

## SEMANTIČKA INTEGRACIJA POSLOVNIH APLIKACIJA SEMANTIC INTEGRATION OF BUSINESS APPLICATIONS

Nenad Aničić

**REZIME:** Zahtev za efikasnom elektronskom komunikacijom koji se postavlja pred poslovnim sistemima, otvorio je niz pitanja kako o načinu integracije tako i o interoperabilnosti ovih sistema. Osnovni problem kojim se rad bavi je problem integracije poslovnih aplikacija koji proističe iz strukturne i semantičke neusaglašenosti poslovnih podataka. U radu se definiše postupak i algoritmi usaglašavanja semantike šema poslovnih dokumenata, transformacije instanci poslovnih dokumenata i razvija prototip softvera koji ovaj postupak i algoritme implementiraju.

**KLJUČNE REČI:** Integracioni standardi, Semantički Web, Integracija poslovnih aplikacija, Semantička interoperabilnost, XML šema, OWL.

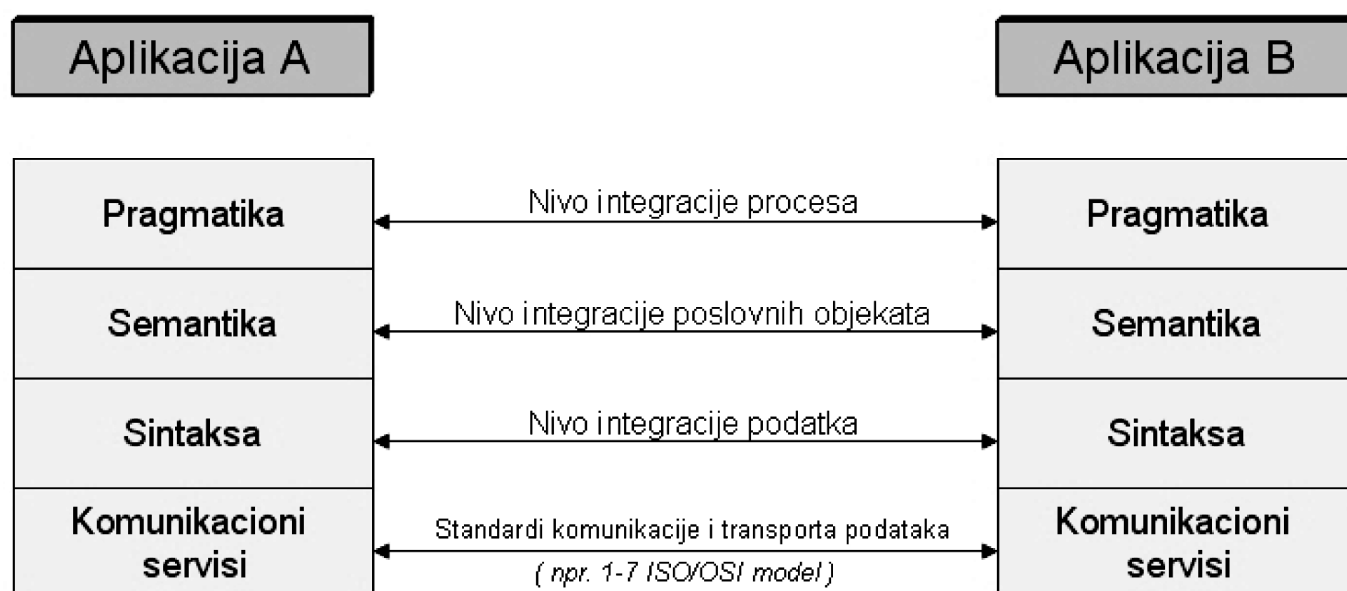
**ABSTRACT:** The increasingly important requirement for efficient electronic communication has brought along a fundamental issue that all business systems need to address: What are effective and efficient ways of achieving integration and interoperability with other systems. The basic problem that this paper attempts to solve is the problem of business application integration which is a consequence of structural and semantic differences among business data. This paper defines an algorithm for semantic reconciliation of business document schemas, an algorithm for transformation of business documents, and provides a software prototype that implements the proposed integration method.

**KEY WORDS:** Integration standards, Semantic Web, Integration of business applications, Semantic interoperability, XML Schema, OWL.

### 1. UVOD

Uspeh Interneta i XML jezika omogućio je razvoj novih, takozvanih Business-to-Business (B2B) tržišta u kojima savremene kompanije moraju da sve efikasnije integrišu svoje poslovne sisteme koji praktično kontrolišu sve aspekte poslovanja ovih kompanija. Stoga, efikasna integracija, radi razmene poslovnih podataka sa drugim eksternim poslovnim sistemima, postaje veoma važna funkcija svake savremene poslovne aplikacije. Ovaj zahtev za efikasnom elektronskom komunikacijom koji se postavlja pred poslovnim sistemima, otvorio je niz pitanja kako o načinu integracije tako i o interoperabilnosti ovih sistema.

Pod integracijom poslovnih aplikacija podrazumevamo razmenu instanci poslovnih dokumenata između dve aplikacije tako da je omogućena njihova interoperabilnost i pored toga što su interfejs modeli ovih aplikacija nezavisno razvijani i, samim tim, neophodno različiti. Termin interoperabilnost, u osnovi označava mogućnost dva sistema da razmenjuju informacije na način koji omogućava korektnu interpretaciju ovih informacija u različitim lokalnim kontekstima ovih aplikacija. Interoperabilnost, prema tome, zahteva nivo kompatibilnosti između poslovnih sistema da bi se minimizirala transformacija podataka koji se razmenjuju sa ciljem da se omođu korektna interpretacija razmenjenih po-



Slika 1. – Nivoi integracije poslovnih aplikacija

dataka. U teoriji, ovi sistemi bi trebalo da budu rigidno usaglašeni sintaksno i semantički u skladu sa aplikativnim domenskim standardom. Međutim, u praksi ovaj način integracije nije moguć zbog različitih tehnoloških pristupa, postojanja zaostavština (uključujući i tehnološki zastarele sisteme) koji su u upotrebi ili zbog autonomnosti svakog od datih sistema.

Problem integracije poslovnih sistema može se posmatrati preko četvoronivoskog apstraktnog modela koji obuhvata: komunikacione servise, sintaksu, semantiku i pragmatiku, i koji je prikazan sa slici 1.

Na najnižem nivou tj. nivou komunikacionog servisa problemi integracije poslovnih aplikacija su posledica različitih hardverskih platformi, operativnih sistema, mrežnih i komunikacionih protokola. Na sledećem, sintaksnom nivou, problem integracije je posledica heterogenosti sintakse i strukture podataka, tj. korišćenja različitih formata za reprezentaciju podataka, kao i korišćenje različitih struktura i šema za formiranje i reprezentovanje poslovnih objekata - entiteta. Na sledećem višem nivou, semantika poslovnih objekata zavisi od načina prikazivanja realnih poslovnih koncepata i konteksta u kojem se ti podaci o njima interpretiraju. I na najvišem nivou imamo integraciju poslovnih aplikacija koje čine neki složeni poslovni proces (pragmatika). Pragmatika definiše način integracije poslovnih aplikacija u okviru nekog poslovnog procesa.

Da bi se mogla izvršiti integracija poslovnih aplikacija i obezbedila efikasna interoperabilnost između aplikacija, potrebno je preduzeti usaglašavanje poslovnih aplikacija na svim nivoima. Problemi najnižeg nivoa interoperabilnosti (komunikacionih servisa) prevaziđeni su zahvaljujući razvoju i uspešnoj primeni zajedničkih mrežnih komunikacionih protokola i "middleware" tehnologija. S obzirom na uspešnu primenu ovih tehnologija problem integracije na ovom nivou je danas lako rešiv i neće biti analiziran u radu.

U osnovni, integracija poslovnih aplikacija može se posmatrati sa dva osnovna aspekta, aspekta integracije poslovnih procesa i aspekta integracije podataka. Razvijeni standardi za modelovanje i implementaciju distribuiranih poslovnih procesa (WSDL, Web servisi, WFMC standard, BPEL, na primer) omogućavaju efikasno modelovanje i implementaciju distribuiranih poslovnih procesa, pa se ni sa ovim najvišim nivoom apstraktnog opisa problema interoperabilnosti ovaj rad neće detaljnije baviti. Osnovni problem kojim se rad detaljno bavi je problem integracije poslovnih aplikacija koji proističe iz sintaksne i semantičke neusaglašenosti poslovnih podataka. Usaglašavanje poslovnih aplikacija tj. poslovnih podataka na sintaksnom i semantičkom nivou je neophodan uslov za uspešnu interoperabilnost poslovnih aplikacija u okviru nekog workflow sistema.

## 2. OPIS PROBLEMA

Jedan od veoma važnih aspekata integracije poslovnih aplikacija je rešavanje problema heterogenosti podataka. Pretpostavimo da dve poslovne aplikacije dele isti konceptualni model (tj. da imaju usaglašene koncepte poslovnih objekata), onda se problem integracije može javiti na sintaksnom nivou i to zbog različite reprezentacije strukture poslovnih objekata. Na primer, ukoliko dva sistema koriste isti format kao standard za sintaksu razmene podataka, onda različiti redosled elemenata u strukturi dokumenta koji opisuju isti semantički koncept, mogu prouzrokovati konflikt i onemogućiti interoperabilnost ovih aplikacija. Kako se za komunikaciju na sintaksnom nivou koristi XML kao de-facto standard za razmenu podataka, to se problem integracije ovih aplikacija svodi na rešavanje strukturne različitosti formata podataka koji se razmenjuju.

Problem integracije na semantičkom nivou javlja se u slučaju kada postoji neusaglašenost specifikacije poslovnih objekata. Tada korišćenje istog sintaksnog formalizma ili reprezentacionog standarda neće rešiti problem u integraciji ovih sistema. Stoga je neophodno na neki način obezbediti automatizaciju provere semantičke usaglašenosti i interpretacije poslovnih objekata. Ovo zahteva proveru usaglašenosti poslovnih modela koji se integrišu i definisanje formalnih preslikavanja iz jednog modela u drugi, čime je omogućena osnova za transformaciju podataka između poslovnih aplikacija. Postavlja se pitanje na koji način je moguće formalno predstaviti koncepte poslovnih modela i omogućiti automatsko usaglašavanje datih koncepata, i sa druge strane, kako u ovako usaglašenim sistemima omogućiti rešavanje problema sintaksne integracije tj. integracije na nivou podataka.

Problemi integracije poslovnih aplikacija se danas rešavaju dogovorom učesnika u okviru neke industrijske grane, regiona ili procesa, rigoroznim usaglašavanjem poslovnih koncepata i tipova poslovnih dokumenata. To je rezultiralo pojavom više savremenih integracionih standarda. Primeri takvih standarda su Open Application Group (OAGIS) [OAG, 2005], RosettaNet, Universal Business Language (UBL) i drugi. Svi ovi standardi koriste opšti generički XML format za razmenu poslovnih dokumenata, tj. definišu tipove poslovnih dokumenata preko XSD šeme (*W3C XML Schema*). Razlike koje postoje u ovim standardima su u pristupu specifikacije standarda. UBL i OAGIS razvijaju standard horizontalno tj. definišu opšti zajednički poslovni model, dok RossetaNet koristi vertikalni pristup definisanju standarda tj. orijentisan ka specifičnim industrijskim domenima. U osnovi savremeni integracioni standardi definišu: 1) skup XML šema koji čini biblioteku zajedničkih komponenti (koncepata) koje predstavljaju gradivne blokove za tipove dokumenata (na primer *Address*, *Item*, *Payment*,  *itd.*), 2) mali skup XML šema koji definiše zajedničke tipove poslovnih dokumenata (na primer *Order*, *Invoice*) i

3) mehanizam za definisanje novih specifičnih tipova dokumenata i ograničenja. Ove XSD šeme su modularne, proširive i višestruko korišćene u skladu sa specifikacijama XML-a [W3C\_XML, 2004] i XSD šeme [W3C\_XSD, 2001]. *Horizontalni* standardi svoje biblioteke komponenti baziraju na konceptualnom modelu informacionih komponenti koje se nazivaju poslovni informacioni entiteti (Business Information Entities - BIEs). Ove komponente su gradivni blokovi za definisanje tipova poslovnih dokumenata koji se reprezentuju u XSD šemi u skladu sa jasno definisanim upustvima (*guidelines*) i pravilima za imenovanje i dizajn (Naming and Design Rules NDR).

U okviru biblioteka poslovnih entiteta zajedničke komponente se opisuju na taj način što se svi njihovi atributi definišu kao opcioni. Time se obezbeđuje više alternativa za njihovo korišćenje u skladu sa kontekstom u kojem će biti korišćene. Pri definisanju šeme poslovnog dokumenta mora se navesti koji atributi su obavezni tj. koji je minimum osobina koje svaka od komponenti mora da ima da bi se obezbedilo njeno "prepoznavanje" u konkretnom kontekstu. Pri opisu šeme mogu se definisati i druga poslovna ograničenja i to najčešće preko *Schematron-a* [Schematron, 2005].

Ovako definisani "otvoreni" integracioni standardi doveli su do pojave slojevitosti integracionih standarda, odnosno do definisanja novih "podstandarda" na bazi osnovnog (baznog standarda). Podstandarde usaglašavaju užii skupovi korisnika specificirajući specifičan skup njihovih tipova dokumenata. Na primer, STAR (*Standards in Automotive Retail*) [STAR, 2006] i AIAG (*Automotive Industry Action Group*) [AIAG, 2006] su konzorcijumi koji opisuju poslovne dokumente preko OAGIS standarda, svaki u svom domenu. *OAGi virtuelni poslovni objektni model*, omogućava poslovnoj aplikaciji kompanije da oko sebe napravi omotač virtuelnog objekta, kroz upotrebu *OAGi API*. Poslovne aplikacije u ovom modelu komuniciraju, kroz integracionu osnovu, razmenom *OAGi Business Object Document (BOD)* između interfejsa virtuelnih objekata. BOD predstavlja skup podataka koji se razmenjuje između sistema u i van preduzeća kao deo neke poslovne transakcije. Oni sadrže dve osnovne komponente koje ih identifikuju, "noun" koji identifikuje poslovne podatke i "verb" koji identifikuje akciju koju treba sprovesti nad podacima. Nezavisan način definisanja standardnih tipova dokumenata doveo je do potrebe za proverom konzistentnosti novih tipova poslovnih dokumenata u skladu sa baznim slojem i otvorio je probleme u komunikaciji i usaglašenosti poslovnih aplikacija definisanih u različitim domenima-grupama.

XML šeme preko kojih se opisuju tipovi poslovnih dokumenata, iako koriste iste zajedničke XML šeme sa opisima poslovnih koncepata i njihovih osobina (na primer OAG XSD šeme: *Fields, Components, CodeLists, Meta*) mogu biti i sintaksno i semantički neusaglašeni. Zbog toga je neophodno razviti postupak kojim će se preko zajedničkih XML šema ti dokumenti sintaksno i seman-

tički porediti i podaci (tj. instance poslovnih dokumenata) transformisati iz jednog formata u drugi. Drugim rečima, primena "otvorenih" savremenih integracionih standarda otvorila je nove potrebe za proveru kompatibilnosti poslovnih modela aplikacija i potrebe za razvojem transformacionih servisa kako da bi se omogućila njihova interoperabilnost.

### 3. METOD SEMANTIČKE INTEGRACIJE

U dinamičkom, heterogenom i distribuiranom sistemu kao što je to Web, semantička interoperabilnost se može ostvariti samo korišćenjem formalizacija pomoću kojih se skriva kompleksnost i implementacioni detalji različitih interfejs modela poslovnih aplikacija. Za ostvarivanje semantičke interoperabilnosti očigledno je da su potrebni modeli i alati za opis semantičkih sadržaja poslovnih poruka koje se razmenjuju, kao i tehnika za automatsko ekstrahovanje i klasifikaciju podataka i meta - podataka.

U osnovi ovaj rad zasnovan je na ideji da se komunikacija između različitih autonomno razvijenih poslovnih aplikacija može realizovati korišćenjem zajedničke (referentne) ontologije kao modela za zaključivanje o zajedničkoj semantici određenog poslovnog domena. Zajednička ontologija je definisana preko specificiranih semantičkih koncepata integracionog standarda. Za reprezentaciju zajedničke ontologije korišćen je OWL DL jezik [W3C\_OWL, 2004], kao ontološki jezik. Koncepti lokalnih ontologija su takođe reprezentovani u OWL jeziku sa jednoznačno definisanim semantičkim preslikavanjima prema zajedničkoj ontologiji.

Predloženi semantički integracioni metod se oslanja na DL formalizam i algoritme za logičko zaključivanje nad DL modelom. Sve faze semantičke integracije opisane su formalno korišćenjem DL terminologije [Aničić, 2006]. Kao implementaciono okruženje za realizaciju ovog metoda korišćeno je Java implementaciono okruženje. Dati prototip je realizovan kroz nadogradnju (implementaciju novih komponenti eng. *plugin*) Protege [Protege, 2005] ontološkog editora. Protege editor omogućava manipulaciju sa ontološkim definicijama kao i pokretanje algoritama logičkog rezonovanja koji su korišćeni u radu. Predloženi metod primenjen je na OAGi standard, dok su kao lokalne ontologije poslovnih aplikacija uzete lokalne šeme STAR i AIAG konzorcijuma, koje baziraju svoje aplikacije na OAGi standardu.

#### 3.1. Opis metoda semantičke integracije

Predloženi metod semantičke integracije sastoji se iz dva osnovna koraka:

1. **Kreiranje integracionog ontološkog modela.** Ovaj korak obuhvata: a) kreiranje zajedničke ontologije poslovnih entiteta, i b) kreiranje ontoloških modela tipova dokumenata. Kreiranje zajedničke ontologije poslovnih

entiteta vrši se transformacijom XSD opisa konceptata standarda po kome se kreiraju šeme poslovnih dokumenata. Kreiranje ontoloških modela tipova dokumenata vrši se transformacijom njihovih XML šema. Kao reprezentacioni ontološki jezik koristi se OWL (Ontology Web Language). U poglavlju 3.3 se diskutuju metamodeli preko kojih se ove ontologije mogu kreirati i objašnjava zbog čega je u radu izabran OWL. Transformacioni algoritam (opisan detaljno u [Anicic et al., 2006]) kojim se kreiraju ontologije definiše skup transformacionih pravila definisanih u skladu sa NDR pravilima i uputstvima za reprezentaciju poslovnih entiteta i dokumenata standarda.

**2. Transformacija podataka.** Ovaj korak obuhvata primenu algoritma za transformaciju podataka tj. instanci poslovnih dokumenata između poslovnih aplikacija baziranih na integracionom ontološkom modelu. Transformacija podataka vrši se prevođenjem XML opisa poslovnog dokumenta u ontološki opis gde se primenom mehanizma logičkog zaključivanja klasifikuju podaci u skladu sa ciljnim lokalnim ontološkim modelom. Ovako klasifikovani podaci se serijalizuju u ciljni XML format i validiraju u skladu sa ciljnom XSD šemom poslovnog dokumenta.

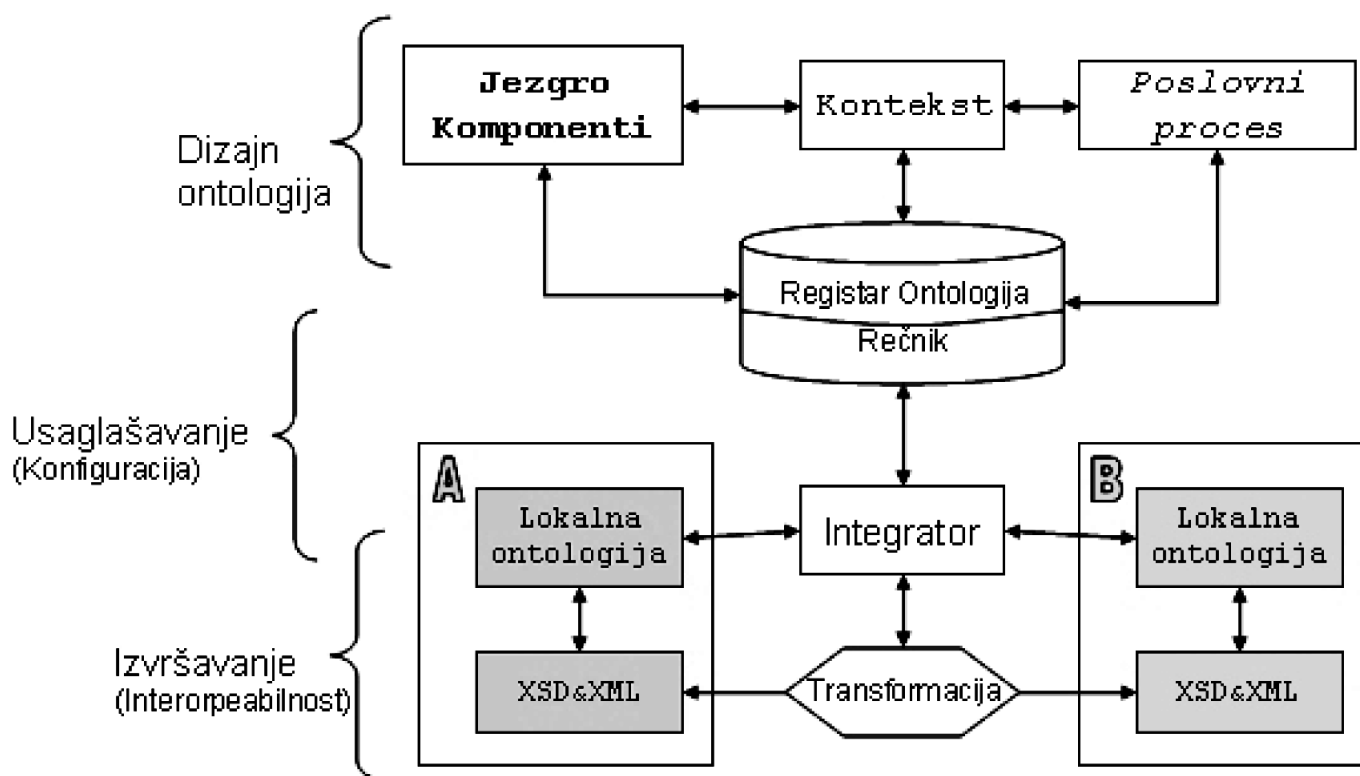
Predložena metoda semantičke integracije omogućava korišćenje postojećih interfejs modela poslovnih aplikacija baziranih na XSD šemama. Naime, tipovi poruka koje se razmenjuju između poslovnih aplikacija u B2B prostoru formalno su reprezentovane preko XSD šeme, a kreirane su uz pomoć jasno definisanih "uputstava za dizajn i imenovanje" ovih tipova poruka.

Semantička nadogradnja, koja se predlaže, bazira se na automatizaciji preslikavanja XSD šema konceptata u odgovarajuće ontologije. Na ovaj način, omogućena je nova generacija integracionih standarda koja omogućava formalnu specifikaciju konceptualanog modela, jasnu logičku specifikaciju ograničenja poslovnih dokumenata, automatizaciju usaglašavanja-definisanja tipova dokumenata, i proveru kompatibilnost postojećih. Sa druge strane, ovako definisan standardni integracioni model omogućava lakšu implementaciju transformacionih servisa da bi se obezbedila šira primena standarda. Predloženi semantički integracioni metod oslanja se na formalizme Semantičkog Web-a za predstavljanje znanja koji su usko povezani za teorijom deskriptivne logike (Description Logic - DL) i algoritmima za logičko rezonovanje nad DL modelom (bazom znanja).

U nastavku se daje opšta arhitektura sistema koja omogućava rešavanje problema semantičke i sintakso-strukturne integracije.

**3.2. Arhitektura sistema semantičke integracije**

Pod integracijom poslovnih aplikacija podrazumeva se razmena pojavljivanja poslovnih aplikacija (*business document instances*) između dve poslovne aplikacija koje su bazirane na dva različita modela sadržaja poslovnih dokumenata (*business document content models*) tj. interfejs modela tako da se postigne uspešna razmena podataka (interoperabilnost). Pod uspešnom razmenom podataka se podrazumeva takva razmena podataka kojom se zadržava ista semantika poslovnih objekata u datom kontekstu.



Slika 2. – Arhitektura sistema integracije

Formalno, integracija podataka poslovnih aplikacija se može definisati preko trojke (G, L, M) gde je

- G je globalna tj. zajednička ontologija, koja predstavlja skup objekata, aksioma i ograničenja koji definišu opšte poslovne koncepte (jezgro poslovnih komponenti) u odgovarajućem jeziku
- L je lokalna ontologija neke konkretne poslovne aplikacije koja definiše opšte poslovne koncepte u konkretnom kontekstu
- M je skup iskaza koji povezuju elemente između zajedničke i lokalne ontologije (G i L)

Arhitektura sistema na kojem se bazira dati integracioni metod prikazana je na slici 2.

Kao što se može videti iz arhitekture sistema, identifikuju se tri faze integracije. Prva je kreiranje ontologija u nekom definisanom kontekstu u okviru nekog definisanog poslovnog procesa, bazirano na definisanom skupu jezgra komponenti i informacija o kontekstu (skup poslovnih pojmova, njihovih osobina i veza). Zatim se vrši usaglašavanje lokalnih ontologija i dodaju aksiomi kojim se omogućavaju i definišu procedure za transformaciju podataka. I na kraju, u fazi izvršavanja tj. fazi interoperabilnosti sistema koriste se informacije o ontologijama kako bi se izvršila transformacija podataka.

*Jezgro komponenti* predstavlja ključni deo arhitekture i sadrži informacije o stvarnim poslovnim konceptima, vezama između njih i definisanje kontekstualnog opisa kojim se opisuje na koji način se informacioni entiteti (bilo osnovni bilo agregirajući) mogu koristiti u konkretnom poslovnom scenariju. Jezgro komponenti sadrži individualne delove poslovnih informacija (poslovni objekti) i skup poslovnih entiteta koji često idu zajedno i koje se nazivaju agregirani poslovni objekti.

Informacije o definisanim lokalnim ontologijama se smeštaju u *Registar* ontologija. Ovaj registar treba da omogući skup servisa za razmenu informacija između poslovnih partnera. Takođe, treba da omogući pristup kako XML šemama koje opisuju tipove poslovnih dokumenata tako i njihovim ontologijama kojima se oni mapiraju u zajedničku ontologiju jezgra komponenti (*CoreComponents*). U okviru identifikacije poslovnih dokumenata potrebno je registrovati i sva poslovna pravila koja važe za tipove poslovnih dokumenata u datom kontekstu. Ova poslovna pravila treba da budu formalno iskazana preko ontološkog jezika kojim se omogućava specifikacija bogatijih semantičkih veza i ograničenja u definisanim ontologijama.

Usaglašavanje lokalnih ontologija za transformaciju predstavlja sledeću fazu u definisanju integrisanog ontološkog modela. U okviru ove faze koriste se lokalne ontologije da bi se definisale veze između koncepata lokalnih ontologija. Ovaj korak je poluautomatski jer se moraju uključiti i dizajneri ontologija za specifikovanje pravila mapiranja podataka i specifikovanje veza koje se nisu mogle identifikovati na osnovu njihovih

ontoloških opisa. Sve informacije, iskazi o preslikavanjima se definišu u *Rečniku*, koji će se koristiti kao osnova za transformaciju. Rečnik treba da obezbedi servise za lak pristup, manipulaciju i validaciju specifikiranih informacija.

Prve dve faze u definisanoj arhitekturi integrisanog sistema obezbeđuju formiranje integrisanog ontološkog modela koji se sastoji od registra ontologija i rečnika preslikavanja.

Transformacija podataka se u potpunosti oslanja na integrisani ontološki model. Dakle, u ovoj fazi koriste se informacije iz *Registra* ontologija koje se transformišu, sa svim poslovnim pravilima i *Rečnika* preslikavanja koji specifikira dodatne aksiome transformacije. Komponenta sistema *Integrator* treba da obezbedi primenu algoritama transformacije između bilo koja dva usaglašena modela poslovnih aplikacija.

### 3.1. Implementacija sistema semantičke integracije

Za realizaciju date arhitekture metoda semantičke integracije neophodno je izabrati formalizam koji na jedinstven način obezbeđuje definisanje ontologija tipova poslovnih dokumenata i njihovih strukturnih i semantičkih ograničenja. Ontologije su veoma važne za integraciju aplikacija zato što omogućavaju zajedničko, deljivo razumevanje podataka koji postoje u domenu problema integracije aplikacija. Glavna svrha ontologija je da omoguće komunikaciju između kompjuterskih sistema koja je nezavisna od individualne systemske tehnologije, informacione arhitekture i aplikacionog domena. Ontologije mogu varirati ne samo po sadržaju nego i po strukturi, implementaciji, nivou opisa, konceptualanom opsegu i jezičkoj specifikaciji. Veliki broj jezika može biti korišćen za njenu izgradnju, uključujući SQL - relacioni račun i programske jezike kao što je Prolog. Međutim, najčešće se koriste jezici koji su prvenstveno specifikovani da podrže konstruisanje ontologija. Jezici kao što je KIF (njegov naslednik CommonLogic) i Open Knowledge Base Connectivity (OKBC) model su primeri tih formalizama, a koriste se i kao osnova za druge ontološke jezike. Ono što je zajedničko za sve ove jezike je da imaju zajedničku apstrakciju: deklarativnu reprezentaciju znanja. Deklarativan način opisa ontologija omogućava visok nivo konceptualne apstrakcije koji omogućava neprogramerima da razumeju, specifikiraju i dinamički menjaju opise i poslovna ograničenja u konkretnom domenu. Ovi opisi su izvršni tj. moguće je upravljati njima kao svim drugim podacima, nezavisno od koda.

Zahtev koji je postavljen u specifikaciji datog metoda je da formalizam mora da bude zasnovan na XML sintaksi jer se ona koristi kao dominantan format za razmenu podataka i informacija između heterogenih sistema. Stoga je za realizaciju ove arhitekture izabran Web ontološki jezik - OWL. Osim što nudi mogućnost formalne specifikacije integrisanog ontološkog modela, ovaj formalizam obezbeđuje i mehanizme zaključivanja

deskriptivne logike čime se olakšava implementacija algoritama usaglašavanja i transformacije.

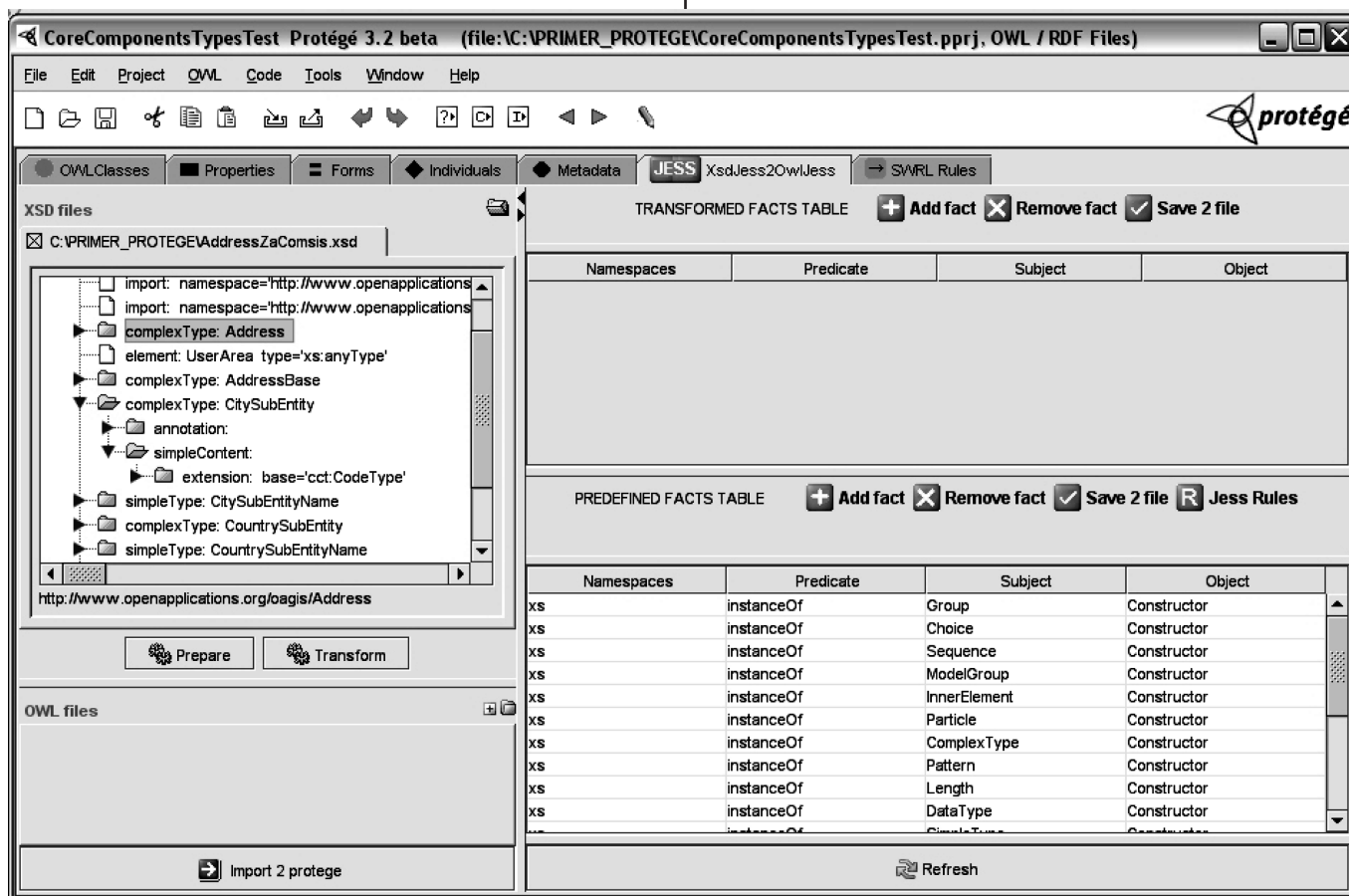
Dakle, OWL ontologije i XML šema zajedno omogućavaju dinamičko upravljanje sistemom zasnovanim na znanju i osnovu za efikasan način integracije poslovnih aplikacija. Ontologija omogućava domensku teoriju, dok XML šema definiše strukturu kontejnera podataka. Ontologija obezbeđuje definisanje deljive domenske (poslovne) terminologije, a XML se koristi da obezbedi njenu prenosivost. Na ovaj način, korišćenjem tehnologija semantičkog Web-a, obezbeđuje se šira primena metoda i laka funkcionalna nadogradnja poslovnog sistema. Primenom ovih tehnologija formalno se opisuju algoritmi za kreiranje i usaglašavanje ontologija, kao i za transformaciju XML podataka. Za definisanje Jezgro komponenti i specifikaciju konteksta izabran je horizontalni integracioni standard OAGi koji definiše biblioteku zajedničkih komponenti (konceptata) koje predstavljaju gradivne blokove za tipove dokumenata, kao i način specifikacije tipova dokumenta u konkretnom kontekstu preko BOD koncepta.

Za primenu predloženog metoda integracije kreiran je prototip alat kojim se automatizuju i primenjuju prethodno opisani algoritmi. Alat je implementiran kao plug-in ontološkog editora Protégé, uz podršku Racer alata za me-

hanizme zaključivanja nad OWL ontologijama i Jess (Java Expert System Shell) ekspertnog sistema za zaključivanje nad bazom znanja preko proceduralnih pravila.

U okviru definisanog okruženja razvijena su i dva plug-in modula. Prvi Xsd2owlTab obezbeđuje automatizaciju kreiranja ontološkog modela. Drugi RacerTab obezbeđuje transformaciju podataka i validaciju i pretraživanje podataka uz interaktivnu i direktnu komunikaciju sa Racer sistemom.

Kreiranje ontologija je omogućeno preko Xsd2OwlTab modula (slika 3). Ovaj modul obezbeđuje implementaciju pravila preslikavanja između XSD-a i OWL-a. Pravila su reprezentovana kao Jess pravila (kao alternativa data je i XSLT transformacija). Ovaj modul nudi i mogućnost definisanja novog skupa pravila i njihovu primenu na datu XML šemu. Modul automatski uključuje opis meta modela XML šeme, koji se onda mogu direktno koristiti u opisu pravila. Korisnički interfejs je jednostavan i preko njega se bira željena XML šema, gde se onda automatski kreiraju činjenice o datom modelu i prebacuju u Jess bazu znanja. Proceduralna Jess pravila koja su definisana prvo kreiraju instance OWL modela koji se zatim direktno prebacuju u ontološki editor. Kao rezultat se dobija ontologija čije su izmene, validacija i usaglašavanje sa drugim ontologijama omogućeni preko Protégé -a. Ovaj



Slika 3. – Ekran za kreiranje ontologija iz XML šeme Xsd2OwlTab

modul se na isti način koristi i za transformaciju instanci XML šeme u OWL individue nad kojima se vrši validacija i transformacija.

Transformacija podataka se vrši preko modula *IntegratorTab*. Ovaj modul omogućava primenu algoritma za validaciju OWL individua u zatvorenom svetu zaključivanja (CWR). Koraci u transformaciji podataka se oslanjaju na Racer mehanizme zaključivanja nad OWL ontologijom (i za TBox i za ABox). Zatim, koraci klasifikacije i ekstrakcije u odnosu na integrisani ontološki model se mogu pokrenuti pojedinačno. Prvo se bira ciljna ontologija, a zatim se nakon kreiranja integralnog ontološkog modela vrši ekstrakcija podataka. Kreiranje pomoćnog "najspeficijalnijeg koncepta" za svaku individuu se vrši u okviru novog pomoćnog TBox-a (zahvaljujući mogućnostima Racer-a za rad sa više TBox-ova) u kome se vrši klasifikacija, sračunavanje obuhvatljivosti i provera zadovoljivosti novokreiranih konceptata.

#### 4. ZAKLJUČAK

U radu je opisan semantički integracioni metod kojim se analizira mogućnost primene tehnologija semantičkog Web-a na rešavanje problema integracije poslovnih aplikacija. Ovaj novi integracioni metod je opisan kroz dva koraka: kreiranje integracionog modela i transformaciju podataka. Prvi korak se odvija u vreme dizajna i obezbeđuje proveru kompatibilnosti ontologija. Kao rezultat dobija se generalizovana ontologija, kao podrška definisanju lokalnih normalizovanih ontologija. Usaglašavanje ovih ontologija rezultira integracionim ontološkim modelom koji je neophodan za sledeći korak - transformaciju podataka. Transformacija podataka se odvija u vreme izvršavanja, gde se instance poslovnih dokumenata transformišu na osnovu opšteg algoritma transformacije. Algoritam transformacije bazira se na mehanizmu logičkog zaključivanja i predstavlja osnovu za semantičku interoperabilnost aplikacija.

Metod semantičke integracije koji je predložen u radu bazira se na semantičkoj nadogradnji integracionih standarda. Ovi standardi predstavljaju dobru osnovu za integraciju, jer su poslovne komponente, tj. pojmovi koji oni nude već usaglašene u dugogodišnjoj primeni tih standarda. Tako je glavni problem u definisanju zajedničke ontologije olakšan. Predloženo rešenje omogućava formalizaciju reprezentacije znanja o konceptima poslovnog sloja (integrisanog ontološkog modela) i definiše algoritam transformacije podataka između

različitih interfejs modela. Predložen metod uspešno je primenjen na OAGi integracionom standardu. Prototip koji je razvijen za primenu datog metoda omogućava osnovu i za testiranje interoperabilnosti poslovnih aplikacija i usaglašavanje njihovih ontoloških modela.

#### 5. LITERATURA

- [1] [AIAG, 2006] Automotive Industry Action Group (AIAG), Web site, [Online]. Available: <http://www.aiag.org>, (Accessed 2006).
- [2] [Anicic et al., 2005] Anicic, N., Ivezic, N., and Jones, A.: *An Architecture for Semantic Enterprise Application Integration Standards*. INTEROP-ESA'05.
- [3] [Anicic & Ivezic, 2005] Anicic, N., Ivezic, N. (June 2005), *Semantic Web Technologies for Enterprise Application Integration*, ComSIS, Vol.2, No.1, 119-144.
- [4] [Aničić, 2006] Aničić, N.: *Integracija poslovnih aplikacija primenom tehnologija semantičkog Web-a*, Doktorska disertacija, Fakultet organizacionih nauka, Beograd, 2006.
- [5] [Anicic et al., 2006] Anicic N., Ivezic N., Marjanovic Z.: *Mapping XML Schema to OWL*, In *Pre-proceedings of I-ESA'06*, March 2006.
- [6] [Baader et al., 2003] Baader, F., Horrocks, I., Sattler, U.: *Description Logics as Ontology Languages for the Semantic Web: Lecture Notes in Artificial Intelligence*. Springer-Verlag, (2003).
- [7] [Haarslev & Moller, 2001a] Haarslev, V., and Moller, R.: *Description of the RACER system and its applications*. In *Proceedings International Workshop on Description Logics (DL-2001)*, 2001.
- [8] [OAG, 2005] The Open Applications Group, Web site, [Online]. Available: <http://www.openapplications.org>, (Accessed 2005).
- [9] [Protege, 2005] Protégé, free, open source ontology editor and knowledge-base framework, Web site, [Online]. Available: <http://protege.stanford.edu>, (Accessed 2005).
- [10] [Schematron, 2005] The Schematron, Web site, [Online]. Available: <http://xml.ascc.net/resource/schematron>, (Accessed 2005).
- [11] [STAR, 2006] Standards for Technology in Automotive Retail (STAR) Web site, [Online]. Available: <http://www.starstandard.org>, (Accessed 2006).
- [12] [W3C\_OWL, 2004] OWL Web Ontology Language Overview, Web site, [Online]. Available: <http://www.w3.org/TR/owl-features>, (2004).
- [13] [W3C\_XML, 2004] Extensible Markup Language (XML) 1.1, Web site, [Online]. World Wide Web Consortium Recommendation, Available: <http://www.w3.org/TR/2004/REC-xml11-20040204>, (2004).
- [14] [W3C\_XSD, 2001] *XML Schema*, World Wide Web Consortium Recommendation, Available at: <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>, (2001).



Nenad Aničić

Fakultet organizacionih nauka, Beograd  
oblast interesovanja: razvoj informacionih sistema, baze podataka, integracija i interoperabilnost poslovnih sistema.