se još uvek nalazi u prelasku iz faze osnivanja u fazu razvoja, ali to ne predstavlja prepreku za njemu uspešnu primenu u svakodnevnom poslovanju preduzeća.

U takvom poslovnom okruženju gde je primena savremene informacione tehnologije neizostavna i gde se razvojem informacionih sistema pružaju razne mogućnosti (poslovne aplikacije za elektronsko

poslovanje, ispitivanje operativnih podataka, sistemi za podršku poslovnom odlučivanju, ekspertni sistemi itd.) za stvaranje kompetitivne prednosti, nedopustivo je da se jedan menadžer izvan svog domena i svojih funkcija ne razume u informacione tehnologije, i da jedan IT stručnjak ne razume logiku organizacionih i poslovnih tokova sa aspekta procesa menadžmenta znanjem.

LITERATURA

- [1] Ćirić B., Poslovna inteligencija, Data status, Beograd, 2006.
- [2] Frappaolo C., Knowledge management, Capstone Publishing, Oxford, UK, 2002.
- [3] Đorđević-Boljanović J., Menadžment znanja, Data status, Beograd, 2009.
- [4] Turban E., Volonino L., Wood G., Information Technology for Management: Advancing Sustainable, Profitable Business Growth, Wiley, USA, 2013.
- [5] Chen H., Fuller S., Friedman C., Hersh W., Knowledge management, Data mining and Text mining in Medical Informatics, University of Arizona, 2007.
- [6] Johansson, Henry.J., [et al.], Business process reengineering: breakpoint strategies for market dominance, J.Wiley&Sons, UK, 1994.

Zaštita privatnosti – anonimizacija podataka

Privacy protection – data anonymization

Aleksandra Bradić-Martinović, Aleksandar Zdravković,
Institut ekonomskih nauka, Beograd

Ovaj rad je sastavni deo projekata 179001, 179015 i III47009 koje finansira Ministarstvo obrazovanja, nauke i tehnološkog razvoja Republike Srbije.

Rezime: Eksponencijalni rast digitalnih podataka u prethodnih dvadeset godina uzrokovao je povećanje mogućnosti ugrožavanja privatnosti pojedinca. Privatnost može biti ugrožena neovlašćenim pristupom i zloupotrebom u poslovnim informacionim sistemima, javnim institucijama, društvenim mrežama, ali i pristupom servisima u kojima se nalaze podaci koji su rezultat istraživanja u društvenim naukama. Širenjem *Open Access* inicijative, koja omogućuje otvoren pristup podacima, problem je dobio novu dimenziju.

U ovom radu predstavili smo postupak anonimizacije podataka, kao jedan od metoda koji istovremeno omogućuje zaštitu privatnosti uz nesmetano obavljanje istraživačkog procesa. U radu smo precizno definisali fazu istraživanja u kojoj je neophodno primeniti postupak anonimizacije, a zatim smo opisali osnovne dimenzije prirode podataka, od kojih zavisi izbor odgovarajuće tehnike. Centralni deo rada je posvećen najvažnijim tehnikama, posmatranim sa stanovišta ciljeva anonimizacije, *k*-anonimnost, *l*-raznolikost i *t*-bliskost.

Ključne reči: baze podataka, anonimizacija, privatnost, digitalne arhive podataka, istraživanje

Abstract: The exponential growth of digital data over the past twenty years has caused an increase possibility of invasion on individual privacy. Privacy can be compromised by unauthorized access and misuse of business information systems, public institutions, social networks, and access to services in which the data resulting from research in the social sciences. The expansion of *Open Access* initiative, which allows open access to the data, the problem is a new dimension.

In this paper we present the process of data anonymization, a method that simultaneously provides privacy to the smooth operation of the research process. We also define in a research phase in which it is necessary to apply the anonymization process, and then we describe the main dimensions of the nature of data, the choice of which depends on proper technique. The central part is devoted to the most important techniques, looking at the goals of anonymity, *k*-anonymity, *l*-diversity and *t*-closeness.

Index terms: databases, anonymization, privacy, digital data archive, research

1. UVOD

Doba digitalizacije omogućilo je relativno brzo, jednostavno i jeftino prikupljanje i čuvanje podataka o fizičkim i pravnim licima. Javne ustanove, zdravstvene institucije, finansijske organizacije i poslovni sistemi poseduju ogromne količine individualnih informacija koje se čuvaju u odgovarajućim bazama podataka. Mogućnosti upotrebe ovih informacija su praktično neograničene, ali se istovremeno mogu posmatrati dobri i loši aspekti ovog fenomena. Sa jedne strane, mora se uočiti veliki potencijal koji ove informacije pružaju za poboljšanje uslova života, kroz kreiranje novih i unapređenje postojećih usluga, kao i mogućnosti spoznaje navika, želja i preferencija korisnika i potrošača. Sa druge strane,

Aleksandra Bradić-Martinović, Institut ekonomskih nauka, Beograd, Zmaj Jovina 12, 11000 Beograd, Srbija, (abmartinovic@ien.bg.ac.rs)

Aleksandar Zdravković, Institut ekonomskih nauka, Beograd, Zmaj Jovina 12, 11000 Beograd, Srbija, (aleksandar.zdravkovic@ien.bg.ac.rs)

međutim ove informacije u velikoj meri zadiru u privatnost pojedinaca ili osetljive aspekte poslovanja preduzeća, a njihovo objavljivanje može uzrokovati niz vrlo loših posledica [3, str.1]. U većini demokratskih zemalja zaštita privatnih i ličnih podataka garantovana je zakonom [9, str.1].

Jedan od najpoznatijih primera ekstrakcije i zloupotrebe informacije iz baze digitalnih podataka je slučaj koji se desio kada je američka kompanija AOL (*America On-line*) objavila 2006. godine detaljne logove pretraživanja servisa AOL, za veliki broj svojih korisnika. Objavljivanje ovih informacija bilo je namenjeno u svrhe istraživanja, ali se ispostavilo da javno objavljivanje podataka podrazumeva mogućnost da svi imaju pristup podacima, a ne samo ograničena akademska zajednica. AOL nije vršio nikakvu obradu podataka, niti primenio neku od naprednijih tehnika anonimizacije (korisnici su samo imali numeričke identifikacione brojeve umesto imena), čime je otvorio potencijalnu mogućnost za otkrivanje identiteta korisnika od strane javnosti. Prvi slučaj otkrivanja identiteta desio se u slučaju Telme Arnold, sa ID #4417749, šezdesetdvogodišnja udovica iz Luburna, država Džordžija. Pronalaženje gospođe Arnold je dobijeno na osnovu ukrštanja nekoliko obeležja: specifične bolesti, godina i loše navike psa. Posledica ovog propusta dobila je veliki negativni odjek u javnosti, a protiv kompanije AOL je podneta tužba.

Komercijalno prikupljanje i objavljivanje podataka je samo jedna dimenzija, a pored nje postoje i podaci o ličnosti koje prikupljaju administrativne institucije i naučno-istraživačke institucije koje se bave istraživanjima u društvenim naukama (sociologija, psihologija i slično) ili prirodnim naukama, kao što je medicina. Obelodanjivanje ovih podataka, bez prethodne anonimizacije može u velikoj meri da ugrozi privatnost ispitanika.

U kontekstu čuvanja podataka posebno mora voditi računa o raspoloživosti podataka u digitalnom obliku, koji se jednostavno i brzo mogu distribuirati i umnožavati. Da bi se izbegle situacije navedene u prethodnom primeru potrebno je pažljivo anonimizirati podatke pre postupka diseminacije. Osnovni cilj anonimizacije je očuvanje privatnosti fizičkog ili pravnog lica. Prema proceni Baker i ostalih [9, str. 2] pitanje anonimizacije je pre svega pravne prirode, ali je direktno povezano sa mogućnostima i primenom informaciono-komunikacione tehnologije u privatnom i poslovnom životu. Osim toga, pitanju je veoma osetljivo i sa stanovišta etike.

U Republici Srbiji ova tema postaje posebno aktuelna nakon 2008. godine, kada je usvojen Zakon o zaštiti podataka o ličnosti, koji ima za cilj da u vezi sa obradom podataka o ličnosti, svakom fizičkom licu obezbedi ostvarivanje i zaštitu prava na privatnosti i ostalih prava i sloboda.

2. POJAM ANONIMIZACIJE

Anonimizacija je postupak trajnog i potpunog uklanjanja identifikacije iz podataka pretvaranjem ličnih informacija u agregatne podatke. Anonimizirani podaci ne mogu više biti povezani sa individualnim podacima na bilo koji način. Drugim rečima, kada se jednom ukloni lični identifikacioni element iz podataka, ti elementi se ne mogu ponovo povezati sa individualnim podacima ili sa konkretnom osobom.

Prikupljanje i diseminacija podataka, uz uključen proces anonimizacije, bez obzira na oblast u kojoj je primenjeno, može se predstaviti šemom na grafiku 1. U bazu podataka, arhiv ili skladište podataka ulaze kompletni podaci koji sadrže identifikatore. Istraživači (korisnici) imaju potrebu za određenim, relevantnim podacima, ali oni ne smeju prilaziti i koristiti podatke koji su povezani sa identifikatorima, pa je pre eksporta podataka potrebno izvršiti proces anonimizacije. Tek nakon toga je omogućeno da se sačuva privatnost lica na koja se podaci odnose.



Grafik 1: Prikupljanje i diseminacija podataka uz proces anonimizacije, prilagođeno [4, str. 1141]

Pre no što pristupimo objašnjenju tehnika kojima je moguće postići anonimizaciju podataka potrebno je pojasniti osnovne odrednice vezane za prikupljene podatke, koji se obično čuvaju u bazama podataka, predstavljenim u obliku tabela. Svaka kolona tabele, koja se definiše kao atribut predstavlja semantičku kategoriju informacije. Za lakšu analizu pravi se podela atributa na sledeće tri kategorije [7, str. 53]: *Identifikatori* su atributi koji mogu jedinstveno identifikovati pojedinca. U Srbiji najbolji primer ovakvog identifikatora je Jedinstveni matični broj građana (JMBG) za fizička lica ili Poreski identifikacioni broj (PIB) za pravna lica. Ova vrsta podataka se prva eliminiše iz seta podataka, jer su oni najlakši i siguran put ka narušavanju privatnosti. *Kvazi-identifikatori*, odnosno *ključni atributi* su oni koji u kombinaciji sa informacijama iz drugog izvora mogu da omoguće identifikaciju. Oni ne moraju da pruže identifikaciju svih osoba u bazi, ali mogu da ukažu na neku osobu ukoliko ona ima specifičan spoj ovih, ključnih atributa. Za fizička lica to su kombinacija atributa npr. datum, mesec i godina rođenja, pol, opština na kojoj žive, a za pravna lica to mogu biti tržišno učešće i grad u kome su registrovani. *Osetljivi atributi* su oni koji sadrže poverljive informacije, npr. bolest pacijenta i koje nikako ne smeju biti objavljene.

Prema istraživanjima neprofitne organizacije Educause [2] bez obzira na to što je pojam anonimizacije relativno lako definisati, to nije slučaj kada pokušamo da posmatramo podatke koji su anonimizirani. U većini slučajeva institucije se suočavaju sa izazovom i rizikom prilikom odabira odgovarajuće procesa koji bi im obezbedio pouzdanu anonimizaciju. Jedan od problema sa kojim se suočavaju je nedostatak precizne definicije pojma informacije koja omogućuje da se izvrši identifikacija (*Personally Identifiable Information - PII*) i informacije pomoću koje nije moguće izvršiti identifikaciju (*Non-PII*). Uzrok tome je složenost problema koji je nemoguće svesti na prost spisak ovih informacija (elemenata), zbog toga što informacije menjaju značenje u zavisnosti od konteksta u kome se nalaze, kao i raspoloživosti ostalih podataka sa kojima su povezani. Zbog ovog problema postupak anonimizacije predstavlja svojevrsan izazov za one koji se bave čuvanjem i publikovanjem setova podataka.

3. TEHNIKE ANONIMIZACIJE

Do sada je razvijen određen broj, manje ili više uspešnih tehnika anonimizacija podataka. Da bi se utvrdila odgovarajuća tehnika kojom će biti sprovedena anonimizacija podataka, potrebno je izvršiti obuhvat više dimenzija koje se odnose na ovaj proces [3, str. 53]

Priroda podataka

Postoji razlika u primeni tehnike anonimizacije u zavisnosti od toga da li su podaci:

- tabelarni podaci*, koji predstavljaju informacije o entitetima (npr. fizičkim ili pravnim licima), njihovim kvazi-identifikatorima (npr. godine, pol, poštanski broj) ili njihovim osetljivim podacima (npr. visina plate, bolesti);
- stavke seta podataka*, kojima su predstavljene transakcije (ili „tržišna korpa“), odnosno koji povezuju, na primer određene osobe i proizvode koje je ta osoba kupila u transakciji i
- grafički podaci, kojima su predstavljeni osetljivi odnosi između entiteta (npr. ljudi koji su članovi iste društvene mreže).

Pristupi anonimizaciji

Moguće tehnike anonimizacije podrazumevaju:

- a) *Uklanjanje (direktnog) identifikatora*, npr. imena ili adrese sa ciljem da se isključi mogućnost povezivanja konkretnih lica uključenih u bazu sa raspoloživim podacima. Uklanjanje identifikatora je najjednostavnija tehnika anonimizacije, a primer je predstavljen tabelom 1. Problem ove tehnike je u tome što ona često nije dovoljna da obezbedi privatnost lica čiji su podaci raspoloživi.

TABELA 1

Primer uklanjanja direktnog identifikatora

Sirovi podaci					
R.br.	Ime	Poštanski broj	Pol	Godine	Hobi
1	Jelena Petrović	11000	Ženski	38	Filmovi
2	Milena Janković	15000	Ženski	32	Filmovi
3	Dragan Todorović	12350	Muški	25	Skejt bord
4	Slavica Pešić	14000	Ženski	48	Heklanje
5	Antoniје Dišić	11050	Muški	62	Pecanje
6	Petar Ljubičić	11070	Muški	28	Fitnes

Anonimizirani podaci					
R.br.	Ime	Poštanski broj	Pol	Godine	Hobi
1	UKLONJEN IDENTIFIKATOR	11000	Ženski	38	Filmovi
2		15000	Ženski	32	Filmovi
3		12350	Muški	25	Skejt bord
4		14000	Ženski	48	Heklanje
5		11050	Muški	62	Pecanje
6		11070	Muški	28	Fitnes

U prethodnom primeru, bez obzira na to što je uklonjen identifikator, verovatno ne bi bilo teško zaključiti koji dvadesetpetogodišnji mladić u malom mestu intenzivno vozi skejt bord. Zbog toga se pored ove tehnike primenjuju i dopunske, sa ciljem smanjenja verovatnoće otkrivanja identiteta u slučajevima izuzetaka, što svakako nije muškarac od 28 godina, koji se bavi fitnessom na Novom Beogradu.

- b) *Agregiranje ili smanjenje preciznosti informacija ili promenljivih*. Ova tehnika se najčešće sprovodi tako što se podaci grupišu, npr. zamena datuma rođenja starosnim grupama. Primer je predstavljen tabelom 2.

TABELA 2

Agregiranje kvazi- identifikatora

Anonimizirani podaci					
R.br.	Ime	Poštanski broj	Pol	Godine	Hobi
1	UKLONJEN IDENTIFIKATOR	11000	Ženski	31-40	Filmovi
2		15000	Ženski	31-40	Filmovi
3		12350	Muški	21-30	Skejt bord
4		14000	Ženski	41-50	Heklanje
5		11050	Muški	61-70	Pecanje
6		11070	Muški	21-30	Fitnes

- c) *Peturbacija (uvođenje šuma) podataka za jednog ili više kvazi-identifikatora*. Kolone sa određenim obeležjima se generalizuju pomoću odgovarajuće distribucije, tako da više ne postoji veza između podataka koja bi omogućila otkrivanje identiteta [5, str. 10]. Ova tehnika spada u statističke tehnike. Primer je dat u tabeli 3.

TABELA 3

Mešanje podataka jednog ili više kvazi- identifikatora

Anonimizirani podaci					
R.br.	Ime	Poštanski broj	Pol	Godine	Hobi
1	UKLONJEN IDENTIFIKATOR	11000	Ženski	sredji	Filmovi
2		15000	Ženski	sredji	Filmovi
3		12350	Muški	mladi	Skejt bord
4		14000	Ženski	sredji	Heklanje
5		11050	Muški	stariji	Pecanje
6		11070	Muški	mladi	Fitnes

- d) *Permutacija podataka jednog ili više kvazi-identifikatora.* Ova tehnika se sprovodi tako što se u kolonama sa određenim obeležjima zamene vrednosti (npr. redosled), kao što je predstavljeno tabelom 4. Istraživač i dalje može da istražuje pojavu, jer poseduje podatke o frekvencijama, ali više nije u stanju da otkrije identitet osobe koja obuhvaćena istraživanjem.

TABELA 4

Permutacija podataka jednog ili više kvazi- identifikatora

Anonimizirani podaci					
R.br.	Ime	Poštanski broj	Pol	Godine	Hobi
1	UKLONJEN IDENTIFIKATOR	11000	Ženski	25	Filmovi
2		15000	Ženski	28	Filmovi
3		12350	Muški	32	Fitnes
4		14000	Ženski	38	Heklanje
5		11050	Muški	48	Pecanje
6		11070	Muški	62	Skejt bord

Ciljevi anonimizacije

Osim navedenih kriterijuma koji utiču na izbor odgovarajuće tehnike anonimizacije, različiti ciljevi zbog kojih postoji potreba da se obezbedi privatnost, takođe imaju uticaj na izbor, sve dok podaci ne postignu odgovarajuće osobine. Tu spadaju sledeće tehnike:

- a) *K-anonimnost.* Tehniku *k-anonymity* predložili su Samarati i Sweeny [8] da bi se sprečila mogućnost povezivanja kvazi-identifikatora, a sa ciljem da se zaštititi privatnost lica uključenih u bazu podataka. Smatra se da set podataka zadovoljava uslov *k-anonimnosti* ukoliko se svaki zapis u tabeli razlikuje od najmanje *k-1* ostalih zapisa, po pitanju svake kombinacije atributa kvazi-identifikatora. Prema tome, za svaku kombinaciju vrednosti kvazi-identifikatora u *k-anonimiziranoj* tabeli, postoji bar još *k* zapisa koji dele te vrednosti. Jednostavno rečeno, svaki kvazi-identifikator se mora pojaviti u identičnim kombinacijama u tabeli u najmanje *k* zapisa. Time je obezbeđeno da pojedinačni zapis ne može biti identifikovan metodom ukrštanja vrednosti obeležja (*cross tabulation*). Takva tabela naziva se *k-anonimizirana* tabela.

Primer: U narednoj tabeli prikazana je medicinska evidencija fiktivne bolnice u Njujorku. Tabela na sadrži direktne identifikatore poput imena, prezimena i broja socijalnog osiguranja. Raspoloživa obeležja podeljena su u dve grupe: 1) osetljivi (zdravstveno stanje) i 2) ne-osetljivi (poštanski broj, godine i zemlja porekla). Obeležja koja su označena kao osetljiva nikako ne smeju biti povezana sa bilo kojom jedinstvenom pojedinačnom vrednošću. Drugim rečima, ne smeju da omoguće otkrivanje identiteta pojedinačnog bolesnika.

TABELA 5
Primer k-anonimizacije

Sirovi podaci				
	Ne-osetljivi podaci			Osetljivi podaci
R.br.	Poštanski broj	Godine	Zemlja porekla	Stanje
1	13053	28	Rusija	Srčano oboljenje
2	13068	29	Amerika	Srčano oboljenje
3	13068	21	Japan	Virusna infekcija
4	13053	23	Amerika	Virusna infekcija
5	14853	50	Indija	Psihološki poremećaj
6	14853	55	Rusija	Srčano oboljenje
7	14850	47	Amerika	Virusna infekcija
8	14850	49	Amerika	Virusna infekcija
9	13053	31	Amerika	Psihološki poremećaj
10	13053	37	Indija	Psihološki poremećaj
11	13068	36	Japan	Psihološki poremećaj
12	13068	35	Amerika	Psihološki poremećaj

Anonimizirani podaci				
	Ne-osetljivi podaci			Osetljivi podaci
R.br.	Poštanski broj	Godine	Nacionalnost	Stanje
1	130**	<30	*	Srčano oboljenje
2	130**	<30	*	Srčano oboljenje
3	130**	<30	*	Virusna infekcija
4	130**	<30	*	Virusna infekcija
5	148**	≥40	*	Psihološki poremećaj
6	148**	≥40	*	Srčano oboljenje
7	148**	≥40	*	Virusna infekcija
8	148**	≥40	*	Virusna infekcija
9	130**	3*	*	Psihološki poremećaj
10	130**	3*	*	Psihološki poremećaj
11	130**	3*	*	Psihološki poremećaj
12	130**	3*	*	Psihološki poremećaj

Postupak anonimizacije ove tabele podrazumevao je: a) potpuno uklanjanje obeležja „Zemlja porekla“, b) zamena jedne ili više vrednosti u polju „Poštanski broj“ zvezdicom, tako da vrednost 1485* može biti bilo koji broj između 14850 i 14859 i c) druga cifra u obeležju „Godine“ zamenjena je, takođe, zvezdicom, tako da je vrednost 3* u rangu od 30 do 39 godina.

Postignut je cilj postupka anonimizacije u tabeli, jer se jedna ista kombinacija obeležja pojavljuje više puta (najmanje tri puta). U ovom slučaju u pitanju je tzv. 4-anonimizirana tabela, jer najmanje 4 zapisa dele isti kvazi-identifikator.

- b) *L-raznovrsnost (l-diversity)*. K-anonimnost često nije dovoljno dobra tehnika, zato što je uprkos prisustvu više identičnih kombinacija obeležja, ipak moguće otkriti identitet pojedinca. Ovo se javlja kao nedostatak raznolikosti kombinacije obeležja. Ukoliko, npr. imamo 10 različitih muškaraca, starih 54 godine, koji žive u malom mestu i oni imaju HIV, onda Piter može biti izložen sumnji/diskriminaciji ukoliko živi u malom mestu i ima 54 godine. Zbog toga je potrebno uvesti raznovrsnost u grupu formiranu na osnovu jednog obeležja. L-raznovrsnost znači da grupa sadrži *l* različitih vrednosti.

TABELA 6
Primer l-raznovrsnosti

Anonimizirani podaci				
	Ne-osetljivi podaci			Osetljivi podaci
R.br.	Poštanski broj	Godine	Nacionalnost	Stanje
1	1305*	≤40	*	Srčano oboljenje
2	1305*	≤40	*	Srčano oboljenje
3	1305*	≤40	*	Virusna infekcija
4	1305*	≤40	*	Virusna infekcija
5	1485*	>40	*	Psihološki poremećaj
6	1485*	>40	*	Srčano oboljenje
7	1485*	>40	*	Virusna infekcija
8	1485*	>40	*	Virusna infekcija
9	1306*	≤40	*	Psihološki poremećaj
10	1306*	≤40	*	Psihološki poremećaj
11	1306*	≤40	*	Psihološki poremećaj
12	1306*	≤40	*	Psihološki poremećaj

U tabeli 6 prikazan je primer tabele na kojoj je primenjena tehnika 3-raznolikosti. U poređenju sa k-anoniziranim tabelom 5, Bob ne može biti identifikovan ukoliko je samo Amerikanac, 31 godine starosti, koji živi na teritoriji poštanskog broja 13053 i ima psihološki poremećaj.

Postoji nekoliko vrsta l-raznolikosti: probablistička, entropijska, rekurzivna i više-atributivna.

- c) *T-bliskost (t-closeness)*. Ova tehnika predstavlja unapređenje prethodno opisanih tehnika i kreirana je sa ciljem da otkloni postojeće nedostatke. Privatnost se može izmeriti kao informaciona dobit istraživača (posmatrača). Pre no što dobije podatke na uvid istraživač već ima određena uverenja / verovanja o pojedinačnim vrednostima osetljivih obeležja (a' priori verovanja). Nakon uvida u podatke istraživač formira nova uverenja / verovanja (a' posteriori verovanja). Razlika koja postoji između a' priori i a' posteriori verovanja je informaciona dobit. Prema novim istraživanjima [5] informaciona dobit se može podeliti u dve celine: ona koja se ostvaruje na osnovu podataka o celoj populaciji i ona koja se ostvaruje na osnovu podataka o specifičnom pojedincu. Za ekvivalente skupove (klase) podataka kažemo da imaju t-bliskost ukoliko razlika između distribucija osetljivih obeležja u skupu i distribucije obeležja u celoj tabeli nije iznad utvrđenog praga t. Za tabelu kažemo da ima t-bliskost ukoliko svi njeni ekvivalentni skupovi imaju t-bliskost. Poenta tehnike je da se prilagodi distribucija pojedinačnih skupova distribuciji koja postoji u celoj populaciji.

Uprkos razvoju i primeni raznovrsnih tehnika kojima se vrši anonimizacija podataka u bazama, postoji niz metoda kojima se ugrožava privatnost. Veoma je važno istaći da se ni jedna od prethodno navedenih metoda ne može primenjivati automatski. Najbolje je da osoba koja je prikupila podatke sa određenim ciljem ima i uvid u rizik po pitanju ugrožavanja privatnosti koji može nastati ukoliko se izvrši diseminacija podataka na neodgovarajući način.

U praksi Arhiva podataka Velike Britanije (UK Data Archive – UKDA) uvek navode primer istraživanja u kome su direktori verskih škola iznosili mišljenje o načinu vođenja lokalne samouprave. Bez obzira na to da je izvršena anonimizacija podataka i uklonjeni svi direktni identifikatori, promakla je greška, jer se ispostavilo da je ugrožena privatnost izvesne žene koja je bila jedini direktor verske škole u tom okrugu, koja je ujedno ženskog pola. Problem je nastao zbog toga što je iznela vrlo oštre kritike, uz uverenje istraživača da niko neće moći da otkrije njen identitet. Ovaj primer pokazuje koliko je potrebno detaljno poznavati materiju na koju se podaci odnose, kao i specifičnost pojedinih kombinacija obeležja. To je često veoma teško obezbediti.

4. ZAKLJUČAK

Osetljivost privatnih podataka posebno je postala aktuelna pojavom digitalizacije i globalnog umrežavanja, jer je dostupnost podataka u savremenom svetu jednostavna i masovna. Podatke prikupljaju i distribuiraju, pre svega, kompanije, administracija i naučno-istraživački centri. Međutim, svi navedeni moraju strogo voditi računa o privatnosti ličnosti, jer su do sada zabeleženi slučajevi drastičnog ugrožavanja privatnosti od strane pojedinih kompanija, čime se krše i zakoni koji su uvedeni u velikom broju zemalja.

Jedan od načina da se privatnost ličnosti zaštiti, a da se podaci ipak distribuiraju, je primena metoda anonimizacije podataka. U ovom radu prikazali smo najvažnije metode koji se koriste za anonimizaciju podataka u razvijenim sistemima za čuvanje i distribuciju podataka, kao i njihove prednosti i nedostatke. Ukazali smo da sam proces i odabir konkretnog metoda anonimizacije zavisi od tipa podataka, od prostog uklanjanja direktnog identifikatora u slučaju manje osetljivih podataka, do primene neke od kompleksnijih metoda anonimizacije na indirektnim (kvazi) identifikatorima u slučaju kada je rizik otkrivanja identiteta visok. Ova oblast se stalno razvija, tako da se pojavljuju novi metodi kojima je moguće sve uspešnije zaštititi privatnost pojedinaca čiji su podaci raspoloživi i dostupni u bazi podataka.

Republika Srbija uspešno prati IT trendove u raznim oblastima poslovanja, administracije i nauke, tako da tema anonimizacije postaje vrlo aktuelna.

LITERATURA

- [1.] Machanavajjhala, J. Gehrke, D. Kifer, (2007), L-diversity: Privacy beyond k-anonymity, ACM Transactions on Knowledge Discovery from Data (TKDD), Volume 1, Issue 1, Article no. 3
- [2.] Educause, Guidelines for Data De-Identification, <http://www.educause.edu/ero>, poslednja poseta oktobar 2013.
- [3.] G. Cormode, D. Srivastava, (2010), Anonymized Data: Generation, Model, Usage, Book of Proceedings International Council for Open and Distance Education, SIGMOD '09
- [4.] G. Haddow, A. Bruce, S. Sathanandam, J. Wyatt, (2011), Nothing is really safe: a focus group study on the processes of anonymizing and sharing of health data for research purposes, International Journal of Public Health Policy and Health Services Research
- [5.] L. Liu, M. Kantarcioglu, B. Thuraisingham, (2008), The applicability of the perturbation based privacy preserving data mining for real-world data, Data & Knowledge Engineering 65, 5-21
- [6.] L. Ninghui, L. Tiancheng, V. Suresh, (2007), t-closeness: Privacy Beyond k-Anonymity and l-Diversity, Data Engineering, ICDE 2007. IEEE 23rd International Conference on Data Engineering
- [7.] M. Stanek, A. Jurišić, M. Babić, (2008), Anonimizacija podatkovnih baz, Bilt-ekon organ inform zdrav 2009, (25)2:53-59
- [8.] P. Samarati, L. Sweeney, (1998), Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, Technical report, CMU, SRI, 1998.
- [9.] S. Baker, K. Kuilwijk, W. Chang, D. Mah, (2003), Anonimization, Data-Matching and Privacy: A Case Study, Steptoe & Johnson LLP, Attorneys at Law

Izbor i implementacija platforme za e-učenje

The selection and implementation of a platform for e-learning

Bogdan Mirković, Fakultet za informacione tehnologije, Slobomir P Univerzitet

Drago Vidović, Fakultet za informacione tehnologije, Slobomir P Univerzitet

Apstrakt – Nagli razvoj informaciono-komunikacionih tehnologija (ICT) i njihova integracija u proces edukacije obilježavaju vrijeme u kome živimo i dovode do promjene u edukacionoj pedagogiji i paradigmi učenja. Sve veći broj obrazovnih institucija ima za cilj unaprijediti proces obrazovanja pokretanjem platforme za e-učenje, ulažući svoj trud u kreiranje i organizaciju resursa, vrijeme, finansijske i ljudske resurse i očekuju pozitivne rezultate. Pored zahtjeva za kompetencijom instruktora/mentora i polaznika u korišćenju tehnoloških mogućnosti na području savladavanja znanja i vještina koje se odnose na savremene pedagoške metode, izbor platforme za e-učenje može biti ključni faktor uspjeha. U ovom radu ćemo predstaviti neka od postojećih softverskih rešenja za podršku e-učenju, kao i moguće načine implementacije i upotrebe konkretne platforme.

Ključne reči – e-učenje, LCMS, LMS, Moodle.

Abstract - The rapid development of information and communication technologies (ICT) and integration of this technology into the educational process is characteristic of the time in which we live and lead to changes in educational pedagogy and learning paradigm. An increasing number of educational institutions involves in their educational processes platforms for e-learning. The aim of this introduction of e-learning is educational process is to reduce financial costs, human resource commitment, time and other resource of educational institutions. Educational institution from e-learning expect positive results. In addition to requirements for the competence of instructors, mentors and students and their skills and knowledge for use of ICT to contemporary learning methods, the choice of platform for e-learning can be a key success factor. In this paper we present some of the existing software solutions developed for support of e-learning and use of specific platform.

Index terms – e-Learning, LCMS, LMS, Moodle.

1. UVOD

Informaciona i komunikaciona tehnologija (ICT) postala je u veoma kratkom periodu temelj za izgradnju modernog društva. Mnoge države podrazumijevaju razumijevanje, posjedovanje vještina i koncepta ICT-a kao jezgre obrazovanja, jednako kao i čitanje, pisanje i računanje. Upravo i jedan od UNESCO-ovih ciljeva jeste i razvijenim i nerazvijenim zemljama osigurati pristup najboljim obrazovnim resursima potrebnim za pripremu mladih za aktivnu ulogu u modernom društvu.[1]

Svjedoci smo sve većeg uticaja informaciono-komunikacionih tehnologija na sve sfere i aspekte modernog društva. Sama pojava prefiksa „e-“ uz sve više različitih privrednih i društvenih aktivnosti upravo to potvrđuje. U skladu sa tim transformiše se i sam proces obrazovanja, te je tako danas

Bogdan Mirković – Fakultet za informacione tehnologije, Slobomir P Univerzitet, PF 70, 76000 Bijeljina, Republika Srpska, BiH (e-mail: bogdanmirkovic@yahoo.com)

Drago Vidović – Fakultet za informacione tehnologije, Slobomir P Univerzitet, PF 70, 76000 Bijeljina, Republika Srpska, BiH (e-mail: drago.vidovic@spu.ba)

potpuno nezamislivo visoko obrazovanje bez upotrebe računara i Interneta od strane studenata i nastavnika.

U skladu sa svojim resursima i mogućnostima korisnika (zaposlenima, učenicima, itd.), organizacije (obrazovne institucije, firme, itd.) odlučuju, ne o tome da li će primjenjivati, nego uolikoj mjeri će primjenjivati informaciono-komunikacione tehnologije u procesu učenja, jer je njigova prisutnost postala neminovnost današnjeg društva i samog obrazovanja. Analiza karakteristika organizacije i njenog okruženja je polazna osnova za efikasnost i efektivnost e-učenja, kako bi se izbjeglo neracionalno investiranje u opremu i kadrove i obezbjedio uspjeh.

2. ZNAČAJ I TREND OVI RAZVOJA E-UČENJA

Upotreba termina e-učenje rapidno raste. Takođe, javlja se sve veći broj definicija e-učenja i ovde ćemo izdvojiti samo neke od njih.

„E-učenje je interaktivno obrazovanje u kome je obrazovni sadržaj dostupan „on-line“ i daje automatski odgovor na obrazovne aktivnosti studenata. „On-line“ komunikacija sa ljudima može, ali i ne mora biti uključena.“ [2]

„E-učenje predstavlja nastavu, obuku ili obrazovni program koji se delimično ili potpuno distribuira elektronskim putem.“ [2]

„E-učenje je proces obrazovanja koji se sprovodi uz upotrebu nekog oblika informacione i komunikacione tehnologije, a s ciljem unapređenja procesa i rezultata obrazovanja.“ [3]

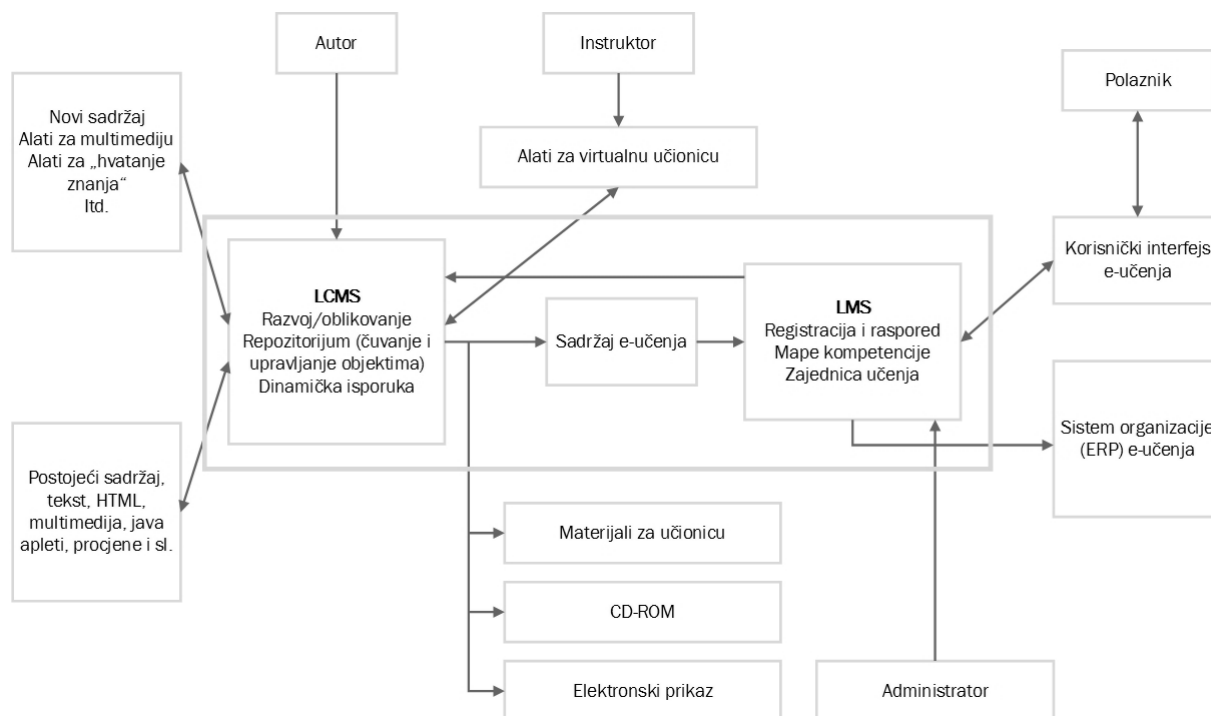
Važno je istaći da e-učenje nije alternativa postojećem obrazovnom procesu. Upotreba raznih multimedijalnih interaktivnih sadržaja i stvaranje virtuelnog okruženja kvalitetno dopunjuju obrazovni proces. [3] Uvođenjem e-učenja uspostavlja se bolja dvosmerna komunikacija i interakcija nastavnika i studenta. Nastavnici i studenti u novom virtuelnom okruženju aktivno sarađuju radi postizanja zadatih ciljeva. Na taj način dolazi do bitnih promjena u obrazovnom procesu u čije središte dolazi student, koji sada preuzima aktivnu ulogu i odgovornost za rezultate obrazovanja, dok se, s druge strane, uloga nastavnika pomjera od voditelja prema koordinatoru obrazovnog procesa.

Sa izuzetkom klasičnog, baziranog na štampanim materijalima, otvorenog i udaljenog učenja, može se dokazati da je pojavljivanje e-učenja direktno povezano sa razvojem i pristupom informacijama i infrastrukturi komunikacionih tehnologija. Bez pristupa ICT infrastrukturi, sposobnost za opstanak edukacionih aktivnosti u e-okruženju je ugrožena. Takođe, danas postoji veliki broj trendova ili pravaca – poslovnih i trendova učenja kao i tehnoloških trendova – koji utiču na koncept e-učenja, kao što su:

- povećana upotreba Web 2.0 tehnologija, koje podržavaju više socijalni i kolaborativni pristup učenju (tzv. Social Learning);
- neformalno učenje;
- samo-regulisano;
- mobilno učenje;
- brzi razvoj e-učenja.

Termin sa kojim se često susrećemo u izučavanju e-učenja je virtuelna učionica. Virtuelna učionica je nastavno okruženje locirano u kompjuterski generisanim i komunikacijski podržanim sistemima. Virtuelne učionice, kao jedan od oblika e-učenja, mogu poslužiti obrazovnim institucijama kao pogodan način za unapređenje njihovog obrazovnog procesa. Kao najpopularniji javno dostupan virtuelni svijet izdvaja se Second Life, koji je mnogim akademskim i obrazovnim ustanovama logičan izbor za izgradnju njihovog prvog virtuelnog obrazovnog objekta. Second Life je u skorije vrijeme postao virtuelna učionica za glavne koledže i univerzitete, uključujući Princeton, Rice University, Pepperdine University, University of Derby (UK) i Vassar, the Open University (UK). Pored Second Life-a, tu su i platforme, kao što su: WebEx (pruža realno vrijeme saradnje uz upotrebu interaktivne table, chat i VoIP tehnologiju koja omogućava razmjenu audio i video podataka) i Heritage Key

(platforma za istorijsko obrazovanje koja virtuelno obnavlja bitne arheološke sajtove iz čitavog svijeta, kao što je Tutankamonova grobnica, podržano je realno vrijeme komunikacije kroz slanje poruka i direktnu konverziju preko Skype-a).



Slika 1: Struktura sistema e-učenja

2.1 LMS i LCMS

U svijetu e-učenja postoje dva ključna pojma, koja je bitno razlikovati, a to su: sistem za upravljanje učenjem (engl. *LMS – Learning Management System*) i sistem za menadžment sadržaja učenja (engl. *LCMS – Learning Content Management System*). Generalno gledano, ovo su dva potpuno različita proizvoda, iako je tokom vremena došlo do približavanja koje bi u budućnosti moglo dovesti do fuzije ove dvije vrste proizvoda e-učenja. [4]

Na žalost, sličnost u akronimima dovela je do pojmovne zbrke kod kupaca. Razlikovanje LMS-a i LCMS-a dodatno je zakomplikovano činjenicom da postoji sve veće preklapanje između njih. Tako, na primjer, danas mnogi LCMS-ovi uključuju i podatke o polaznicima i njihovim aktivnostima te prate napredak i performanse. Sa druge strane, sve veći broj LMS-ova nudi i neki oblik šablona ili alata za pomoć pri razvoju sadržaja e-učenja.

Sistem za upravljanje učenjem (LMS) je infrastrukturna platforma kroz koju se sadržaj učenja isporučuje i njime upravlja. Sistem je snabdjeven softverskim alatima koji obavljaju mnoštvo funkcija vezanih za online i off-line administraciju kurseva u upravljanja performansama sistema.[9]

Primarna svrha LMS-a je automatizacija administriranja procesa učenja. LMS registruje korisnike, smješta i prati objekte učenja (lekcije, seminare, ...) u kataloge i bilježi podatke o polaznicima i njihovim aktivnostima prateći njihov napredak i performanse kroz sve tipove aktivnosti učenja. Takođe, izvještava o svim ostalim aktivnostima u sistemu. Najčešće ne posjeduje vlastiti autorski alat već se usredsređuje na upravljanje sadržajima razvijenim od strane drugih izvora. Glavne funkcije LMS-a su:

- prijava i naplata: ova funkcija procesira, provjerava i vrši autorizaciju prijave i izvodi naplatu (u slučaju komercijalnih javnih portala) kod polaznika;
- upravljanje procesom: ova programska funkcija upravlja procesom izvođenja obuke (lekcija, modula i kursa). Proces teče na sljedeći način:
 - o registrovani polaznik kroz pokreće određeni modul ili kurs;

- LMS traži dotični modul ili kurs te, kad ga nađe, poziva njegov prvi SCO (engl. *Sharable Content Object*);
- po završetku izvođenja, SCO poziva LMS koji poziva sljedeći SCO itd., sve do završetka dotičnog modula ili kursa.
- testiranje: prvi nivo na kojem se provodi testiranje su kviz pitanja uz lekcije. Drugi nivo su testovi nakon određenog broja lekcija ili modula, odnosno čitavog kursa.
- mentoring i praćenje: omogućavaju vođenje kursa, tj. upravljanje virtualnim razredom ili napretkom pojedinačnog polaznika. Mentor određuje članove virtualnog razreda prati napredak svakog člana na temelju statistika testova, komunicira sa svim članovima virtualnog razreda ili sa pojedinačnim polaznicima. Praćenje daje osnovu za izradu statistika. Prati strukturu tačnih i netačnih odgovora na nivou lekcije, modula i kursa, što omogućuje praćenje uspjeha polaznika. Svi zapisi se čuvaju u bazi podataka.
- korisničke funkcije: koje polazniku omogućavaju planiranje i praćenje vlastitog razvoja putem obrazovnih lanaca, komunikaciju sa mentorom, koordinatorom ili ostalim članovima grupe, uvid u najvažnije statuse njegovog računa kao i statistike njegovog rada, te pomoćne funkcije kao što je tehnička podrška, rječnik i sl.
- administratorske funkcije: koje obuhvataju aktivnosti kao što su kontrola sadržaja, evidencija, praćenje i izvještaje, komunikaciju, održavanje pregleda modula i kurseva, održavanje obrazovnih lanaca, *backup* i sl.

Ono što predstavlja bit svega je namjera da upravljaju okruženjem koje se edukuje, daju mjesto gdje sadržaj može biti organizovan i prezentovan korisnicima, tj. onima koji uče. Vrlo često se susrećemo i sa pojmom virtualno obrazovno okruženje (engl. *VLE – Virtual Learning Environment*) i treba napomenuti da je to sinonim pojmu LMS.

LCMS predstavlja multi-korisničko okruženje gdje ljudi koji rade na razvoju metoda učenja mogu kreirati, sačuvati, ponovo koristiti, upravljati i isporučiti digitalni sadržaj za učenje, a sve to iz centralnog skladišta podataka. LCMS omogućuje korisnicima, kao što su administratori, autori, predavači i predmetni stručnjaci stvaranje novih ili učitavanje postojećih nastavnih sadržaja. Alati uključuju proizvode kao što su MS Word i programi za obradu teksta, PowerPoint, instrukcijski softver (Authorware, ToolBook, itd.), flash animacije, HTML dokumente, ilustracije i grafike, audio isječke, video isječke i pitanja za kvizove.

3. KLJUČNA PITANJA IZBORA SISTEMA ZA UPRAVLJANJE UČENJEM

Izbor pravog sistema za upravljanje učenjem i postizanje uspješnosti realizacije ovog koraka, predstavlja uznemiravajući faktor za organizaciju. Ovo naročito važi za one organizacije koje su se oduvijek oslanjale na netehnološke metode obučavanja. Proces izbora LMS-a bi trebalo da bude deo sveukupnog strateškog plana e-učenja. Poznata je činjenica da pogrešno odabran LMS može jako loše uticati na finansijsku sliku u organizaciji.

Postoji nekoliko osnovnih opcija prilikom određivanja vašeg LMS-a [5]. To su:

- kupovina komercijalnog LMS rješenja;
- izrada LMS-a prema vašim potrebama; i u skorije vrijeme;
- prihvatanje i prilagođavanje *Open Source* rješenje.

Podrazumijeva se da najbolja opcija zavisi od vašeg budžeta, vremena projektovanja, koliko dobro raspoloživi LMS program podržava vaše jedinstvene potrebe i od vašeg cjelokupnog plana razvoja e-učenja. Najčešća situacija je da nijedan raspoloživi LMS neće imati sve karakteristike, niti izvoditi sve zahtjevane funkcije koje 100% ispunjavaju vaše potrebe.

Odgovori na sledeća pitanja mogu pomoći prilikom donošenja odluke o izboru odgovarajućeg LMS-a:

- Da li je u vašoj ustanovi već odabran LMS za potrebe čitave ustanove?
- Da li je željeni LMS dostupan (kupljen ili instaliran lokalno ili smješten na nekom drugom serveru)?
- Postoji li neka druga ustanova koja vam može osigurati upotrebu LMS-a i pod kojim uslovima?
- Ako odaberete sistem koji ne uključuje tehničku podršku, imate li tehničko znanje i resurse da sami pružite podršku učenicima i nastavnicima?
- Ima li vaša ustanova mogućnosti za razvoj vlastitog rješenja? Da li bi bilo održivo? Ima li za to potrebe?
- Podržava li vaša ustanova *Open Source* kulturu i zapošljava li mnogo motivisanih *Open Source* razvojnih programera?

Ključni koraci koje treba preduzeti kod pravilnog odabira LMS sistema su [6]:

- funkcionalne karakteristike;
- odluka o tome kakva će biti implementacija rješenja;
- obuka korisnika za rad u e-learning LMS sistemu.

Na odluku o izboru platforme za e-učenje, pozitivno ili negativno, utiče i ocjena infrastrukture organizacije u kojoj se vrši implementacija e-učenja. [8] Potrebno je obratiti pažnju na:

- Računarsku mrežu: kakav pristup web-u organizacija trenutno ima? Postoje li trenutno u organizaciji projekti (kao što su web konferencije) koji mogu poboljšati infrastrukturu virtualne učionice?
- Učionice: da li organizacija ima učionice ili prostorije koje se mogu koristiti za inicijative e-učenja? Iako e-učenje ima potencijal da se odvija bilo gdje i bilo kada, neka istraživanja pokazuju da učenici postižu bolje rezultate ako materijalima za učenje ne pristupaju sa uobičajenog radnog mjesta.
- Osnovne kompetencije: postoje li vještine i sposobnosti koje bi olakšale pribavljanje rješenja e-učenja? Na primjer, postoji li već jaka jedinica za izradu web stranica i e-business inicijativa na webu?

3.1. Funkcionalne karakteristike

Prva stvar koju je potrebno razmotriti prilikom uvođenja sistema za edukaciju na daljinu, bez obzira da li se radi o *Open Source* „besplatnim“ ili komercijalnim software-ima ili sistemima, je razmatranje funkcionalnih karakteristika.

Najčešća greška se pojavljuje kada IT odjeljenje preuzme neki gotov LMS paket sa interneta, instalira ga, a zatim počne da pregleda funkcionalnosti. Potrebno je razmotriti koje su osnovne karakteristike koje rješenje mora da već sadrži u sebi bez posebne potrebe za dopunskim modifikacijama, odnosno napraviti spisak svih karakteristika koje sistem obavezno mora da ima, koje bi bilo lijepo da ima, i koje su nepotrebne i opterećuju sistem.

U slučaju da postojećim rješenjima nije moguće zadovoljiti sve funkcionalne karakteristike koje su navedene kao obavezne, potrebno je razmotriti postojanje mogućnosti razvoja takvih karakteristika kroz interni razvojni tim, ili kroz partnera koji podržava implementaciju. Investicije potrošene na modifikaciju i prilagođavanje, su trošak koji nikad neće biti povraćen u slučaju da se organizacije odluče za prelazak na neki drugi sistem. Zato je potrebno dobro razmisliti o modifikaciji čak i ako je u startu isplativija ta solucija.

3.2. Odluka o tome kakva će biti implementacija rješenja

Potrebno je donijeti odluku da li će sistem biti interno hostovan i održavan, ili će kompletna usluga održavanja hostovanja biti poverena partneru koji radi implementaciju. Ključna stvar leži u prednostima i manama jednog i drugog rješenja:

- interno hostovanje: sve tehničke aspekte i održavanje samog sistema može biti i dalje u rukama partnera koji pruža podršku, ali u tom slučaju mu je potrebno omogućiti VPN pristup sistemu unutar organizacije. Druga opcija je da zbog smanjenj troškova sama IT služba pruža potpunu podršku u kom slučaju se gubi onaj najbitniji deo a to je pomoć u samoj „sadržajnoj“ implementaciji.
- hostovano kod partnera: sistem je praktično postavljen i održavan od strane partnera. Organizacija koja dobija ovakvu vrstu usluge praktično se fokusira samo na implementaciju sadržaja i materijala, bez nepotrebnog opterećenaj tehničkim detaljima funkcionisanja sistema. Praktično u toj organizaciji najviši nivo uloge na LMS sistemu je predavač. Uloga sistem administratora ostaje u partnerskoj kompaniji, koja ima omogućen pristup serveru. Ovakva podjela puno doprinosi brzini implementacije LMS sistema, jer praktično dobra partnerska kompanija mora da služi i kao mentor kreiranja sadržaja, uvođenja novih kurseva u organizaciju i slično. Mane ovakvog načina poslovanja je to što partnerska kuća ima direktan uvid u sva dešavanja na platformi, što nekad nije poželjno i pored potpisanih ugovora o čuvanju poslovne tajne.

3.3. Obuka korisnika za rad u e-learning LMS sistemu

Najkompleksniji deo implementacije, ako je vaš izbor *Open Source* platforma, je obuka korisnika. Budući da većina „besplatnih“ LMS sistema ne nudi dovoljno kvalitetnu pomoć prilikom planiranja i izvođenja obuke za korisnike LMS sistema (predavače, menadžere, studente) kod implementacije *Open Source* e-learning LMS sistema treba prilikom odabira pažljivo procijeniti ukupne troškove i potrebne resurse za uspješnu obuku korisnika, kako onih koji kreiraju kurseve i sadržaj, tako i krajnjih korisnika koji su polaznici.

Na ovom mjestu organizacije često se prave greške, ne uzimajući ove troškove u obzir, razmišljajući da je to na kraju lako, a ne uviđajući da je to ključno mjesto u projektu u kojem se najčešće lomi uspješnost uvođenja bilo kog novog sistema.

3.4 Metode ocjenjivanja i poređenja sistema za upravljanje učenjem

Primjenom sistema za upravljanje učenjem u organizaciji, on postaje specifičan deo i vrlo teško ga je zamijeniti. Dakle, potrebno je odvojiti dovoljno vremena i resursa i proći sve faze metodologije za odabir sistema za upravljanje učenjem kako bi se smanjila mogućnost pogrešnog odabira i postigla vremenska ušteda u implemenaciji sistema.

CARNet-ova metoda ocjenjivanja velikim brojem kriterijuma obuhvata sledeće grupe kriterijuma koji se vrednuju:

- radnu okolinu studenta;
- radnu okolinu autora materijala;
- radnu okolinu predavača i pedagoške alate;
- administraciju;
- zahtjeve za domaće tržište (jezičke pakete, valute);
- tehničke preduslove;
- opšta svojstva.

Mjerenje prednosti pojedinih karakteristika između potencijalnih sistema za upravljanje učenjem poželjno je izvršiti kroz objektivni alat (kao što je na primjer EduTools Summative alat koji nudi dva načina poređenja: putem automatizovanog poređenja različitih LMS na osnovu informacija koje poseduje sama EduTools zajednica, putem interaktivnog automatizovanog sistema koji omogućava da

pored poredjenja tehničkih i funkcionalnih karakteristika i sam stav korisnika dođe do izražaja kroz lično ocjenjivanje zastupljenosti pojedinih kriterijuma u svakom alatu).

Takođe, moguće je koristiti i LMS KnowledgeBase Brandon Hall Research koji sadrži više od 130 LMS sistema sa više od 200 pojedinačnih karakteristika svakog od njih. Na njihovoj web stranici (www.brandon-hall.com) postoji i alat za poredjenje odabranih sistema. Slične usluge se nude i na web stranici Bersin&Associates (www.bersin.com).

Poželjno je koristiti alate koji imaju mogućnost uvođenja težinskih koeficijenata za svaki kriterijum kojim bi se odredila važnost pojedinog kriterijuma za organizaciju i time dobio najbolji alat sa aspekta organizacije za koju se alat bira.

4. IMPLEMENTACIJA SISTEMA ZA E-UČENJE ZASNOVANA NA OPEN SOURCE RJEŠENJIMA

Veliki broj ljudi kada čuje termin *Open Source* povezuje sa terminom “besplatan softver”. Tu nastaje prva velika greška vezana za implementaciju bilo kog *Open Source* softvera, pa i LMS-a. *Open Source* je za bilo koju ozbiljnu implementaciju sve osim besplatan jer ni za jedan softver, a posebno za informacioni sistem ne treba zanemariti ukupne troškove vlasništva takvog softvera ili sistema (engl. *TCO – Total Cost of Ownership – ukupni troškovi vlasništva*). U najvećem broju slučajeva prilikom implementacije jednog sistema za e-učenje potrebno je angažovati partnersku firmu koja poznaje LMS koji se implementira i koji će svojim savjetima da spriječi nepotrebno gubljenje resursa (vrijeme zaposlenih, novac, itd.). [7]

4.1. Claroline

Claroline je besplatan LMS, razvijen u PHP programskom jeziku sa podrškom za MySQL bazu podataka. Originalno je razvijen od strane IMP (*Institut de Pedagogie universitaire et des multimedias of the UCL*), njegov današnji razvoj je rezultat saradnje između IPM i ECAM (*Institut Supérieur Industriel*), odnosno, konzorcijuma univerzita iz cijelog svijeta.

Osnovna prednost Claroline je da je razvijen od strane predavača, za predavače, odnosno da se najbitniji koncept razvoja usmjerava ka pedagogiji i kvalitetnom sadržajnom korišćenju tehničkih alata i pedagogiji kako bi se podstakao ovaj vid edukacije na univerzitetskom nivou.

Claroline podržava standarde kao što su SCORM i IMS/QTI za razmjenu sadržaja odnosno, mogućnosti da se dijelovi ili kompletni kurservi izrade u drugim programima koji podržavaju navedene standarde i zatim uvezu u Claroline.

Neke od osnovnih funkcionalnosti koje su razvijene za Claroline su danas standard za sve LMS sisteme, a to su:

- upravljanje dokumentima;
- kreiranje online vježbi i testova;
- razvoj putanje učenja (engl. *learning path*);
- koordinacija grupnog rada.

4.2 Moodle

Moodle je web aplikacija napisana u PHP-u, koja podržava više vrsta baza podataka (posebno MySQL i PostgreSQL). Sistem je preveden na 65 jezika, a Moodle se trenutno koristi u 163 zemlje. Kako je Moodle projekat otvorenog koda, ima brojnu zajednicu korisnika i izvrsnu dokumentaciju i podršku. Mnogi korisnici aktivno učestvuju u stvaranju Moodle-a, radom na postojećim dijelovima, izradom novih modula, testiranjem proizvoda ili pružanjem podrške korisnicima.

Moodle sistem za upravljanje učenjem pruža profesorima punu računarsku podršku pri organizaciji i izvođenju online kursa, pružajući im mogućnosti:

- izrade velikog broja kurseva na jednom sistemu;
- planiranje kurseva – raspored aktivnosti, kalendar;
- upravljanje korisnicima, korisničkim ulogama i grupama korisnika na kursu;
- rad sa već postojećim datotekama i obrazovnim sadržajem;
- provjera znanja i ocjenjivanje korisnika;
- zadavanje online zadataka;
- praćenje aktivnosti korisnika;
- mnogobrojni alati za komunikaciju i kolaboraciju među korisnicima;
- upravljanje sistemom – izrada sigurnosnih kopija, statistike, logovi.
- obiman sistem pomoći.

5. ZAKLJUČAK

Promjene i konstantno napredovanje informaciono-komunikacione tehnologije (ICT) značajno su promijenili proces obrazovanja, što je rezultiralo porastom broja sistema za obuku putem Interneta. Sve značajne obrazovne institucije u svom radu koriste e-učenje u nekom obliku i time održavaju konkurentnost i ispunjavaju zahtjeve savremenog društva, odnosno savremenog procesa obrazovanja.

Sama primjena informaciono-komunikacionih tehnologija u procesu obrazovanja i implementacija e-učenja nije garant održivosti i ostvarenja cilja obrazovnih ustanova, sama po sebi, ukoliko nije pažljivo uvedena kroz detaljnu analizu same institucije i njenih potreba i specifičnost za adekvatnom platformom.

LITERATURA

- [1.] J. Maravić, "CARNet – Časopis Edupoint", UNESCOV dokument o informacijskog i komunikacijskog tehnologiji i obrazovanju, vol. III, no. 19, 2003, pp. 23-30
- [2.] J. Sloan, "CARNet – Časopis Edupoint", "E-learning je potrebno analizirati", vol. III, no. 19., 2003, pp. 17-22
- [3.] D. Vidović, „INFOTEH-JAHORINA“, Projektovanje sistema za udaljeno učenje i testnog centra, vol. 10, Ref. F-41, 2011, pp. 1084-1087
- [4.] „LMS – Learning Management System“, (pristup: 05.11.2013.), [dostupno na <http://www.edukacija.net/Default.aspx?tabId=9&ItemID=46&mid=48>]
- [5.] S. Naidu, E-learning – A Guidebook of Principles, Procedures and Practices, 2nd Revised Edition, Australia, Melbourne, 2003
- [6.] "Pravilan odabir Open Source LMS sistema", (pristup: 07.11.2013.), [dostupno na <http://www.e-learning.rs/pravilan-odabir-open-source-lms-sistema>]
- [7.] G. Bubaš, „CARNet - ČasopisEdupoint“, Metodika i komunikacija u obrazovanju na daljinu, vol. III, no. 19, pp. 12-15
- [8.] "Kako započeti LMS projekat?", (pristup: 05.11.2013.), [dostupno na <http://www.edukacija.net/Default.aspx?tabId=9&ItemID=239&mid=54>]
- [9.] "Sustavi za upravljanje učenjem", (pristup: 07.11.2013.), [dostupno na http://elacd.carnet.hr/index.php/Sustavi_za_upravljanje_ucenjem]

Automatsko generisanje i upotreba okvira za razvoj JAVA desktop aplikacija

Automatic generation and framework use for developing JAVA desktop applications

Drago Vidović, Fakultet za informacione tehnologije, Slobomir P Univerzitet

Bogdan Mirković, Fakultet za informacione tehnologije, Slobomir P Univerzitet,

Apstrakt – Svedoci smo pojave sve većeg broja firmi na tržištu softvera, koje su spremne u svakom trenutku da odgovore na zahtjeve klijenta i razviju poslovne aplikacije sa neznatnim razlikama u kvalitetu. U ovakvom okruženju presudan faktor za dobijanje posla postaje vrijeme, kao i cijena koja zavisi od vremena koje je potrebno da se projekat realizuje i dobije tražena poslovna aplikacija. Kako smanjiti vrijeme izrade softvera, povećati produktivnost i time postati konkurentniji? Upotreba automatskih generatora koda i stvaranje okvira za razvoj aplikacija, može biti odgovor na postavljeno pitanje. U ovom radu ćemo govoriti o razvoju okvira i njegovoj upotrebi za kreiranje poslovnih desktop aplikacija u JAVA programskom jeziku. Okvir predstavlja skelet aplikacije, koji sadrži kompletan kôd osnovnih funkcija celog sistema, a koji se može prilagođavati za potrebe konkretne aplikacije.

Ključne reči – automatsko programiranje, desktop aplikacija, okvir.

Abstract – Nowadays, there are growing number of companies in the software market which are ready at any time to respond to new software request from clients nad to develop applications with slight differences in the quality. In this environment, a crucial factor in getting a business opportunities for software development becomes both time and price that depends on the time needed to realize the software. The question is, how to reduce software development time, increase productivity and thus become more competitive? The use of automatic code generator and create a framwork for software development may be the answer to the above question. In this paper we present a work for development of a framework and its use for the creation of business desktop applications in programming language Java. The presented framework contains a complete code of basic functions of the whole system which can be customize for specific applications development.

Index terms – automatic programming, desktop application, framework.

1. UVOD

Sve veća konkurentnost na tržištu razvoja softvera nameće potrebu za konstantnim usavršavanjem metoda izrade, testiranja, dokumentovanja i održavanja softvera. Danas su softverski proizvodi često visokog stepena složenosti, sa neretko desetinama hiljada datoteka i milionima linija programskog koda. Uspješna konstrukcija i održavanje ovakvih programa zahtjeva primjenu određenih tehnika i upotrebu odgovarajućih alata, stvaranje okvira koji znatno olakšavaju i unapređuju rad na razvoju objektno-orijentisanih softvera. Termin okvir (engl. *framework*) najčešće se koristi u softverskom inženjerstvu, kada se govori o projektovanju i implementaciji objektno-orijentisanog softvera. Okvir možemo shvatiti kao skelet aplikacije u kome se nalazi kompletan kod osnovnih funkcija cijelog

Drago Vidović – Fakultet za informacione tehnologije, Slobomir P Univerzitet, PF 70, 76000 Bijeljina, Republika Srpska, BiH (e-mail: drago.vidovic@spu.ba)

Bogdan Mirković – Fakultet za informacione tehnologije, Slobomir P Univerzitet, PF 70, 76000 Bijeljina, Republika Srpska, BiH (e-mail: bogdanmirkovic@yahoo.com)

sistema i koji se prilagođava potrebama konkretne aplikacije. Dakle, razvoj aplikacije upotrebom okvira koji je dobro struktuiran i koji implementira standardne funkcionalnosti koje se javljaju u većini desktop aplikacija koje zahtijevaju bazu podataka, može predstavljati povoljno područje za usavršavanje procesa izrade ovakvih aplikacija, primjenama različitih tehnika automatskog generisanja koda na generisanje ovako jednog okvira.

Od svog početka, 1950-1960, softversko inženjerstvo nastoji poboljšati i unaprijediti proces razvoja softvera, dajući niz alata i tehnika koji su dovodili do inkrementalnog poboljšanja produktivnosti i kvalitete razvijenog softvera. Ovo je još uvijek mlada disciplina, koja je u neprestanom razvoju i kao trenutne tendencije u razvoju softvera navodi:

- aspekte (engl. aspects): obezbeđuju alate za dodavanje ili uklanjanje generiškog koda;
- agilno programiranje: koje uključuje brz razvoj funkcionalnih jedinica softvera, iterativnost, manje dokumentacije, timski rad, saradnju sa klijentom i prilagođavanje aplikacije tokom životnog ciklusa projekta;
- eksperimentalno softversko inženjerstvo: grana koja se bavi pravljenjem eksperimenata na softveru, prikupljanjem podataka iz eksperimenata i postavljanjem zakona i teorema na osnovu tih podataka. To je potpuno naučna metodologija;
- razvoj vođen modelom (engl. model-driven development): razvoj tekstualnih i grafičkih alata koji koriste transformacije modela i generisanje koda da bi generisali dobor organizovane fragmente koda koji služe kao osnova za kreiranje aplikacije;
- okviri: takozvane proizvodne linije softvera – sistematičan način da se proizvede familija softverskih sistema, umjesto jednog individualnog proizvoda. Ovaj postupak zahtijeva dobro planiranu i razvijenu ponovnu upotrebu koda. U suštini, to je pokušaj industrijalizacije razvoja softverskog sistema.

2. UPOTREBA OKVIRA U RAZVOJU APLIKACIJA

Na samom početku svog razvoja i upotrebe, prvi okviri koji su razvijani bili su kreirani za potrebe kreiranja korisničkog interfejsa. Prvi objektno-orijentisani okviri koji su zaista smatrani za okvire bili su MVC (engl. *Model View Controller*) za Smalltalk i MacApp za Epl aplikacije. Okviri su slični softverskim bibliotekama, jer i jedni i drugi predstavljaju kod koji se može ponovo koristiti upkovan u API (engl. *Application Programming Interface*). Biblioteke služe za ponovnu upotrebu koda, ali ne podrazumijevaju neku konkretnu strukturu aplikacije, već samo obezbjeđuju funkcionalnosti pomoću kojih aplikacija radi. Sa druge strane, okviri predstavljaju apstraktnu specifikaciju nekog tipa aplikacije. [1]

Ne postoji jedinstvena definicija okvira, i skoro svako ko se bavio ovom granom softverskog inženjerstva dao je svoju definiciju okvira. Uzimajući u obzir veliki broj autora koji su dali svoje definicije, jasan je pokazatelj da nema slaganja oko toga šta je zapravo okvir aplikacije. Ovde ćemo dati samo neke od definicija okvira:

- „Okvir je konkretno, ali nekompletno rješenje za probleme velike važnosti koji se ponavljaju.“ [1]
- “Okviri su generatori aplikacija određenog domena.” [2]
- “Okvir je apstrakcija u kojoj generički kod za standardne funkcionalnosti može selektivno da se prekrije korisničkim kodom kojom će se dobiti specifične funkcionalnosti.” [1]
- “Okviri su aplikacije na visokom nivou apstrakcije koje se odnose na jedan određen domen, a koje mogu biti „prekrojene“ za individualnu primjenu.” [3]

Upotreba okvira nam pomaže u sledećim aspektima razvoja aplikacije:

- pojednostavljuje rad sa složenim tehnologijama;
- vezuje pojedinačne komponente/objekte u korisne cjeline;

- primorava tim da kod implementira konzistentno, čime se obično postiže fleksibilnija aplikacija sa manje grešaka;
- kod je jednostavniji za održavanje.

Dobro isprojektovan okvir bi trebalo da ima sledeće osobine:

- uz okvire, programeri ne moraju da počinju od nule svaki put kada pišu konkretnu aplikaciju. Okviri pomažu programerima da nađu rešenje za domene problema i da lakše održavaju ta rešenja. Takođe obezbeđuju dobro projektovanu infrastrukturu tako da kada se kreiraju i dodaju novi dijelovi, mogu biti integrisani uz minimalan uticaj na druge dijelove okvira; [4]
- koliko god elegantan ili dobro projektovan okvir bio, on neće biti upotrebljen osim ako cijena razumijevanja i zatim korišćenja njegovih apstrakcija nije niža nego procijenjena cijena pisanja tih funkcionalnosti od nule, tj. bez upotrebe okvira; [4]
- da bi uspeo, okvir mora biti kompletiran, fleksibilan, proširiv i razumljiv; [5]
- okvir treba da nađe načine da minimizira potencijalne greške programera i da proširi međuplatformsku prenosivost; [6]
- okviri nameću model koji zahtijeva od programera da se prilagode. Sa druge strane, okviri povećavaju produktivnost programera; [4]
- ciljevi izrade okvira bi trebalo da budu izrada aplikacija od postojećih komponenti, upotreba malog broja tipova komponenti iznova i iznova i pisanje što je manje moguće koda. Krajnji cilj u kreiranju okvira je da programer uopšte ne piše kod; [5]

Dakle, upotreba dobro isprojektovanog okvira, koji je kreiran kroz iskustvo u radu na razvoju većeg broja aplikacija sličnog ili istog domena, znatno unapređuje i doprinosi kvalitetu razvoja softverskog rešenja. Trenutak kada je okvir dostigao najvišu tačku njegovog kvaliteta i daje najbolje rezultate njegove upotrebe, ima najbolju strukturu, najlakši za upotrebu i najrazumljiviji za sve članove tima, je trenutak kada bi trebalo razmišljati o automatskom generisanju okvira za potrebe aplikacije sa specifičnim domenom problema koji rješava.

3. AUTOMATSKO PROGRAMIRANJE

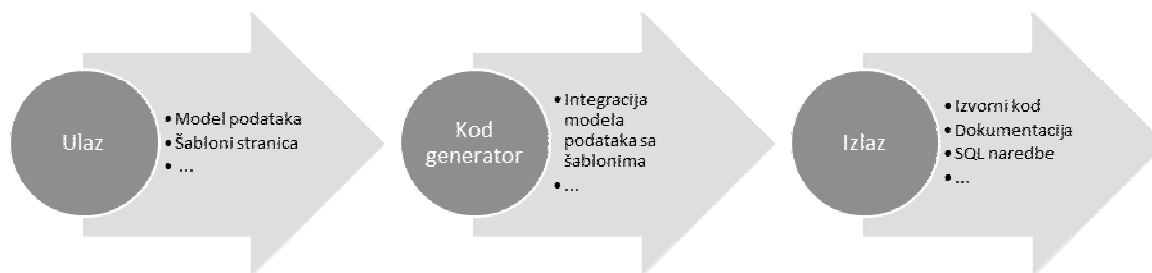
Radeći na razvoju aplikacija i uočavanje dijelove koji se ponavljaju, pred programera ili razvojni tim treba da nametne pitanje prelaska na automatsko generisanje koda, koje će u većem broju slučajeva doprinijeti unapređenju razvoja.

Automatsko programiranje predstavlja kompjutersko programiranje u kome neki mehanizam generiše program umjesto da čovjek, programer, piše kod. [8]

Prvi korak, koji je potrebno proći do primjene automatskog načina generisanja koda, jeste da jasno shvatite i priznate da postoji problem, problem sa generisanjem koda u vašem timu, nakon čega će se nametnuti pitanje "Da li je automatsko generisanje koda pravo rešenje za vas?". Navešćemo neke moguće situacije koji će vas natjerati da pozitivno odgovorite na postavljeno pitanje. Imate potrebu da napišete 1,050 Java fajlova (fajlovi sa sličnom strukturom), a vaš tim izbegava monotoniju u svom radu. Dakle, velika količina fajlova koje je potrebno kreirati i izbjegavanje monotonije, dovoljni su razlozi kako biste imali pozitivan stav o potrebi za automatskim načinom generisanja koda.

Naravno, ovde moramo napomenuti da upotreba automatskog generisanja koda i ne mora uvek biti vaša potreba i moguće rešenje, kao što su na primer projekti sa manjim brojem potrebnih fajlova koji pri tom nemaju sličnu strukturu.

Kada se govori o automatskom programiranju, neophodno je definisati i pojam kod generator: "*Kod generator predstavlja softversku komponentu koja služi za automatizovano pravljenje drugih softverskih komponenti*".



Slika 1. – Opšti mode kod generatora

Upravo kod generatori mogu dati rješenje za automatsko generisanje velike količine koda koji se ponavlja, samim tim poboljšati radnu atmosferu u timu, koja bi mogla biti narušena pojavom monotonije u radu i kao rezultat dati okvir, deo aplikacije ili bilo koji odgovarajući izlaz koji je zahtijevao minimalne napore oko obazbeđivanja potrebnih ulaznih podataka.

Kod generatore na višem nivou možemo podijeliti u dvije odvojene kategorije [8]:

- Pasivni kod generatori: Kod ovog modela, generatori proizvode skup kodova koje inženjeri dalje kako žele menjaju ili brišu. Pasivni kod generatori ne zadržavaju nikakvu odgovornost za krajnji kod, bilo u kraćem ili dužem vremenskom periodu. Primera radi, dobro poznati i često korišćeni, “Čarobnjaci” (engl. wizards) u razvojnim okruženjima (engl. IDE – Integrate Development Environment) spadaju u grupu pasivnih kod generatora.
- Aktivni kod generatori: Ovaj model kod generatora zadržava odgovornost za kod u dužem periodu, dozvoljavajući pokretanje procesa generisanje više puta tokom vremena, za isti deo izlaza. Ako postoji potreba za promenom, članovi tima mogu uneti nove parametre (ulazne podatke za generator) i ponovo ga pokrenuti, kako bi izmenili izlazni kod.

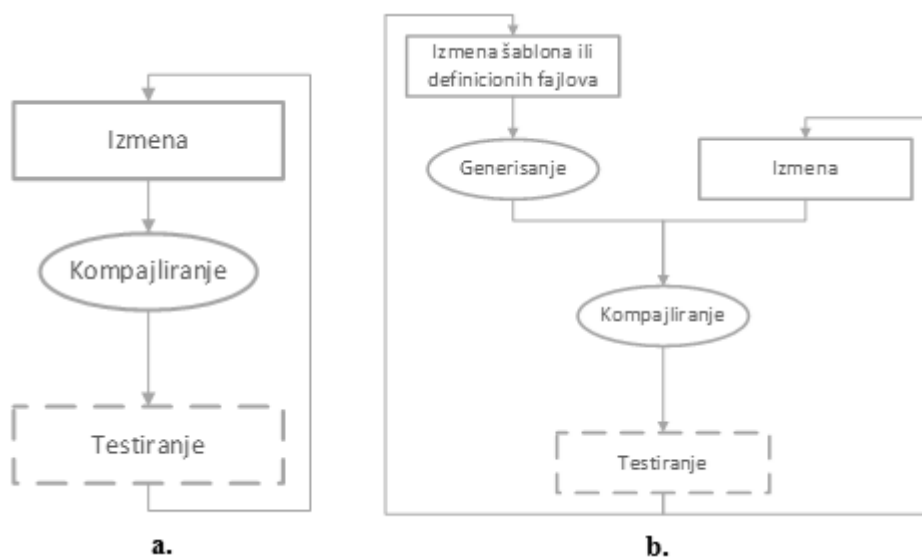
Automatsko generisanje koda, pruža značajne prednosti za softverske inženjere na svim nivoima razvoja softverskog proizvoda, počev od nivoa za pristupa bazama podataka, korisničkog interfejsa, jediničnih testova, RPC sloja do izgradnje dokumentacije. Ove prednosti se prije svega uključuju:

- Kvalitet: Veliko količine ručno napisanog koda imaju tu tendenciju nedoslednosti napisanog koda, iz razloga što inženjeri pronalaze novije ili bolje pristupe za njihov rad. Dok, kod generatori iz šablona gotovo trenutno kreiraju velike količine doslednog koda i kada se otkrije neko poboljšanje ili uoči neki nedostatak, jednostavnom promenom šablona i ponovnim pokretanjem kod generatora, nedostaci ili uvedena poboljšanja koda primenjuju se dosledno na sav kod.
- Doslednost: Kod koji je izgrađen (generisan) upotrebom nekog kod generatora je dosledan u dizajnu nekog API-a, što neće dovesti do iznenađenja i poteškoća za razumijevanje interfejsa.
- Jedinstvena tačka znanja: Podrazumeva se da upotrebom kod generatora možete, na primer, samo na jednom mestu promeniti naziv određene tabele, nakon čega je moguće dobiti brze promjene na ostalim nivoima.
- Više vremena za dizajn: Sa kod generatorima, inženjeri mogu jednostavnim prepravljnjem šablona modifikovati način ponašanja API-a, zatim ponovo pokrenuti generator i realizovati unešene promene. Na ovaj način, kod generatori smanjuju vreme potrebno za primenu velikih prepravki i ostavljaju više vremena za sam dizajn ili testiranje nekih prototipa.

Međutim, kod generatori mogu predstavljati i mač sa dve oštrice. Potrebno je obratiti pažnju na slučajeve kada mogu proizvesti više problema nego koristi, kao na primer: kada obim posla nije dovoljno veliki da bi pravljenje generatora bilo isplativo u smislu vremena i resursa, kada deo aplikacije koju trebamo generisati nema dobro poznatu strukturu, kada skup osobina ili dizajn aplikacije nisu dovoljno precizno definisani unapred ili se često menjaju, za delove koji nisu šablonski, tj. koji su jedinstveni.

3.1. Generisanje koda

Kao što možemo videti na slici 2. (a.), klasični dijagram toka za razvoj i otklanjanje grešaka koda sastoji se od: *izmene, kompajliranja i testiranja*.



Slika 2. – (a.) Klasično dijagram toka izmjene, kompajliranja i testiranja, (b.) Dijagram toka sa uključenim generisanjem koda

Generisanje koda unosi nekoliko novih elemenata u klasični dijagram toka, kao što će se videti na slici 2. (b.). Faze izmjene, kompajliranja i testiranja, su i dalje prisutne, dok novinu čine faze: izmjene šablona i definicionih fajlova i generisanje. Kao prvo, vrši se izmjena šablona ili definicionih fajlova, nakon čega se pokreće generator koji generiše izlazne fajlove, nakon čega se ti fajlovi kompajliraju zajedno sa prilagođenim kodom, te se na kraju vrši njihovo testiranje.

3.2. Koraci u razvoju generatora koda

Prilikom razvoja kod generatora, treba se pridržavati sledećeg niza koraka:

- razvoj testnog izlaznog koda: najpre treba odrediti kako bi izlaz iz generatora trebao izgledati, njegovo format. Ovo će vam dati jasan cilj i jasnu sliku onoga šta treba da ekstraktujete iz ulaznih fajlova.
- dizajn generatora: kako bi trebalo da generator prihvata ulaz? Kako će parsirati izlaz? Kako razviti izlaz – koristiti šablone ili jednostavno pisati u fajl? Ovo su neka od pitanja o kojima bi trebali razgovarati, pre nego što pređete na kodiranje.
- razvoj parsera ulaza: implementirati izabrane načine parsiranja ulaza u kod generator.

4. RAZVOJ SOPSTVENOG KOD GENERATORA I UPOTREBA GENERISANIH IZLAZA

Prije nego što krenemo u razvoj sopstvenog kod generatora i upustimo se u tehnike automatskog generisanja koda, potrebno je izdvojiti jedan deo vremena (treba imati na umu da razvoj ovakvog rješenja može trajati kao i razvoj konkretne aplikacije, ali da njegova ponovna upotreba može da opravda uloženi trud i vrijeme) za analiziranje dosadašnjeg načina rada i uočavanje pogodnih dijelova za automatizaciju.

U našem slučaju govorimo o poslovnim desktop aplikacijama, koje kao skladište za smještanje podataka koriste bazu podataka. Ako posmatramo više ovakvih aplikacija, jasno ćemo uočiti da sve one donekle zahtijevaju iste funkcionalnosti, prije svega u pogledu:

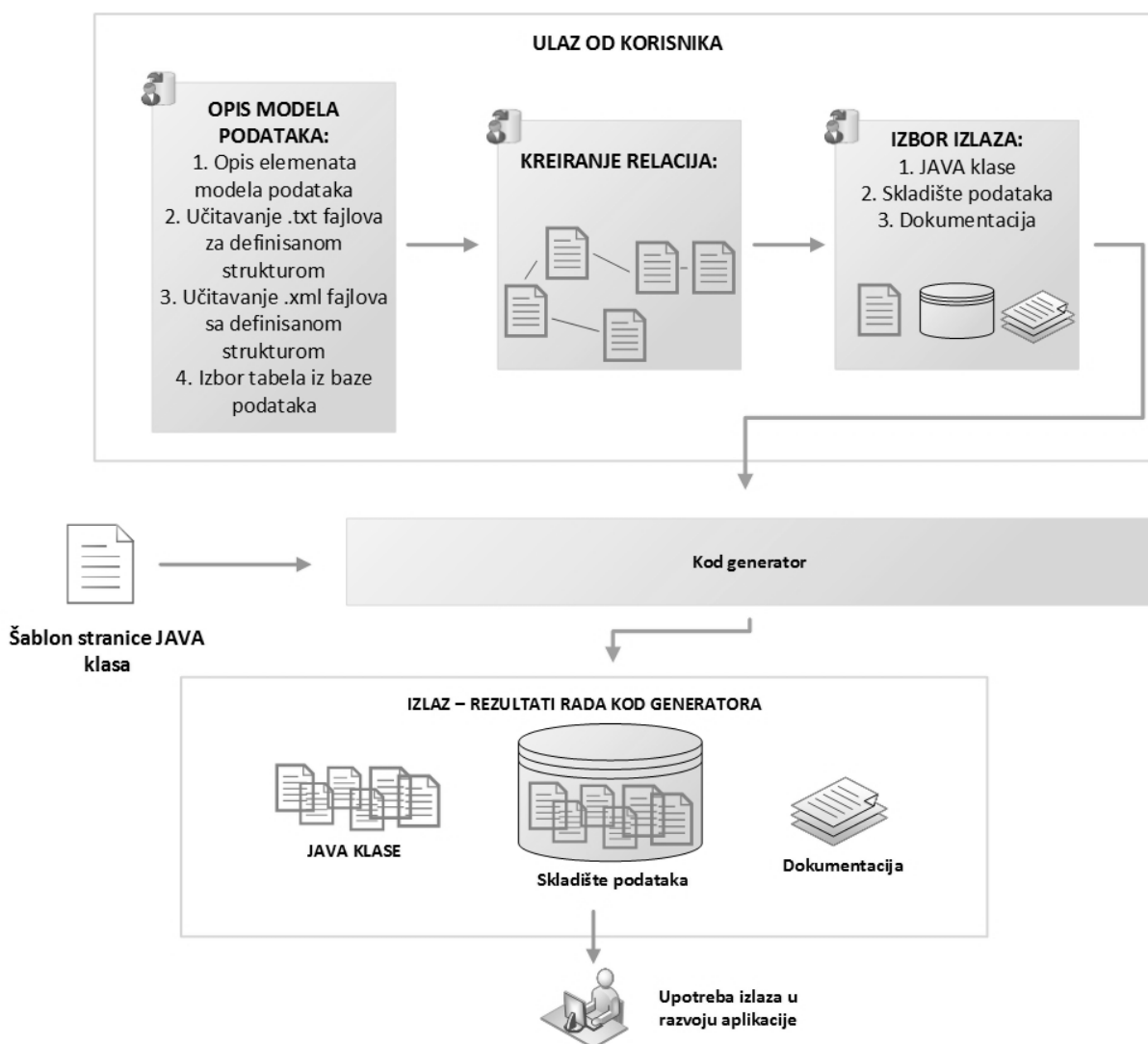
- pristupa podacima: poslovne aplikacije predstavljaju informacioni sistem određene firme, koji sadrži poslovne podatke i operacije za manipulaciju tim podacima. Poslovni podaci su

smješteni u nekoj od baza podataka i gotovo uvijek postoji potreba za pregledom, unosom, brisanjem i izmjenom tih podataka (engl. CRUD- Create Read Update Delete). Ove operacije predstavljaju osnov za druge, složenije operacije;

- prikaza podataka: podatke smještene u bazi podataka je potrebno prikazati korisniku na odgovarajući način, koristeći neki od raspoloživih grafičkih komponenti (tekstualnih polja, tabela, lista, itd.).

Dakle, posao oko pisanja koda za pristup i prikaz podataka može biti jako težak i zamoran, koji će vrlo brzo dovesti do monotonije u radu i znatno povećati šanse za neuspjeh kompletnog projekta. Ulaganje truda i vremena u razvoj kod generatora, koji će ove poslove obavljati umjesto nas, smatra se potpuno opravdanim.

Posao pisanja velike i često vrlo kompleksne i ponavljajuće količine koda, sada zamjenjujemo znatno lakšim i interesantnijim poslom unosa ulaza i upotrebe dobijenih izlaza. Konceptualni model rada našeg kod generatora dat je na slici ispod.



Slika 3. – Konceptualni model rada kod generatora

4.1. Ulaz

Umjesto da piše veliku količinu koda za kreiranje klasa, na gore predstavljeni način, korisnik ima mogućnost da ih "opisuje", tj. definiše njihova imena, attribute, na neki od sledećih načina:

- kreiranjem novog tekstualnog fajla: korisnik kreira novi tekstualni fajl čiji sadržaj u odgovarajućoj unaprijed definisanoj strukturi opisuje jednu klasu u sistemu;

- učitavanjem već postojećeg tekstualnog fajla i njegova modifikacija: ukoliko se javi potreba upotrebe klasa koje su korišćene u nekim od prethodnih projekata, moguće ih je učitati i po potrebi mijenjati;
- učitavanjem podataka iz XML fajlova: budući da XML ima mogućnost kreiranja fajlova sa tačno definisanom strukturom, predstavlja još jedan od mogućih i pogodnih načina na koji se mogu opisati potrebne klase;
- učitavanje tabela iz baze podataka: korisnik može odabirom baze podataka i tabele iz baze da kreira klasu koja po strukturi odgovara izabranoj tabeli;

Sljedeći korak, u obezbeđivanju ulaza u kod generator, bio bi određivanje relacija između tabela.

4.2. Izlaz

Parsiranjem tekstualnih fajlova koji sadrže opise klasa i njihovih međusobnih relacija, uz upotrebu dobro definisanih šablona (dobro prethodno provjerenih i testiranih dijelova koda) generišu se izabrani izlazi. Šabloni su obično tekstualni fajlovi koji sadrže fiksne dijelove, sa dinamičkim (jedinostvenim) poljima koja se popunjavaju podacima koji zavise od modela modataka.

4.3. Upotreba izlaza

Dobijanjem izlaza iz kod generatora, posao oko razvoja aplikacije nije završen, ali u velikoj mjeri je olakšan, jer je kreiran jedan okvir (okruženje) koji će pomoći programeru i olakšati njegov posao, na način da je veći deo koda zaduženog za rad sa podacima i njihovo prikazivanje unutar grafičkih komponenti, kreirao umjesto njih.

5. ZAKLJUČAK

Zahtjevi koje postavlja današnje tržište softvera uslovljavaju neprestalno usavršavanje rada IT organizacija, kako bi poslove rešavale u što kraćem vremenskom roku, uz minimalne troškove rada. Pred ovako postavljenim uslovima, IT organizacije zarad svog opstanka moraju naći načina kako ove faktore smanjiti i upravo način koji je ovdje predložen, može biti moguće rješenje. Svaka sledeća razvijena aplikacija, kako pojedincu, tako i cijelom razvojnom timu donosi novo iskustvo i povećava kvalitet izrade softvera. Takav put, vođen iskustvom, predstavlja idealan način za izdvajanje i definisanje okvira, odnosno dijelova aplikacija koje mogu biti kreirane automatski, uz minimalan uloženi napor i u samo nekoliko trenutaka.

LITERATURA

- [1.] "Chetan Panchal: What is framework?", (pristup: nn), [dostupno na <http://blog.newtonicaonline.com/2008/12/06/what-is-framework/>]
- [2.] "Enciklopedija:", (pristup: nn), [dostupno na http://en.wikipedia.org/wiki/Object-oriented_design]
- [3.] "Marcus Eduardo Markiewicz, Carlos J.P. Lucena: Object Oriented Framework Development", (pristup: nn), [dostupno na <http://www.acm.org/crossroads/xrds7-4/frameworks.html>]
- [4.] "Xavier Amatriain", (pristup: nn), [dostupno na Framework, <http://xavier.amatriain.net/Thesis/html/node23.html>]
- [5.] Taligent: Building Object-Oriented Frameworks
- [6.] Johnson, R. E: How to Design Frameworks
- [7.] S. Lawrence, J. M. Atlee, "Softversko inženjerstvo – teorija i praksa", prevod trećeg izdanja
- [8.] J. Herrington, Code Generation in action, USA, Manning, 2003

Projektovanje razvojnog okruženja za mikrokontrolere u nastavi

Designing microcontroller development environment for teaching

Dragoljub Pilipović, Željko Gavrić, Fakultet za informacione tehnologije, Slobomir P Univerzitet,
Slobomir - Bijeljina, Republika Srpska

Apstrakt - Brz razvoj na području elektronike donosi mnogo mogućnosti ali i obaveza. Potreba za stalnim usavršavanjem je danas imperativ. Uočili smo da na visokoškolskim obrazovnim ustanovama ne postoji jedinstven stav šta treba učiti za odličan oblik primjenjene elektronike - za mikrokontrolere. Zato smo izabrali konkretan mikrokontroler i za njega projektovali razvojno okruženje koje bi se moglo koristiti u nastavi i to prvenstveno na laboratorijskim vježbama. To uključuje i potreban softver za pisanje asemblerkih i C programa.

Ključne reči – mikrokontroleri, razvojni sistem, MPLAB, PIC, nastava.

Abstract - The rapid development in electronics brings a lot of opportunities and obligations. The need for continuous improvement is modern imperative. We observed that there is not a unique standpoint at the high educational institutions to what learn for a great form of applied electronics - for microcotrollers. That's why we chose a particular microcotroller and then projected development environment for it that could be used in teaching, primarily on laboratory exercises. This includes the necessary software for writing assembler and C programs.

Index terms – microcontrollers, development environment, MPLAB, PIC, teaching.

1. UVOD

Ovaj rad bavi se područjem primjenjene elektronike i to njenim istaknutim dijelom – mikroprocesorskim sistemima (mikrokontrolerima u užem smislu). Mikrokontroleri su digitalna poluprovodnička komponenta koja je naizgled slična, ali se dosta razlikuje od svima poznatih računarskih procesora. Tom procesoru trebaju dodatne komponente da bi proradio, poput memorije ili U/I periferija. On čini srce mikroračunarskog sistema ali sam za sebe nije operativan. Mikrokontroler u sebi sadrži i mikroprocesor i sve dodatne komponente neophodne za rad. Na taj način se štedi i vrijeme za konstruisanje nekog hardverskog uređaja. Moglo bi se uslovno reći da je mikrokontroler skoro pa isto kao mikroračunar, naravno gledajući po funkcionalnosti.

Iako mikrokontroler integriše dosta dodatnih komponenti u sebi, to nije dovoljno za nastavne potrebe. Tu prazninu popunjavaju razvojni sistemi za mikrokontrolere. Njihova primjena olakšava i skraćuje vrijeme potrebno za projektovanje operativne i potpuno funkcionalne opreme. Mikrokontrolerski razvojni sistem pored standardnih sklopova mikroračunara (mikroprocesor, RAM, ROM, takt generator) posjeduje i:

Dragoljub Pilipović – Fakultet za informacione tehnologije, Slobomir P Univerzitet, Pavlovića put 70,
76300 Slobomir, Republika Srpska, Bosna i Hercegovina (e-mail: dragoljub.pilipovic@gmail.com).

Željko Gavrić – Fakultet za informacione tehnologije, Slobomir P Univerzitet, Pavlovića put 70, 76300
Slobomir, Republika Srpska, Bosna i Hercegovina (e-mail: zeljko.gavric@spu.ba).

- upravljački sklop koji omogućuje resetovanje sistema,
- upravljački sklop za zaustavljanje sistema,
- upravljački sklop za rad “korak po korak”,
- punjenje memorijskih lokacija,
- sistem za prikazivanje sadržaja registara,
- spoljašnju memoriju,
- programator,
- periferne jedinice i
- programsku razvojnu podršku.
- Programsku razvojnu podršku čine:
 - assembler (Assembler),
 - prevodilac (Compiler),
 - program punilac (Loader),
 - program za traženje grešaka (Debugger),
 - uređivač programskog kôda (Editor) i
 - simulator.[1]

Mi ćemo pojam razvojnog sistema proširiti sa dodatnim komponentama, prilagođenim nastavnim potrebama, i tako će nastati razvojno okruženje. Prvi korak u projektovanju razvojnog okruženje predstavlja biranje konkretne hardverske platforme i njenog centralnog dijela – konkretnog modela mikrokontrolera. Kada to završimo izabraćemo odgovarajuću softversku podršku, te onda ide najobimiji posao u projektovanju razvojnog okruženja: projektovanje razvojne štampane pločice na kojoj su predviđene dodatne potrebne elektronske, strujne, mehaničke i druge komponente. Da bi se sistem stavio u operativno stanje potrebno je napisani program u izabranim programskim jezicima prevesti u mašinski oblik, napuniti programsku memoriju mikrokontrolera i onda sve testirati da se provjeri postojanje grešaka pri izvršavanju. Na kraju će se predložiti šta bi moglo da se radi na časovima sa ovim razvojnim okruženjem.

2. IZBOR MIKROKONTROLERA

Današnja opšta računarska arhitektura pripada Harvard tipu (za razliku od ranije i jednostavnije Von-Neumanove arhitekture), u kojem se u odvojenim dijelovima računara nalaze memorija sa podacima i memorija sa programskim instrukcijama.[1] Na osnovu te polazne činjenice brzo pretragom po Internetu na sajtovima univerziteta i prodavnica elektronike smo najčešće pronašli ove mikrokontrolere:

- AmTel AVR (od Alf Vegar Risc) serija mikrokontrolera (instrukcije su RISC tipa, 8-bita),
- Intel 8051 (instrukcije su CISC tipa, 8-bita),
- MicroChip PIC 16Fxxx (instrukcije su RISC tipa, 8-bita),
- TI (od Texas Instruments) MSP430 (instrukcije su RISC tipa, 16-bita) i
- ARM (od Acorn/Advanced Risc Machine) čije instrukcije su RISC tipa, 32-bita.[2]

Poslije korišćenja ovih kriterijuma masovnosti i vidljivosti, napravljeno je ukrštanje osnovnih karakteristika i došlo se do ishoda da nam je potreban RISC mikrokontroler sa 8-bita. To su u našoj listi AVR i PIC serija mikrokontrolera. Takođe po kriterijumu upotrebljivosti dobili smo isti ishod. Ipak gledajući po kriterijumu cijene i lokalne dostupnosti, PIC je bolje prošao. Ovome doprinosi i broj integrisanih perifera, vrsta pakovanja i priključivanja mikrokontrolera na štampane pločice. Najzad

prevagnulo je subjektivno mišljenje da će studenti lakše savladati programiranje u asemblerskom jeziku za PIC jer on ima bolje imenovane instrukcije i stilski je ljepši.

Između brojnih modela 16Fxxx serija PIC mikrokontrolera izabrali smo PIC 16F1827. Ovaj srednje pozicionirani mikrokontroler je proizvod firme MicroChip i pripada klasi osmobarbitnih mikrokontrolera (8-bitna magistrala za podatke; jedan 8-bitni radni akumulator) sa RISC arhitekturom. Harvard arhitektura je novija od Von-Neumanove i njena glavna odlika je što su magistrala za podatke i magistrala za programske adrese odvojene jedna od druge. Ovo omogućava veliki protok podataka ka procesoru što implicira i veću brzinu izvršavanja programa. Uobičajeno je da Harvard arhitektura ima manji broj instrukcija i one se uglavnom izvršavaju u jednom ciklusu.[3]

Naš PIC 16F1827 ima 49 instrukcija i sve se one izvršavaju u jednom ciklusu osim onih za skok, grananje i indirektno adresiranje. Odvajanjem programa od memorije za podatke omogućava se da instrukcije ne budu striktno osmobarbitne, nego su one kod našeg PIC-a dužine 14 bita. Većina instrukcija su naslijeđene iz ranijeg perioda, a neke druge su optimizovane za rad sa C programskim naredbama. Model posjeduje programsku fleš memoriju za 4096 programskih riječi, radnu memoriju od 384 bajta, kao i EEPROM za podatke veličine 256 bajtova.

Interni oscilator ima podesiv raspon frekvencija od 31 kHz do 32 MHz (u tom slučaju je za jednu instrukciju potrebno 125 ns za izvršavanje), dok se radni napon kreće u granicama od 1,8 V do 5,5 V.

Adresiranje može biti direktno, indirektno i relativno putem dva 16-bitna registra, preko kojih se može doći do sadržaja podatkovnih i programskih memorijskih lokacija. Pri radu sa podrutinama i prekidima koristi se hardverski stek dubine 16 nivoa sa automatskim spremanjem i kasnijim vraćanjem radnog konteksta.

Integrirane periferije su brojne, te izdvajamo neke od njih:

- Watch-dog tajmer (WDT),
- Tajmer 0 je 8-bitni i ima podesiv prescaler,
- Tajmer 1 je 16-bitni brojač/tajmer sa preskalarom i sopstvenim 32 kHz oscilatorom,
- Tajmeri 2, 4 i 6 su 8-bitni sa preskalerom, postskalerom i 8-bitnim period registrom,
- 15 ulazno-izlaznih linija i jedna linija samo za ulaz,
- Tri modula za Capture, Compare i PWM (Pulse-width modulation) funkcije,
- Komunikacija može da ide preko MSSP (Master Synchronous Serial Port) sklopa,
- Druga serijska komunikacija ide preko eUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter) koji se koristi za RS-232 i RS-485 standarde,
- Modul za konverziju analognih veličina u digitalne vrijednosti 10-bitne rezolucije sa 12 kanala, itd.
- Vidimo da PIC 16F1827 ima dosta mogućnosti koje će se moći izučavati na časovima, ali i za praktičnu primjenu u privredi.

3. IZBOR SOFTVERA ZA RAZVOJ PROGRAMA

Efikasno korišćenje mikrokontrolera zahtjeva odlično poznavanje njihove interne arhitekture, prvenstveno iz razloga ograničenih resursa na njima. Potpuno iskorišćenje sa programerske strane se može postići kontrolom na nivou jedne instrukcije što pruža način pisanja programa kao niza asemblerskih instrukcija. Takav program može da se piše u običnom uređivaču kôda, npr. opšte poznati Notepad. Ipak mnogo je udobnije i posljedično efikasnije koristiti neki IDE (Integrated Development Environment) softverski paket. Predlažemo za tu svrhu MPLAB paket firme MicroChip, čiji je proizvod i izabrani model mikrokontrolera. Postoje dva osnovna kraka ovog softvera: starija MPLAB 8.xx i novija MPLAB X. Prvo se izvršava samo na 32-bitnoj Windows platformi i bilo je

kompletno originalni proizvod MicroChip-a. S druge strane, MPLAB X je zasnovan na otvorenom i besplatnom NetBeans-u firme Oracle.

Koliko god je za najefikasnije iskorišćenje programskih ciklusa mikrokontrolera bitan rad sa assemblerom (što dovodi do negativne posljedice – velikih napora i vremena uložnog u pisanju takvih programa), toliko je za optimalan rad potreban neki apstraktniji jezik. Našli smo SDCC (Small Device C Compiler), PICBASIC firme MEL, a firma Mikroelektronika iz Beograda nudi mikroC, mikroBasic i mikroPascal. Ipak svi se poprilično plaćaju (osim prvog, ali on ionako ima samo poluzvaničnu podršku za naš PIC model), dok su Mikroelektronikini dodatno vezani za njihov hardver (koji ćemo još spomenuti kasnije u radu). I ovdje je izbor u stvari jednostavan: Microchip ima dva sopstvena C prevodioca. To su stariji HighTec C i noviji XC8 prevodilac. Oba postoje u besplatnoj verziji koja se od plaćenih razlikuju po stepenu optimizacije veličine dobijenog mašinskog kôda iz C izvornika, tj. besplatne verzije proizvode veće zauzeće u programskoj memoriji mikrokontrolera. Ipak, za nastavne potrebe to nije negativna strana.

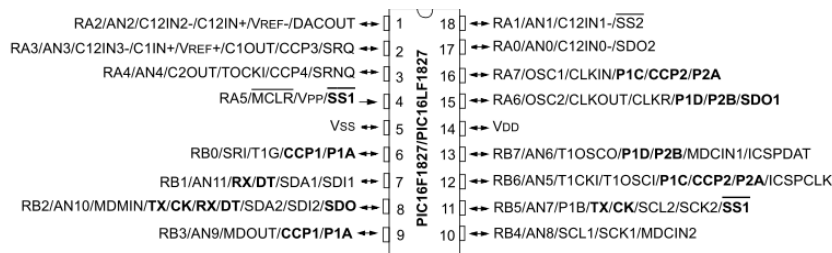
MPLAB X je univerzalni skup alata za razvoj softvera za skoro hiljadu modela mikrokontrolera matičnog proizvođača svih nivoa (8-, 16- i 32-bitnih PIC mikrokontrolera i dsPIC digitalnih signalnih procesora). Sastoji se od uređivača kôda, prevodilaca, emulatora, simulatora i dibagera. Prepoznaje ukucane naredbe i dopunjava ih, potom ih boji po vrstama namjene. Sam uvlači dijelove petlji i funkcija; ima alat za refaktorisanje. Postoje i prozori sa nadgledanjem vrijednosti promjenjivih i unutrašnjih memorijskih lokacija pri izvršavanju programa korak po korak. Moguće je upravljati sa više projekata odjednom. Pošto je baziran na NetBeans-u podržava sve značajne operativne sisteme (Windows, Linux, Macintosh), kao i sistem dodataka (plug-ins). [7]

4. PROJEKTOVANJE RAZVOJNE PLOČICE

Osnova razvojnog sistema je matična (razvojna) pločica, koja je u stvari štampana pločica i predstavlja pasivnu elektronsku komponentu na koji će se spojiti i preko nje ostvariti vezu sve elektronske komponente poput otpornika, kondenzatora itd. uključujući i sam mikrokontroler.[5] Razvojni sistem za mikrokontrolere je onaj hardverski uređaj koji se koristi u fazi pisanja i testiranja programa a na sebi ima više komponenti koje mu služe da mikrokontroler bude u okruženju sličnom onome u kome će se u toku produkcione faze uposliti. Univerzalni razvojni sistemi imaju podršku za mikrokontrolere raznih proizvođača i arhitektura, dok ciljani imaju podršku za tačno jedan model ili seriju veoma sličnih modela.

Pri projektovanju razvojne pločice imali smo kao uzor Mikroelektronike Ready-for-PIC, EasyPIC i BigPIC serije razvojnih sistema koji se u svijetu smatraju najboljim u odnosu šta se dobija za uloženo. Ipak naši zahtjevi su dosta niži vezano za broj i raznovrsnost komponenti koje su potrebne za nastavne uslove.

Razvojna pločica će biti relativno mala, te će imati dimenzije 10 puta 10 centimetara. Centralni dio zauzimaće PIC16F1827 mikrokontroler koji se nalazi u 18-pinskom PDIP (Plastic Dual In-line pin Package) kućištu, što je način pakovanja integrisanih kola u obliku pravougaonog kućišta sa dva reda metalnih nožica. Kućište ima radni raspon temperatura između -40° C i +85° C. Slika 1 prikazuje spoljni izgled mikrokontrolera sa rasporedom i značenjem nožica.

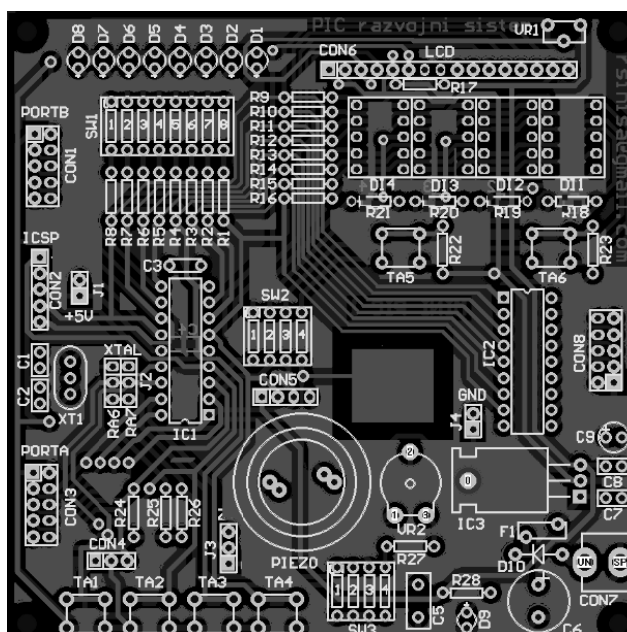


Slika 1: Raspored i značenje nožica na PIC 16F1827.

Razvojna pločica bi trebala biti izrađena od RF-4 materijala, koji je kompozitnog tipa sastavljen od fiberglasa i epoksi smole sa karakteristikama izolatora, nezapaljivosti i nehidroskopnosti (ne upija

vlagu). Bakarni provodni vodovi će biti presvučeni HASL (hot air solder levelling) završnicom radi sprečavanja presjecanja i rđanja vodova. Pošto će štampa biti dvostrana, rupe će biti metalizirane radi lakšeg lemljenja komponenti. U ćoškovima će se izbušiti veće nemetalizirane rupe radi postavljanja pločice na plastične odstoynike da bi bila odvojene od podloge ispod nje.

Razvojna pločica treba biti projektovana u programu za dizajn pločica (EDA – Electronic Design Automation), poput KiCAD-a. On je besplatan, postoji za sve opšte operativne sisteme i ima sve korake u dizajnu pločica, poput definisanja šeme, kreiranja izgleda štampe, prikaz u 3D prozoru, jednostavan autoruter i sl. Krajnji izlaz se može dobiti u tekstualnom Gerber RS-274X formatu, što predstavlja industrijski standard za razmjenu slika štampanih pločica.[6] Štampa pločica može da se uradi u kućnim/hobi uslovima, ali preporučujemo da se uradi kod privrednih organizacija koje nude profesionalne usluge poput električnog testiranja odštampanih pločica ili Internet firmi koje nude slične usluge poput kineskog Seed Studio servisa [13]. Primjer projektovane štampe pločice se nalazi na slici 2 preuzete sa [12].



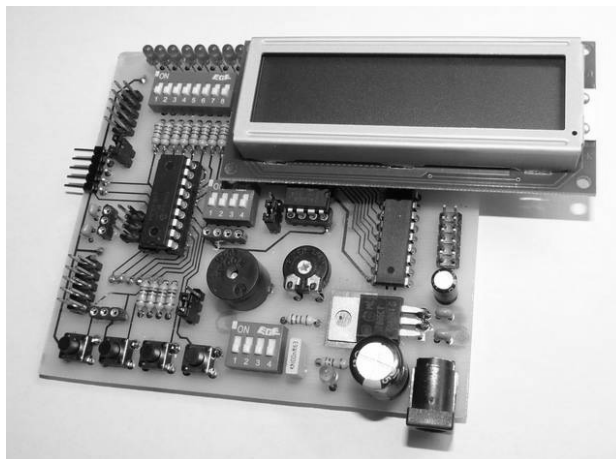
Slika 2: Šema razvojne pločice za PIC 16Fxxx mikrokontrolere.

1. Otpornici. Oni imaju sljedeće vrijednosti: R1 - R8, R28 = 330R; R9 - R16 = 150R; R17 = 10R; R18 - R23 = 1K; R24 - R26 = 4K7; R27 = 2K2. Svi otpornici su od 0,25W. Ipak, predložene vrijednosti za otpornike od R1 do R8 i R28 zavise od upotrebljenih LED dioda. Proračunati su za struju od oko 10mA što odgovara skoro svakoj LED diodi (sem ako nije malostrujna od 2 mA), ali možete da pojačate te otpornike ako se priključuje LED diode koje imaju jači sjaj. Otpornik R17 zavisi od toga kako je povezano pozadinsko osvjetljenje konkretnog LCD displeja. Ako ga nema, izostavlja se. Ako redni otpornik za pozadinsko osvjetljenje već postoji na displeju, ovdje se stavlja običan provodnik (žica); a ako ne postoji, pogledati u tehničkim podacima za taj displej kako je povezano pozadinsko osvjetljenje i prema tome ga proračunati. Vrijednost od 10 Ω odgovara za displej tip A, a za drugi tip B, treba staviti 100 Ω .

2. Kondenzatori. Kondenzatri C1 i C2 trebaju da imaju vrijednost od 22 pF; C3 i C5 vrijednosti 100 nF i međusobni razmak od 200 milicimetara; C7 i C8 vrijednosti 100 nF i razmak od 100 milicimetara; C4 vrijednost 100 nF, SMD tip; C6 da ima vrijednosti 470 μ F i 25 V, stojeći tip; i na kraju, C9 vrijednosti 10 μ F i 16 V, stojeći tip. Kondenzator C4 je paralelan sa C3 i treba postaviti samo jedan od njih. Ako je postavljen SMD ili želite da lakše vadite PIC mikrokontroler iz podnožja postavite C4, u protivnom postavite C3. Kondenzator C6 nije neophodan ako postoji ispravljač koji dostavlja dobro filtriran napon ili ako je stabilisan napon u pitanju.

3. Svjetleće diode. LED diode od D1 do D9 mogu da budu bilo koje veličine od 3 mm; dok dioda D10 može da bude bilo koja za struju od 1A.

- 4. Trimeri.** Trimeri sa potenciometrom VR1 od 10K stojeći tip i VR2 takođe od 10K ali ležeći tip.
 - 5. Tranzistori.** Svi od T1 do T4 su SMD (surface-mount technology) tipa.
 - 6. LED displeji.** Sedmosegmentni LED displeji DI1 do DI4 mogu da budu bilo koji sa zajedničkom anodom i sa visinom 9,9 mm.
 - 7. Oscilirajući kristal.** Pošto interni oscilator u PIC mikrokontroleru ima već u sebi “okrugle” brojeve za učestanost, za XTAL je predviđen kristal od 3,2768 MHz jer je on jako zgodan za binarno djeljenje tako da se dobije cjelobrojna učestanost od npr. 100 ili 200, što je pogodno za tajmere i za LED multipleksiranje.
 - 8. Osigurač.** Resetabilni osigurač F1 je predviđen da zaštiti razvojni sistem.
 - 9. Integrisana kola.** U IC1 se postavlja PIC 16F1827-I/P, a u IC2 ide ULN2803.
 - 10. Tasteri.** Za male tastere od TA1 do TA6 bi bilo najbolje da im se visina poklapa sa visinom LED displeja.
 - 11. Prekidači.** DIP prekidači su SW1 sa 8 pozicija i SW2 sa 4 pozicije.
 - 12. Konektori.** Konektori CON1, CON3 i CON8, kao i kratkospojnik J2 mogu da se naprave od dvoredne letvice poput Cvilux HN2X20. Konektor CON2 služi za priključenje programatora i donje iglice su mu pod uglom. CON4 i CON5 može da se napravi takođe od navedene letvice. Obratite pažnju da je potrebno dva komada od po tri iglice - jedan komad se postavlja na mjesto kristala kako bi mogli da ga mjenjamo. CON6 je tipa “board to board” konektor. CON7 ima u stvari 3 izvoda, ali treći, onaj sa strane, nije predviđen da ide na štampanu pločicu te je višak (treba ga ukloniti). Inače služi za priključenje izvora napajanja.
 - 13. Kratkospojnici.** Džamperi J1 i J4 su sa dvije prave iglice, J2 je ima 2 x 3 prave iglice, J3 ima 1 x 3 pravih iglica. Potrebno je nabaviti i odgovarajući broj kapica za spajanje kratkospojnika.
 - 14. Postolja.** Postolja za integrisana kola su DIL18 tipa i u njih idu PIC i ULN2803.
 - 15. Zvučnik.** U PIEZO se može staviti bilo koji zvučnik koji radi na piezoelektričnom konceptu.
 - 16. LCD displeji.** LCD displej bi trebalo da bude sa 16 mjesta za slova u dva reda i ne mnogo većih dimenzija od 80 mm x 36 mm x 14 mm. Primjer za tip A je žuto-zeleni Ampire AC-162BYILY-H, dok je primjer za tip B plavi Winstar WH1602B-TMI-CT. Oba su STN (super-twisted nematic) tipa, imaju pozadinsko osvjetljenje i isti prošireni skup znakova za prikaz. U principu može se staviti bilo koji model LCD displeja koji funkcioniše sa drajverom HD44780. Potrebna je i kontaktna letvica od 16 pinova koji se leme na sam LCD displej. Skup internih karaktera između ostalog uključuje latinsku abecedu i rusku ćirilicu. Za pisanje na našem jeziku predlažemo nekoliko mogućnosti. Kod latiničnog pisma nedostaje 6 slova (š, č, ž, đ, ć i dž), te prvo predlažemo zamjenu tih slova sa ekvivalentima bez dijakritičkih znakova (česta pojava “ošišane latinice”): š => s, č => c, ž => z, đ => dj, ć => c i dž => dz. Drugi predlog je zamjena nedostajućih slova sa dva uzastopna slova nekarakteristična za naš jezik: š => sh, č => ch, ž => zh, đ => dh, ć => cc i dž => dz. Pošto oba modela LCD displeja imaju rusku ćirilicu predlažemo zamjenu nedostajućih 5 slova (љ, њ, џ, ђ i љ) sa pristupom kao u drugog predloga: љ => лъ, њ => нь, џ => дж, ђ => чъ i љ => дъ. Manja modifikacija ovog predloga glasi da slovo ђ bude zamjenjeno sa њ, a slovo љ sa я.
- Štampana pločica je projektovana tako da može da se izradi bez metaliziranih rupa ali u tom slučaju treba obratiti pažnju na redosljed pri lemljenju. Prvi korak je da se kroz rupe provuče žica i zalemi sa obe strane pločice. Naravno, ovo nije potrebo raditi ako su napravljene pločice sa metaliziranim rupama kao što ih izrađuje pomenuti Internet servis. Izgled kompletno završenog razvojnog sistema se vidi na slici 3.[11]



Slika 3: Kompletiran razvojni sistem.

5. PROGRAMATOR I DIBAGER

Mikrokontroler treba poslije pisanja programa i prevođenja u HEX oblik preko uređaja za programiranje (tzv. programatora) napuniti tim mašinskim naredbama iz HEX zapisa.

Sve DIP prekidače postaviti u isključen položaj, džemper J1 je postavljen, J2 postavljen na RA6/RA7, J3 na AD, J4 nevažno ako je privremeno izvađen ULN2803 i onda postaviti LCD displej u konektor CON6. U konektor za napajanje CON7 uključiti konektor iz ispravljača (plus pol je u sredini) napona od 7 V do 25 V (najbolje što približnije 12 V, sve preko toga izazvaće povećano zagrijavanje stabilizatora 7812 koji je stavljen u IC3). PIC 16F1827 je, naravno, već postavljen u svoje podnožje.

Priključiti programator (pretpostavimo da je to PICKit 2, malo dalje ćemo o karaktersikama programatora) u USB port PC računara, pa priključiti ICSP (In-Circuit Serial Programming) konektor od programatora na ICSP konektor razvojnog sistema (on je obilježen sa CON2).

Sada pokrenite program "PICKit 2 Programmer Application" skinut sa sajta proizvođača MicroChip-a. Softver bi trebalo da prepozna model PIC mikrokontrolera i ispiše njegovu oznaku u redu ispod menija ("Device: PIC16F1827"). Zatim u meniju odabrati Tools \ Target VDD Source \ Auto-Detect.

Sada sa menija odabrati File \ Import HEX, onda izabrati testnu binarnu datoteku "test_all.HEX", pa kliknuti na taster "Write" i sačekati nekoliko sekundi da se završi programiranje PIC-a. Ako je sve u redu, statusni prozor će biti ispunjen zelenom bojom sa tekstom: "Programming Successful". Nakon toga, na LCD displeju treba da se pojavi pozdravni tekst. Ako se ne vidi dobro tekst, probati podesiti kontrast okretanjem trimera potencijometra VR1.

Mikrokontroler je tako isprogramiran i sve je spremno za izvršenje upisanog programa, ali prije toga treba odspojiti programator od razvojnog sistema.

Sa predstavljenim razvojnim sistemom može se koristiti bilo koji programator, ali mi preporučujemo PICKit 2 i PICKit 3. Obadva su proizvodi firme Microchip, međutim postoje i kompatibilni programatori drugih proizvođača i pojedinaca koji imaju potpuno istu funkcionalnost kao originalni a povoljniji su za kupovinu u svrhe obrazovanja.

Ipak između dva spomenuta modela postoje određene razlike. PICKit 2 je jednostavniji u svojoj konstrukciji što doprinosi robusnosti i dugovječnosti, ali nema mogućnost da bude dibager za naš model mikrokontrolera bez jednog hardverskog dodatka tzv. debug header-a. PICKit 3 je moćniji i ima podršku za sve modele PIC i dsPIC mikrokontrolera jer za svaki PIC u njega se učitava drugi program (taj programator interno ima takođe mikrokontroler). Bitna razlika je što ovaj drugi podržava ICD (In-Circuit Debugging). To je mogućnost praćenje izvršavanje programa direktno na isprogramiranom mikrokontroleru, zaustavljanje i nastavak izvršavanja, mijenjanje vrijednosti memorijskih registara, te na taj način lako uočavanje grešaka u programu. Sve je ovo moguće uraditi iz MPLAB razvojnog softvera. Za oba modela programatora postoji funkcija programiranja bez povezivanja sa PC računarom, jedino je potrebno obezbijediti napajanje sa samostalnog USB priključka.

6. ZAKLJUČAK

Kroz osmišljavanje i projektovanje razvojnog sistema za mikrokontrolere prošli smo razne korake u više tehnika i disciplina da napravimo jednu zaokruženu cjelinu. Tako smo dobili odličnu kombinaciju softvera i hardvera za malo uloženi sredstva. Upotreba ovog razvojnog sistema može da bude u hobi ili lične svrhe, te za manju proizvodnu svrhu, ali glavna svrha je upotreba u obrazovanju. Omogućava proučavanje arhitekture mikrokontrolera, pisanje asemblerskih i C programa, kao i isprobavanje napisanih programa u što realnijim uslovima. Na časovima je moguće isprobati dosta integrisanih periferija i tako demonstrirati studentima mogućnosti mikrokontrolera.

Kao nastavak, predlažemo projektovanje sopstvenog kompatibilnog PICkit 2 i/ili PICkit 3 programatora. Takođe se može mnogo preciznije opisati šta se može isprobati sa pomenutim razvojnim okruženjem, npr. nešto poput tekstualnog tutorijala ili čak video tutorijala.

LITERATURA

- [1.] D. Pilipović, Skripta za Mikroprocesorske sisteme – laboratorijske vježbe, Bijeljina, 2007.
- [2.] Vikipedija – slobodna enciklopedija: “Microcontroller”, (pristup: 15.10.2013.), [dostupno na <http://en.wikipedia.org/wiki/Microcontroller>].
- [3.] Vikipedija – slobodna enciklopedija: “List of common microcontrollers”, (pristup: 15.10.2013.), [dostupno na http://en.wikipedia.org/wiki/List_of_common_microcontroller].
- [4.] Vikipedija – slobodna enciklopedija: “PIC microcontroller”, (pristup: 15.10.2013.), [dostupno na http://en.wikipedia.org/wiki/PIC_microcontroller].
- [5.] Vikipedija – slobodna enciklopedija: “Printed circuit board”, (pristup: 15.10.2013.), [dostupno na http://en.wikipedia.org/wiki/Printed_circuit_board].
- [6.] Vikipedija – slobodna enciklopedija: “Gerber format”, (pristup: 15.10.2013.), [dostupno na http://en.wikipedia.org/wiki/Gerber_format].
- [7.] Vikipedija – slobodna enciklopedija: “MPLAB”, (pristup: 15.10.2013.), [dostupno na <http://en.wikipedia.org/wiki/MPLAB>].
- [8.] B. Dimirijević, S. Aleksić, I. Antić, Razvojni sistem za mikrokontroler PIC 16F877A, Elektronski fakultet, Niš, [dostupno na <http://es.elfak.ni.ac.rs>]
- [9.] M. Verle, PIC Microcontrollers, Mikroelektronika, Beograd, 2008.
- [10.] Microchip: “Data sheet for PIC16(L)F1826/27”, (pristup: 15.10.2013.), [dostupno na <http://ww1.microchip.com/downloads/en/devicedoc/41391D.pdf>].
- [11.] Forum Elite Security: “Mikrokontroleri od A do Š”, (pristup: 15.10.2013.), [dostupno na <http://www.elitesecurity.org/t72345>].
- [12.] Forum Elite Security: “PIC razvojni sistem”, (pristup: 15.10.2013.), [dostupno na <http://www.elitesecurity.org/t454860>].
- [13.] Seed Studio: “Fusion PCB Service”, (pristup: 15.10.2013.), [dostupno na <http://www.seeedstudio.com/depot/fusion-pcb-service-p-835.html?cPath=18>].

Microsoft SQL Server 2014 In-Memory OLTP: Hekaton

Mahir Zajmović, Hadžib Salkić, Fakultet poslovne informatike, Sveučilište/Univerzitet „VITEZ“
Vitez, BiH

Apstrakt - Hekaton je nova baza podataka optimizirana za pohranu podataka i OLTP opterećenja. Hekaton je u potpunosti implementiran u SQL Server te kao takav ne predstavlja poseban sistem. In-Memory OLTP omogućuje OLTP opterećenja kako bi se postigao znatan napredak u performansama te smanjilo vrijeme obrade transakcija. Korisnik jednostavno kreira memorijski optimizirane tabele. Hekaton tabele su potpuno izdržljive, a pristupa im se pomoću T-SQL-a na isti način kao i standardnim SQL Server tabelama. Upit može referencirati kako Hekaton tabelu tako i „standardnu“ tabelu, a transakcija može ažurirati podatke u obje vrste tabela. T-SQL pohranjene procedure (eng. stored procedures) koje služe samo za pristup Hekaton tabelama mogu da se prevedu u mašinski kod za još veća poboljšanja performansi. Mašina je dizajnirana za visok nivo konkurentnosti dok blokiranje sistema je svedeno na minimum.

Ključne riječi – baza podataka, SQL Server, OLTP, Hekaton, upravljanje

Abstract - Hekaton is new database optimized for data storage and OLTP workload. Hekaton is fully implemented in SQL Server, and as such does not constitute a separate system. In-Memory OLTP workload allows to achieve a significant improvement in the performance and reduce the processing time of transactions. The user simply creates an optimized memory tables. Hekaton tables are completely durable, and they are accessed by using T-SQL in the same way as a standard SQL Server tables. Request can be referenced to Hekaton table and a "reference" table, a transaction can update data in both tables. T-SQL stored procedures that are used only to access Hekaton tables can be translated into machine code for even greater performance improvements. The machine is designed for a high level of competitiveness while blocking system is reduced to a minimum.

Index terms – database, SQL Server, OLTP, Hekaton, control

1. UVOD

SQL Server i drugi veliki sistemi za upravljanje bazama podataka prvenstveno su osmišljeni pod pretpostavkom da je „glavna“ (eng. main) memorija veoma skupa, te da se svi podaci pohranjuju na disku. Ova pretpostavka više ne vrijedi, a tokom posljednjih trideset godina cijene memorije su pale po deset puta za posljednjih pet godina. Danas možemo kupiti server sa procesorom od 32 jezgre i 1TB memorije za relativno malu sumu novca što ranije nije bio slučaj. Veličina OLTP baze podataka upotpunosti se može uklopiti u 1TB. Prepoznajući ovaj trend SQL Server posljednjih godina kreće sa razvojem baze podataka koja je optimizovana za rad na uređajima sa velikom količinom memorije i velikim brojem procesora. Novu bazu podataka nazvali su kodnim imenom „Hekaton“ koja je usmjerena na velika radna opterećenja OLTP-a.[1]

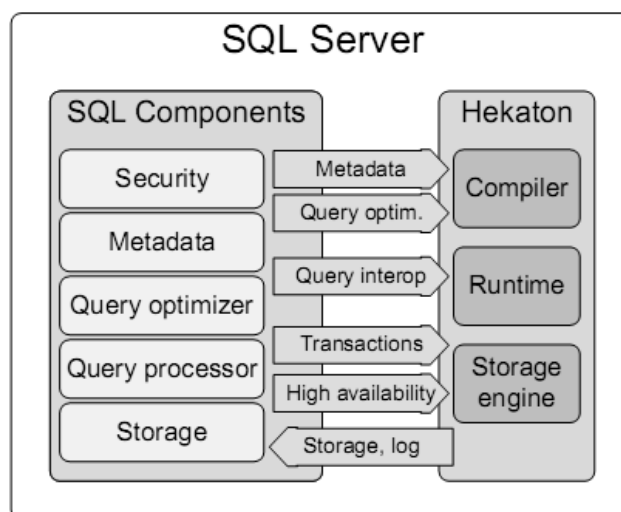
Ono što je najvažnije, Hekaton „engine“ je integriran u SQLServer, i to nije zaseban DBMS. Kako koristiti Hekaton? Korisnik SQL Servera treba da jednu ili više tabela u bazi podataka označi da su memorijski optimizovane. Ovaj pristup korisnicima nudi velike prednosti u usporedbi sa drugim DBMS rješenjima. Prvo, korisnici će izbjeći gnjavažu i trošak drugih DBMS. Drugo, samo veličina performansi jer samo kritične tabele moraju biti u glavnoj memoriji. Treće, pohranjene procedure (eng. stored procedures) pristupaju samo Hekaton tabelama koje se mogu kompilirati u mašinski kod radi poboljšanja performansi.

Memorijski optimizovanim tabelama upravlja Hekaton i iste su u potpunosti pohranjene u glavnoj memoriji. Hekaton tabela može imati nekoliko indeksa te dva indeksa vrste na raspolaganju: hash indeks i range indeks. Sadržaj iz Hekaton tabela može se pregledati te ažurirati pomoću T-SQL upita na isti način kao da su to standardne SQL tabele. T-SQL pohranjene procedure koje pristupaju Hekaton tabelama mogu se kompilirati u mašinski kod. Ovo je daleko najbrži način za postavljanje upita i ažuriranja podataka u Hekaton tabelama.

Hekaton je dizajniran za visoke novoe konkurentnosti, da bi to postiglo oslanja se na particionisanje. Svaki nit (eng. tread) može pristupiti bilo kojem redu u tabeli bez zastoja. Hekaton „engine“ koristi latchfree (lock-free) strukture podataka kako bi se izbjegle smetnje među nitima te na novi izrazito optimističan način kontrolira konkurentsku tehniku kako bi se izbjegle smetnje tokom obrade transakcija.

2. HEKATON ARHITEKTURA

Hekaton „engine“ upravlja korisničkim podacima i indeksima. Ovo omogućava transakcijske operacije sa tabelama i zapisima u njima, hash i range indekse u tabelama, bazne mehanizme za pohranu te mogućnost oporavka i visoku dostupnost. Hekaton compailer podržava T-SQL pohranjene procedure, uključujući upite unutar njih, tabele i indekse metapodataka koje pretvara u mašinski kod. Sistem Hekaton runtime je relativno mali dio koji omogućuje integraciju s SQL Server resursa i služi kao zajednička biblioteka dodatnih funkcionalnosti koje omogućavaju uspješno izvršavanje pohranjenih procedura.[2]



Slika 1: Hekaton glavne komponente i integracija sa SQL Serverom

Hekaton iskorištava brojne servise koji su već dostupni u SQL Serveru. Glavna integracija prikazana je na slici 1.

- **Metapodaci:** Metapodaci o Hekaton tabelama i indeksima pohranjeni su SQL Server katalogu. Korisnici iste mogu pregledavati i upravljati njima koristeći potpuno iste alate kao i da rade sa standardnim tabelama i indeksima.
- **Optimizacija upita:** Upiti koji su unutar pohranjenih procedura optimizirani su pomoću alata za optimizaciju koji dolazi sa SQL Serverom. Hekaton kompajler sastavlja upit u izvornom kodu.
- **Upiti:** Hekaton omogućuje operaterima da za pristup podacima u Hekaton tabelama koriste interpretirane SQL Server planove upita. To uključuje operatere za umetanje, brisanje i ažuriranje podataka u Hekaton tabelama.
- **Transakcije:** standardna SQL Server transakcija može pristupiti i ažurirati podatke, kako u standardnim tabelama tako i u Hekaton tabelama.

- **Visoka dostupnost:** Hekaton je integriran sa AlwaysOn opcijom za visoku dostupnost koji dolazi sa SQL Serverom. Hekaton tabele u bazi podataka mogu se čitati na sekundarnim poslužiteljima.
- **Skladištenje, log:** Hekaton prijavljuje njegove obnove u regularnom SQL Server dnevniku transakcija koji koristi SQL Server fajl datoteku za pohranu kontrolne tačke. Hekaton tabele se automatski obnavljaju kad se obnovi i baza podataka.[3]

3. HEKATON TABELA

U narednom tekstu prikazat ćemo SQL kod za kreiranje baze podataka, Hekaton tabele te umetanja zapisa u Hekaton tabelu.[4] Prvo ćemo da kreiramo bazu podataka pod imenom „ManTest“:

```
USE [master]
GO
CREATE DATABASE [ManTest]
    CONTAINMENT = NONE
    ON PRIMARY (
        NAME = N'ManTest',
        FILENAME = N'D:\SQLServer2014\MSSQL\DATA\ManTest.mdf',
        SIZE = 5120KB,
        FILEGROWTH = 1024KB
    )
    LOG ON (
        NAME = N'ManTest_log',
        FILENAME = N'D:\SQLServer2014\MSSQL\Log\ManTest_log.ldf',
        SIZE = 1024KB,
        FILEGROWTH = 10%
    )
GO
```

Zatim kreiramo FileGroup i vezu sa bazom podataka:

```
USE [ManTest]
GO
-- Add MEMORY_OPTIMIZED_DATA filegroup to the database.
ALTER DATABASE [ManTest]
    ADD FILEGROUP [ManTestFG] CONTAINS MEMORY_OPTIMIZED_DATA
GO
-- Add file to the MEMORY_OPTIMIZED_DATA filegroup.
ALTER DATABASE [ManTest]
    ADD FILE
        ( NAME = ManTestFG_file1,
          FILENAME = N'D:\SQLServer2014\MSSQL\DATA\Datasample_database_1') -- You might
need to check and correct the path here.
    TO FILEGROUP ManTestFG
GO
```

SQL Server FileGroup je potreban za pohranu memorijski optimiziranih podataka u bazu podataka SQL Servera. FileGroup garantira za trajnost podataka u slučaju pada servera. U slučaju pada servera podaci se iz FileGroup-a učitavaju u memoriju.

Sada ćemo kreirati memorijski optimizovanu tabelu:

```
-- Create memory optimized table and hash indexes on the memory optimized table:
CREATE TABLE dbo.Person_memoryOptimizedTable
(
```

```
    PersonID INT NOT NULL
        PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT = 1024),

    PersonName NVARCHAR(100) NOT NULL,

    DateAdded DATETIME NOT NULL

) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_AND_DATA)
GO
```

Razlika između standardne i Hekaton tabele jeste što prilikom kreiranja iza naziva tabele upisujemo „memoryOptimizedTable“. Nakon što smo kreirali Hekaton tabelu sada je potrebno da umetnemo zapise u istu:

```
-- Inserting records into the Memory Optimized Table from [Person] table in
[AdventureWorks] DB:

INSERT INTO dbo.Person_memoryOptimizedTable
SELECT

    [BusinessEntityID],

    CONCAT([FirstName], ' ', [MiddleName], ' ', [LastName]) AS PersonName,

    GETDATE() AS DateAdded

FROM [AdventureWorks2012].[Person].[Person]
```

Nemoguće je umetnuti zapise u Hekaton tabele na više baza podataka. Ovim primjerom pokazali smo način na koji se kreira jedna Hekaton tabela te način na koji se vrši umetanje podataka u istu.[5]

4. ZAKLJUČAK

Hekaton je nova baza podataka s ciljanim radnim opterećenjem OLTP-a razvijena od strane Microsoft-a. Optimizirana je za velike količine memorije i veliki broj procesora. U potpunosti je integrisana u SQL Server koji omogućava korisnicima da iskoriste značajna poboljšanja performansi koja se postižu upotrebom Hekatona. Hekaton postiže visoke performanse i skalabilnost pomoću vrlo učinkovite latch-free strukture podataka.

AlwaysOn opcija koja dolazi sa SQL Serverom osigurava visoku dostupnost i transparentnost. Hekaton „engine“ isporučuje više od jednog reda veličine poboljšanje učinkovitosti i skalabilnosti uz minimalne promjene korisničkih aplikacija ili alata.

LITERATURA

- [1.] Florian Funke, Alfons Kemper, Thomas Neumann, *HyPersonic Combined Transaction AND Query Processing*, PVLDB 4(12): 1367-1370
- [2.] Cristian Diaconu, Craig Freedman, Erik Ismert, Per-Åke Larson, Pravin Mittal, Ryan Stonecipher, Nitin Verma, Mike Zwilling, *Hekaton: SQL Server's Memory-Optimized OLTP Engine*, Microsoft
- [3.] Martin Grund, Jens Krüger, Hasso Plattner, Alexander Zeier, Philippe Cudré-Mauroux, Samuel Madden, *HYRISE - A Main Memory Hybrid Storage Engine*, PVLDB 4(2): 105-116
- [4.] <http://sqlwithmanoj.wordpress.com/tag/hekaton/>
- [5.] <http://thomaslarock.com/2013/08/sql-server-2014-in-memory-oltp-hekaton-useful-links/>