

Integracija heterogenih sistema u cilju povećanja funkcionalnosti poslovnog sistema

Miloš Pantelić
SON, Služba za razvoj informacionih sistema
JKP "Infostan
Beograd, Republika Srbija
pmilos@infostan.co.rs

Ljubica Masal, Predrag Miladinović
SON, Služba za projektovanje i programiranje
JKP "Infostan
Beograd, Republika Srbija
ljubica.masal@infostan.co.rs, mpedja@infostan.co.rs

Sadržaj — Ugovaranje realizacije poslova, unutrašnjih poslovnih procesa ili funkcija sa nekom drugom kompanijom ili osobom, u poslednje vreme se sve više povećava, kako u privatnom tako i u javnom sektoru u skoro svim poslovnim sistemima. Da bi se prevazišao problem nekonzistentnosti podataka, jedinstvenog sagledavanja izvora podataka koji se mogu pojaviti na različitim platformama i lokacijama nameće se potreba za razvojem vremensko-prostornog integratora osnovnih komponenti poslovnog sistema, na osnovu koga će se na višem nivou realizovati poslovne funkcije u skladu sa konkretnim zadacima na određenim radnim mestima u okviru izabrane organizacione strukture. Cilj ovog rada je sagledavanje mogućnosti kreiranja takve arhitekture sistema koja će obuhvatiti sve elemente informacionog sistema bez obzira na izvor podataka, konkretno aplikativno rešenje ili protokol za razmenu podataka.

Ključne reči – poslovni procesi; integracija; informacioni sistemi; organizaciona struktura; heterogeni sistemi; metodologija razvoja; arhitektura sistema; architecture-driven modernization; knowledge discovery meta-model.

I. UVOD

U proteklim godinama, modernizacijom informacionih sistema obezbeđene su određene dobiti za poslovne sisteme. Postavlja se zahtev, da se analizira arhitektura softvera koji se koristi kao podrška tekućim potrebama, pre svega u cilju održavanja, što se kasnije posmatra i kao dobit za projektne timove od kojih se traži migracija na druge platforme ili na modernija okruženja. Da bi uspeali da zadovoljimo klijente i njihove potrebe, moramo da koristimo unutrašnji potencijal preduzeća, da sagledamo potencijal konkurencije i da u skladu sa individualnim sposobnostima zaposlenih uz saradnju i sinergiju sa konkretnim potrebama i ograničenim resursima, ostvarimo misiju organizacije [1]. S obzirom da se već godinama unazad, u našem okruženju određene poslovne aktivnosti i strateške odluke usmeravaju u zavisnosti od političkih promena, za opstanak poslovnog sistema i konkretnog informacionog sistema, mora se naći modus koji će sve postojeće i buduće promene na dovoljno fleksibilan način prihvatiti kako u funkcionalnom tako i u operativnom smislu. Autori ovog rada će sagledati neka od postojećih rešenja koja se koriste kao okvir posmatranja i rešenja sličnih, dinamičkih sistema, koji pravovremeno mogu odgovoriti na nove zahteve i koji će biti stabilni bez obzira na prirodu promena. U oblasti

razmatranja mogućnosti da se izvesni procesi ili poslovne funkcije prepuste ugovornom obavezom trećim licima, sagledaće se osnovne i modaliteti delegiranja odgovornosti kao i kriterijumi za izbor postojećih outsourcing modela. Postoje različiti modeli i pristupi u razvoju alata za transformaciju modela i šablona koji generišu modele u tekst. Korišćenjem nekih od postojećih alata progresivno se povećava lanac koji se može koristiti u projektima različitih migracija, u transformaciji postojećih aplikacija koji su u klijent server arhitekturi, na *n-tiers* i *SOA* arhitekture [2]. Takav lanac sadrži tri koraka: ekstrakcija sveobuhvatnog modela postojeće aplikacije iz izvornog koda, datoteka za konfiguraciju, razvojnih repozitorijuma, zatim, transformacija modela postojeće aplikacije u sveobuhvatni model ciljne aplikacije ili ciljni model i na kraju generisanje izvornog koda iz modela ciljne aplikacije. Usled različite prirode i tehnološko heterogenog nasleđa postojećih sistema, postoji nekoliko načina za izvlačenje validnih modela. Neki predlažu da se formira generički model i meta-model koji se može proširivati. Osnovni okvir, uključujući implementaciju određenih standarda i skup smernica koji se do sada dobro pokazao u praksi, pri kreiranju vlastitog rešenja u iznalaženju modela za različite – nasleđene sisteme, podrazumeva otkrivanje modela, razumevanje određenih gledišta i transformaciju modela u novi softverski sistem. Kao polazna osnova za predlog modela koristiće se KDM (*knowledge discovery meta-model*) [3], a uzori za pronalaženje rešenja će biti ADM (*architecture-driven modernization*) [4] u okviru koga će se na transparentan način doći do arhitekture sistema koja će zadovoljavati potrebe određenog poslovnog sistema.

II. MODELI AUTSORSINGA

Pri izboru modela outsourcing tehnologije polazi se sa stanovišta za koju organizaciju je namenjen, mada se može vršiti i kombinacija modela. Do danas su poznata tri različita tipa modela u praksi. To su model za državnu upravu, model zasnovan na ceni i projektni model [5].

Model za državnu upravu se zasniva na strategiji usvajanja projekata ili projektne organizacije ili projektnog tima. Ovaj model ima tri različite smernice i to stručne usluge ili pojačanje osoblja sa stručnim licima, zajedničko upravljanje i upravljanje uslugama ili potpuni outsourcing. Sva tri modela imaju svoje prednosti i mane.

Model zasnovan na ceni odnosi se na definisanje smernica za plaćanje dobavljačima ili određivanje cene za određene radne naloge. Postoje različiti tipovi određivanja cene koji se koriste u praksi, a radi sagledavanja celine mogu se razvrstati u tri vrste: vreme i materijal, upravljanje kapacitetima i projekti sa fiksnim cenama.

Projektno/ugovorni posao ili hibridni model je kombinacija različitih karakteristika modela za državnu upravu i modela zasnovanih na ceni. Ako se IT organizacija posmatra kao celina mogu se uvideti sve prednosti i mane navedenih modela i mogu se parcijalno koristiti delovi modela u operacijama kroz različite projekte ili kao podrška ugovorenim poslovima.

Sa druge strane, ako se govori o dugoročnoj saradnji dve organizacije na ostvarenju zajedničkog cilja, a da se pri tome orjentišu na određeni rezultat i uspeh, mogu se uočiti određene prednosti autsorsinga [6]. Pored značajnog **smanjenja troškova, objektivnost u donošenju odluka** dolazi u prvi plan. Prevazilaze se lični interesi unutar samih organizacija i uticaji koji se odražavaju na sagledavanje realnog stanja posla i izbora bolje alternative za organizaciju. **Iskustvo u upravljanju** projektima i neophodno **znanje** u obavljanju sličnih poslova i **dostupnost** stručnih lica javlja se kao dodatna prednost. Nedostaci primene autsorsing pristupa ogledaju se u **promeni navika i načina obavljanja** već ustaljenih procesa i procedura. Javljaju se poteškoće u prihvatanju novih procesa i pravila od strane zaposlenih. Za većinu organizacija nedostatak znanja i iskustva u upravljanju projektima javlja se kao glavni razlog prihvatanja autsorsing rešenja. Na kraju, kao glavni nedostatak ovog pristupa je i **dugoročna zavisnost od dobavljača**, jer se svo znanje i do određenog trenutka prikupljeno iskustvo prenosi na drugu kompaniju, uključujući i ekspertizu upravljanja projektima. U slučaju da se ne uspostavi partnerski odnos, klijenti se odlučuju da najosetljivije i najvažnije delove svog poslovanja zadrže, birajući pri tome da manje delove projekta autsorsuju [7].

III. KLAUD ILI NOVI POSLOVNI MODEL

Klaud sistem obezbeđuje dinamičke resurse, što znači da je moguće prilagođavanje u hodu. Na ovaj način jednostavnije je uraditi rekonfiguraciju, odnosno promeniti opterećenje i raspodeliti optimalno korišćenje postojećih resursa. Nivo usluga koje pruža ovo okruženje zavisi od dogovora dobavljača - provajdera i kupca. Resursi u smislu infrastrukture, aplikacija i podataka raspoređeni su širom mreže, gde se preko iste distribuiraju usluge jednog ili više korisnika. Isporuka proizvoda (usluga, rešenja) vrši se u realnom vremenu putem mreže. Samom provajderu omogućeno je da kreira ili daje nove usluge agregacijom postojećih usluga i da prilagodi rešenja na različite modele distribucije. Kod klauz tehnologija mora se voditi računa o softverskoj aplikaciji, programskoj platformi, skladištenju podataka, kao i o računarskoj infrastrukturi. Nivo apstrakcije je predstavljen u više celina: *hardver kao servis, softver kao servis* i *podaci kao servis*. Ove tri celine zajedno čine *platformu kao servis*. Svaki sloj predstavlja nivo apstrakcije koje korisnik ne vidi. Korisnik ima pristup osnovnim i potrebnim komponentama, čime mu je pojednostavljen pristup

resursima kao i sama funkcionalnost sistema, tako da se startovanje - korišćenje aplikacije svodi na jednostavan mehanizam izbora iz liste potrebnih funkcija i unos predviđenih parametara. Prema slojevitom modelu, klauz je podeljen u pet slojeva apstrakcije [8]:

- aplikacioni sloj,
 - softversko okruženje,
 - softverska infrastruktura,
 - jezgro softvera i
 - hardver.
- A. *Aplikacioni sloj* - klauza je najvidljiviji sloj krajnjem korisniku. Pristup je obično obezbeđen putem veb portala, što predstavlja spoj korisnika sa ostalim servisima klauza. Sam servis u aplikacionom sloju može biti sastavljen od mreže servisa ali za samog korisnika to je vidljivo kao jedan jedini servis.
- B. *Sloj softverskog okruženja* - obezbeđuje programski jezik okruženja za programere klauz aplikacija. Takvo okruženje nudi niz dobro definisanih interfejsa za programiranje aplikacije i interakciju sa drugim klauz aplikacijama.
- C. *Sloj softverske infrastrukture* - obezbeđuje sredstva za komunikaciju sa drugim višim slojevima koje koriste aplikacioni i sloj softverskog okruženja. Usluge koje spadaju u ovaj sloj su podeljene na računarske resurse, skladištenje podataka i komunikaciju. Ovaj sloj omogućava fleksibilno skladištenje podataka, odnosno visoku dostupnost, pouzdanost, dobre performanse i konzistentne podatke.
- D. *Sloj jezgra softvera* - predstavlja softversko okruženje za upravljanje fizičkim servisima u skladištima podataka. Ovi slojevi softverskog jezgra su obično implementirani kao operacija sistemskog jezgra, nadzor virtuelne mašine ili posredničko grupisanje.
- E. *Na samom kraju nalazi se sloj fizičkog hardvera*. Hardver može biti zakupljen od provajdera, što je tipična ponuda tradicionalnih autsorsing projekata.

IV. PROCESI KAO OSNOVA ZA INTEGRACIJU

Na pitanje - kako procesi mogu uticati na postizanje željenih integracija heterogenih informacionih sistema kao i na načine za dinamičko i fleksibilno integrisanje izvora podataka pronalazimo odgovor u vezi šablona ili šema sa ontologijom [9].

A. *Podešavanje šema sa ontologijama*

Ontologija integracije šema višestrukih heterogenih izvora koji obuhvataju globalne šeme, polazi od lokalne šeme ili posmatra svaki izvor informacija posebno sa svojom ontologijom, a različite ontologije direktno povezuju na višem nivou apstrakcije. Hibridna rešenja kombinuju dva pristupa na osnovu postojećih - nasledenih ontologija. To su *pristup*

jedinstvenog izvora informacija i pristup korišćenja elemenata zajedničkog izvora podataka ili određenog rečnika.

Mapiranje se koristi za opis odnosa između ontologija. Mapiranje može biti definisano ručno ili automatski pomoću leksičkih odnosa, utemeljeno na najvišem nivou i semantičkoj sličnosti.

B. Podešavanje (navođenje) operacija po principu procesa

Izvor informacija ima višestruku upotrebu. Za svaku pojedinačnu upotrebu, on nudi korisnicima različite operacije. U jednostavnim slučajevima, takve operacije su kao udaljene procedure, ali generalno, posebno u slučaju web portala mogu biti složenije i mogu zahtevati veću interakciju korisnika. Nad izvorom informacija obavljaju se lokalne operacije, kao parcijalni zadaci za neki određeni slučaj korišćenja, npr. kalkulacije ili upiti. U ovom koraku, mora se modelirati svaki slučaj upotrebe izvora informacija kao proces, opisan u određenom jeziku. Opisani proces, koristi ontologiju iz prethodnog odeljka u službi bilo kog objekta koji učestvuje u tom procesu. Lokalne operacije su navedene kao lokalne aktivnosti i mogu biti mapirane na konkretnoj primeni. Svaka interakcija sa korisnikom je navedena kao aktivnost komunikacije. U principu, sam izvor informacija može biti integrisani izvor informacija, odnosno može da koristi operacije drugih izvora informacija. U tom slučaju onaj koji poseduje informaciju, može ali ne mora da to prikaže. U prvom slučaju on navodi operaciju kao proces aktivnosti, a u drugom slučaju kao lokalne aktivnosti.

C. Integracija izvora komponovanjem procesa

Zamislimo situaciju u kojoj su javno dostupni mnogi izvori informacija i njihove specifikacije - ontologije i opisi. Opisi procesa služe kao semantički opisi odgovarajućim operacijama. Da bi imali integrisani sistem koji obuhvata neke informacije izvora, korisnik sistema identifikuje svaki slučaj korišćenja, način implementacije funkcionalnosti korišćenjem operacija koje pruža drugi izvor informacija ili uzima lokalno dostupne informacije. Operacije drugih izvora informacija mogu biti ugrađene kao proces aktivnosti ili lokalne aktivnosti date opisom slučaja korišćenja. Osnovne aktivnosti u procesu imaju konkretne odgovornosti i kao procesi specifikacije i one se direktno izvršavaju. Mapiranje između različitih ontologija može biti integrisano kao lokalna operacija podrške interoperabilnosti na nivou podataka.

V. UZORI ZA KREIRANJE MODELA

A. KDM

KDM je specifikacija od OMG kompanije koja se bavi postavljanjem standarda i distribuiranjem objektno orijentisanih sastava poslovnih procesa. KDM je predstavnik postojećih softverskih sistema i njihovih okruženja koji definišu zajedničke metapodatke koji su potrebni za integraciju upravljanja životnim ciklusom aplikacije (mogućnost uvođenja poslovnog menadžmenta u programsko inženjerstvo), radi povezivanja arhitekture, kodova, testiranja i praćenja procesa. Cilj KDM-a je da osigura međusobno

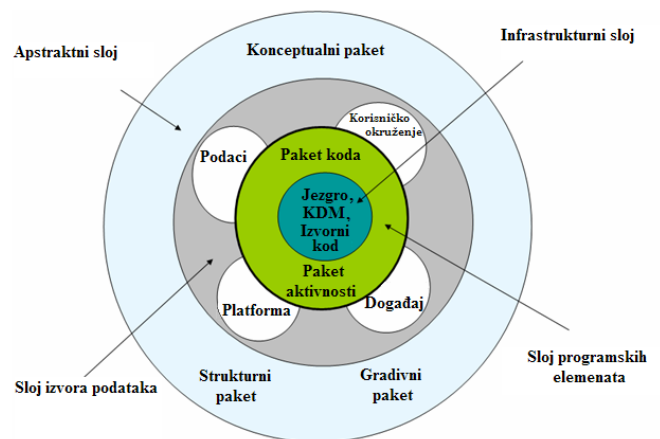
delovanje alata za održavanje, evoluciju, procenu i modernizaciju.

„KDM sadrži 12 paketa koji su podeljeni u četiri sloja: [10].“

- **Infrastrukturni sloj,**
- **Sloj programskih elemenata**
- **Izvorni sloj**
- **Apstraktni sloj**

a) **Infrastrukturni sloj** – sadrži *paket jezgra*, *KDM paket* i *paket izvorog koda* koji skupa obezbeđuju malo zajedničko jezgro za sve ostale pakete, za sve elemente koje model sadrži u postojećem sistemu i veze koje slede iz elemenata meta modela i vraćaju se ka izvornom kodu elemenata radi proširenja jedinstvenog mehanizma. Osnovni paket određuje nekoliko modela koji su korišćeni od strane drugih KDM paketa.

b) **Sloj programskih elemenata** - sastoji se od kodnog i paketa aktivnosti. *Paket koda* predstavlja elemente programiranja koji su određeni programskim jezicima (tipovi podataka, procedure, klase, metode, promenljive,...). *Paket KDM koda* obezbeđuje veći nivo detalja i integrisanost sa delom arhitekture softverskog sistema. *Prikaz tipova podataka* izjednačen je sa ISO standardom. *Paket aktivnosti* obuhvata elemente ponašanja aplikacije koji uključuje detaljnu kontrolu protoka informacija.



Slika 1. KDM se sastoji od 12 paketa raspoređenih u 4 sloja [11]

c) **Izvorni sloj**- predstavlja radno okruženje postojećem softverskom sistemu. Sastoji se od paketa platforme, ulazno-izlaznog paketa, paketa događaja, i paketa podataka. Paket platforme - predstavlja radno okruženje softvera, koji je povezan sa operativnim delom, posrednički programi, uključujući i protok između komponenti sistema. Paket korisničkog okruženja predstavlja saznanje iz prakse i rada na postojećem softverskom sistemu. Paket događaja predstavlja saznanje vezano za događaje i ponašanje postojećeg dela softvera. Paket podataka - predstavlja proizvode povezane sa

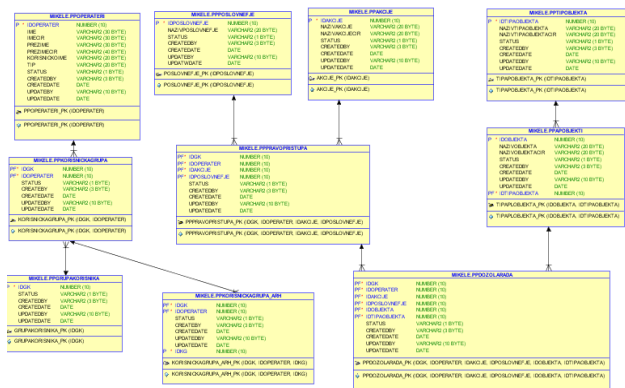
stalnim podacima kao što su indeksirane datoteke, relacione baze i ostala skladišta podataka.

d) **Apstraktni sloj** - predstavlja domen i sažetak primene aplikacije. Sastoji se od *konceptualnog, strukturnog i gradivnog paketa*. *Konceptualni paket* - odnosi se na poslovni domen znanja i poslovna pravila, ukoliko se ta informacija može dobiti iz postojeće aplikacije. *Strukturni paket* opisuje elemente meta-modela za predstavljanje logičkog dela u sistemskom softveru u podsistemima, slojevima i komponentama.

e) **Gradivni paket** predstavlja inženjerski pogled na sastav softvera, odnosno sastavne elemente.

VI. PREDLOG REŠENJA

Postojeći izvori podataka u našem razmatranom sistemu nalaze se na različitim platformama. Raznovrsnost struktura podataka iziskuje da se na određenom sloju objedine izvorni podaci bez obzira na mesto gde se fizički nalaze, bez obzira da li su organizovani u mrežne, relacione, objektne baze podataka, ili se nalaze u nekim datotečnim sistemima ili su pak u vidu tabelarnih prikaza ili tekstualnih datoteka. Na osnovu različitih, heterogenih sistema, stvara se prividni a u suštini stvarni popis komponenti sistema na jedinstvenom nivou koji će biti osnova za sagledavanje nasledenih resursa. Na istom nivou, zavisno od potreba, obezbeđen je mehanizam proširenja u slučaju dodatnog uključenja novih izvora podataka. Da bi sistem bio osiguran od neovlašćenog pristupa na različitim nivoima integratora osmišljen je i razvijen sloj za kontrolu pristupa. Relacioni model sloja za kontrolu pristupa predstavljen je na slici 2.



Slika 2. Relacioni model sloja za prava pristupa

Za komponente sistema zavisno od dodeljenih prava vlasnika izvora podataka, mogu se razviti potrebne funkcije za ažuriranje postojećih podataka, dodavanje novih ili brisanje postojećih podataka. Sistem za pretraživanje po izabranim kriterijumima se dodeljuje na svim nivoima elemenata sistema. Napomenimo da se za svaki autonomni već nasledeni sistem može dodati skup dodatnih svojstava u posebno kreiranim tabelama u okviru vlastite šeme sa pravima pristupa koja nisu ograničena od neke druge instance, tako da se mogu upotpuniti, obogatiti jednom rečju

modernizovati poslovne funkcije na određenom radnom mestu u okviru posmatranog poslovnog sistema. Na četvrtom sloju lica zadužena za organizaciju posla definišu tipove aktivnosti i radne zadatke koji se javljaju kao jedinične operacije implementirane kao ustaljene procedure ili funkcije. Na osnovu tako definisanih postupaka povezuju se i grupišu za određene tipove radnih mesta, odnosno delegiraju se odgovornosti za kasniju funkcionalnost tog segmenta posla. Izvesni delovi sistema su nabavljeni od različitih dobavljača: sistem za upravljanje dokumentacijom – poslovi vezani za pisarnicu, poslovi vezani za evidenciju radnog vremena i poslovi vezani za fakturisanje - sistem objedinjene naplate za koji je korišćena autsorsing tehnologija. Da bi adekvatno koristili podatke iz pomenutih sistema i njihovih autonomnih izvora podataka sa sopstvenim ontologijama, moramo na višem nivou razviti ontološki sistem koji obuhvata sve komponente našeg sistema.

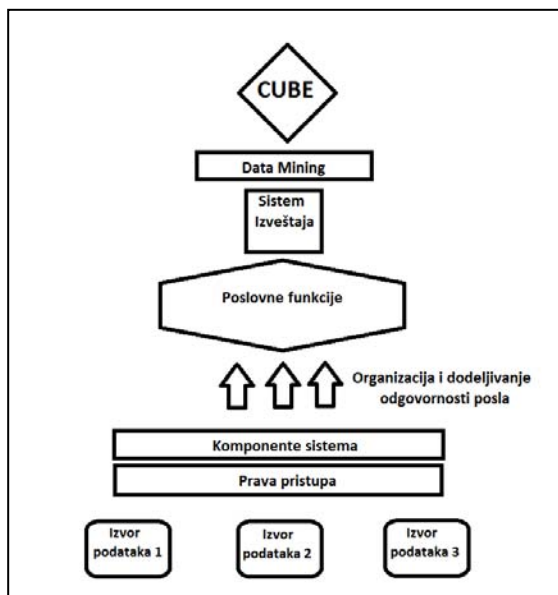


Slika 3. Komponente sistema

Na sledećem sloju u skladu sa potrebama a zavisno od unutrašnjih uticaja bez obzira na tip organizacione strukture, mogu se određeni poslovi grupisati u veće celine, organizovati i delegirati određenim licima a samim tim pored funkcionalnog zaduženja odrediti odgovornost. Na taj način se na željenim nivoima ili slojevima formira celina koja sa svojim funkcionalnim elementima posmatra kao efikasno sredstvo za ostvarenje zajedničkog cilja poslovnog sistema. Rezultat kvalitetnih podataka sa očuvanim integritetom na svakom sloju na kome se koristi u sistemu postaje osnova za kvalitetno izveštavanje u zahtevanoj formi i sa određenim sadržajem. Sistem izveštavanja se pozicionira na posebnom sloju, gde se posredstvom poznatih komponenti sistema pristupa konkretnim izvorima podataka. Za izveštavanje se može koristiti neki od dostupnih sistema koji u grafičkom okruženju mogu na jednostavan način prikazati sadržaj i u zahtevanoj formi, na pismu i jeziku koji se koristi u regionu. Tako kreirani izveštaji se publikuju na određenom serveru u organizaciji pogodnoj za pronalazjenje i korišćenje u konkretnim funkcionalnim segmentima

informatičnog sistema. U određenom razvojnom alatu se mehanizam pozivanja određenog izveštaja svodi na definisanje parametara u skladu sa politikom definisanom u sloju sa pravima pristupa.

Primer koji se u našem sistemu daje, koristi izvore podataka na jednoj produkcionoj instanci sa zahtevanim podacima, testnoj instanci i na instanci starog sistema koji se nalazi na serveru koji je nezavistan u odnosu na prve dve instance.



Slika 4. Arhitektura sistema za integraciju

ZAKLJUČAK

Korišćenjem postojećih saznanja koja se koriste u oblasti modernizacije postojećih sistema možemo doći do fleksibilne arhitekture zahtevanog informatičnog sistema za određeni poslovni sistem i pri tome prevazići različitost koja se ogleda kroz korišćenje nekonzistentnih i autonomnih sistema u cilju povećanja funkcionalnosti. Arhitektura ovako osmišljenog sistema je neka vrsta paralelnih i distribuiranih informatičnih sistema koji se mogu porediti sa virtuelizovanim računarima. Sama tehnologija je u sve većoj primeni posebno kod složenih informatičnih sistema. Neke od prednosti ovakve organizacije informatičnog sistema su pristojne cene, dostupnost servisa, dobra raspodela resursa, jednostavna implementacija i administracija.

Svaki poslovni informatični sistem mora biti sposoban da manipuliše velikim brojem podataka. Takav informatični sistem je predstavlja tehnološku platformu koja omogućava da firma integriše i koordinira svoje poslovne procese. Sistem mora da centralizuje obradu podataka, ali obezbeđuje i

deljenje informacija kroz sve funkcionalne nivoe, kao što mora da obezbedi i hijerarhiju upravljanja podacima ili procesima. Ovako nešto može se postići integracijom heterogenih sistema kroz predloženi model arhitekture.

LITERATURA

- [1] <http://www.iim.ftn.uns.ac.rs/pom/attachments/article/138/Upravljanjepromenama.pdf>
- [2] William M. Ulrich, Philip Newcomb, Information Systems Transformation: Architecture-Driven Modernization Case Studies, Morgan Kaufmann, 2010, San Francisco, California, USA.
- [3] <http://www.omg.org/cgi-bin/doc?admtf/2008-06-10>
- [4] <http://www.omg.org/technology/kdm/index.htm>
- [5] <http://technologyoutsourcingblog.com/types-of-technology-outsourcing-models>
- [6] <http://www.executivebrief.com/outsourcing/outsourced-project-management>
- [7] <http://www.kdmanalytics.com>
- [8] Lamia Youseff, Maria Butrico, Dilma Da Silva <http://www.cs.ucsb.edu/~lyouseff/CCOntology/CloudOntology.pdf>
- [9] Sudhir Agarwal and Peter Haase, Process-Based Integration of Heterogeneous Information Sources, Institute of Applied Informatics and Formal Description Methods (AIFB), University of Karlsruhe, Germany.
- [10] <http://www.omg.org/spec/KDM/1.0/PDF/>
- [11] <http://www.omg.org/spec/KDM/1.0/PDF/>, strana 31, slika 7.1 - Layers, packages, and separation of concerns in KDM.

ABSTRACT

Arranging of internal business processes or functions with another company or person, recently, more and more increases in both, the private and public sector in almost all business systems. To overcome the problem of inconsistent data, consideration of a single data source that can occur on a variety of platforms and locations, there is a need for the development of space-time integrator core components of the business system, upon which the higher-level business functions implemented in accordance by the specific tasks at certain positions within the chosen organizational structure. The aim of this study is to gain the possibility of creating such a system architecture that will encompass all the elements of an information system, regardless of the source of the data, specifically applicable solutions or protocol for data exchange.

Key words – bussines process; integration; information systems; oragnizational structures; heterogeneous systems; development methodology; system architecture; architecture-driven modernization; knowledge discovery meta-model.

DATA INTEGRATION OF HETEROGENEOUS SYSTEMS IN ORDER TO INCREASE THE FUNCTIONALITY OF THE BUSINESS SYSTEM

Milos Pantelic, Ljubica Masal, Predrag Miladinovic