



**Prof. dr. sc.  
Siniša Sribljčić**



**Mr. sc.  
Dejan Škvorc**



**Mr. sc.  
Daniel Skrobo**



**Mr. sc.  
Miroslav Popović**



**Mr. sc.  
Ivan Gavran**



**Ivan Žužak**



**Mr. sc.  
Ivan Benc**



**Mr. sc.  
Ivan Skuliber**



**Dr. sc.  
Andro Milanović**



**Mr. sc.  
Matija Podravec**

**Prof. dr. sc. Siniša Sribljčić, mr. sc. Dejan Škvorc,  
mr. sc. Daniel Skrobo, mr. sc. Miroslav Popović,  
mr. sc. Ivan Gavran, Ivan Žužak**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu, Zagreb, Hrvatska  
*Faculty of Electrical Engineering and Computing,  
University of Zagreb, Zagreb, Croatia*

**Mr. sc. Ivan Benc, mr. sc. Ivan Skuliber**

Ericsson Nikola Tesla d.d., Zagreb, Hrvatska  
*Ericsson Nikola Tesla d.d., Zagreb, Croatia*

**Dr. sc. Andro Milanović**

Combis d.o.o, Zagreb, Hrvatska  
*Combis d.o.o, Zagreb, Croatia*

**Mr. sc. Matija Podravec**

Hrvatske telekomunikacije d.d., Zagreb, Hrvatska  
*Hrvatske telekomunikacije d.d., Zagreb, Croatia*

### ***Ključne riječi:***

**Programirljivo internetsko okružje**

**GRID računarstvo**

**Računarstvo zasnovano na uslugama**

**Integracija i komponiranje usluga**

**CRO-GRID poliprojekt**

### ***Key words:***

**Programmable Internet Environment**

**GRID computing**

**Service Oriented Computing**

**Service integration and composition**

**CRO-GRID polyproject**

## Sažetak:

Brzi razvoj aplikacija zasnovanih na uslugama jedan je od najvažnijih zahtjeva suvremenog informacijskog i telekomunikacijskog tržišta. Ovaj članak predstavlja programirljivo internetsko okružje (PIE - *Programmable Internet Environment*), programski sustav za brzu gradnju raspodijeljenih aplikacija povezivanjem dostupnih usluga. Programirljivo internetsko okružje izgrađeno je prema konceptima i pravilima računarstva zasnovanog na uslugama (*Service Oriented Computing*) i GRID-a, područjima raspodijeljenog računarstva čija važnost znatno raste unazad nekoliko godina. Ovime je omogućeno povezivanje udaljenih računala u jedinstveni sustav te gradnja raspodijeljenih aplikacija slaganjem usluga dostupnih putem privatnih mreža ili putem Interneta. Sustav programirljivog internetskog okružja ostvaren je unutar projekta "CRO-GRID posrednik", dijela poliprojekta "CRO-GRID", čiji je cilj bio izgraditi hrvatsku nacionalnu GRID mrežu, potrebnu programsku potporu te prototipne aplikacije. Istraživanja provedena u okviru projekta "CRO-GRID posrednik" dio su istraživanja koja se izvode u projektu "*Middleware Architecture* (MidArc)", u suradnji Istraživačkog odjela Instituta za telekomunikacije u Ericssonu Nikoli Tesli i Zavoda za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

## Abstract:

Rapid development of applications based on available services is one of the most important demands of today's information and telecommunications market. This paper presents the Programmable Internet Environment (PIE), a software system for composing distributed applications from publicly exposed services. The PIE system has been created according to the paradigms and concepts of the Service Oriented Computing (SOC) and GRID. The PIE system enables connection of separate computers into a system and programming of distributed applications based on the composition of services that are located either on the computers of the system or on the computers accessible by the Internet. The PIE system has been developed as a part of the project entitled CRO-GRID Mediator, within a larger polyproject entitled CRO-GRID. The goals of the polyproject were the creation of the Croatian national GRID network with necessary software and prototype applications. The research done within the project CRO-GRID Mediator is a part of the research done within the project entitled *Middleware Architecture* (MidArc), conducted in collaboration between the Research Department at Ericsson Nikola Tesla's Research and Development Center, and the Department of Electronics, Microelectronics, Computer and Intelligent Systems at the Faculty of Electrical Engineering and Computing, University of Zagreb.

## 1. GRID okolina i računarstvo zasnovano na uslugama

GRID okolina i računarstvo zasnovano na uslugama (SOC - *Service Oriented Computing*) pojmovi su iz područja raspodijeljenog računarstva čiji značaj je znatno narastao unazad nekoliko godina. Većini korisnika koji se računalima služe uglavnom za jednostavne poslovne aplikacije ili multimediju, ovi pojmovi zasad još uvijek nisu zanimljivi. Međutim, akademske institucije i tvrtke počinju koristiti njihove mogućnosti. Rješavanje računski i memorijski vrlo zahtjevnih problema (*Grand Challenge Problems*) nepraktično je i pomoću najjačih super-računala, jer do pronalaska rješenja treba čekati nekoliko dana, pa i nekoliko mjeseci. Primjeri takvih zahtjevnih problema su: analiza velikih baza podataka (red veličina terabajta ili više - npr., analiza ponašanja korisnika ili grupa korisnika), automatizirano donošenje poslovnih odluka, predviđanje ponašanja tržišta, simuliranje promjena klime, meteorološko modeliranje vremena, ponašanje složenih bioloških spojeva i dr.

Upravo primjena koncepta iz GRID svijeta i računarstva zasnovanog na uslugama, omogućava znatno jednostavnije i brže rješavanje navedenih problema. Osnovna ideja je povezati međusobno lokacijski razdijeljena računala i grozdove (*cluster*, skupina surađujućih računala povezanih lokalnom mrežom) u jedinstven sustav koji se krajnjem korisniku predstavlja kao jedno prividno super-računalo. Način na koji se stvara privid super-računala te načini njegovog korištenja jasno odjeljuju GRID koncepte i koncepte računarstva zasnovanog na uslugama.

GRID sustavi povezuju računala u jedinstven prividni sustav zvan prividna organizacija (*virtual organization*). Za sva računala u jednoj prividnoj organizaciji vode se bilješke o trenutačnom stanju sklopovskih parametara - opterećenju procesora, količini slobodne radne memorije, prostora na radnim diskovima, stanju lokalne mreže i dr. Ovisno o tim bilješkama, zadaci koji su poslani na rješavanje GRID sustavu, raspoređuju se na računala koja su trenutačno najprikladnija za njihovo rješavanje. Dakle, u GRID sustavima posebna pažnja posvećuje se radnim svojstvima sklopovlja kako bi se što brže izvršili zadaci poslani u sustav. Potrebno je naglasiti razliku između grozdova računala i GRID sustava, budući da se ta dva pojma često poistovjećuju - grozdovi računala smješteni su na jednoj lokaciji i uglavnom su povezani jednom lokalnom mrežom, dok se jedan GRID sustav može sastojati od međusobno vrlo udaljenih računala smještenih na potpuno različitim pristupnim mrežama.

Sustavi izgrađeni na konceptima računarstva zasnovanog na uslugama labavo povezuju (*loose coupling*) računala i grozdove na Internetu. Računala su međusobno povezana

putem informacija u središnjim registrima ili na osnovi podataka u lokalnim memorijama. Za razliku od GRID sustava u kojima su informacije vezane uz sklopovlje, informacije u sustavima zasnovanim na uslugama sadrže podatke o svim javno dostupnim programskim uslugama koje određeno računalo može izvršavati.

Podaci o programskoj usluzi se sastoje od tehničkih parametara (npr. pristupna točka usluge, potrebni parametri, vrsta izlaznih parametara, koji komunikacijski protokol je potrebno koristiti i dr.) i netehničkih parametara (semantički opis usluge, očekivana brzina odgovora, kvaliteta usluge i dr.). Korisnik povezuje i uslužuje usluge u veću smislenu cjelinu - procese, odnosno raspodijelenu aplikaciju koja izvršava određen složen zadatak. Dakle, dok se GRID sustavi vrlo pojednostavljeno mogu zamisliti kao lokacijski raspodijeljeno super-računalo, sustave zasnovane na uslugama moguće je zamisliti kao veliki repozitorij usluga čijim povezivanjem se dobiva dodana vrijednost - raspodijeljena aplikacija.

Bitno je istaknuti kako su koncepti GRID sustava i računarstva zasnovanog na uslugama tehnološki potpuno neovisni. Oba koncepta predstavljaju niz standarda i smjernica za povezivanje različitih računala u veće cjeline određene namjene. Zbog toga je donedavno postojao veći broj različitih sustava koji su ostvarivali navedene koncepte, ali u različitim tehnologijama (npr. REST, RPC, DCOM, CORBA, i dr.). Međutim, unazad nekoliko godina, zahvaljujući sveprisutnosti Interneta, počeli su prevladavati GRID sustavi i sustavi računarstva zasnovanog na uslugama izgrađeni pomoću skupa standarda za internetske usluge (*Web Services Standards*). Kako je skup standarda za internetske usluge namijenjen različitim oblicima povezivanja raznorodnih računala, upravo ta tehnologija predstavlja osnovu za potpunu platformsku neovisnost GRID sustava i sustava zasnovanih na uslugama.

Od kraja 2000. Istraživački odjel Instituta za telekomunikacije u Ericssonu Nikoli Tesli d.d. sa Zavodom za elektroniku, mikroelektroniku, računalne i inteligentne sustave (ZEMRIS) na Fakultetu elektrotehnike i računarstva (FER) surađuje na području raspodijeljenog računarstva. U okviru te suradnje dosad je razvijeno nekoliko prototipova raspodijeljenih sustava, a posljednji među njima, programirljivo internetsko okružje (PIE - *Programmable Internet Environment*) predstavlja prototip sustava zasnovanog na uslugama. Uz suradnju između Ericssona Nikole Tesle i Fakulteta elektrotehnike i računarstva, programirljivo internetsko okružje je dijelom razvijeno u okviru CRO-GRID poliprojekta, hrvatske nacionalne GRID inicijative.

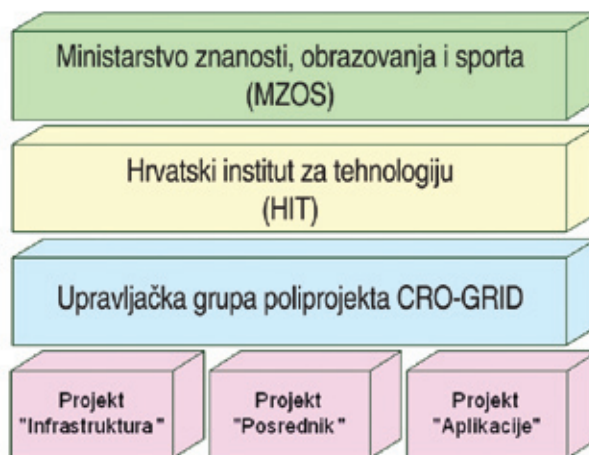
Programirljivo internetsko okružje se može zamisliti kao jednostavan raspodijeljeni operacijski sustav koji omogućava povezivanje računala i njihovih javno dostupnih usluga smještenih bilo gdje na Internetu. Krajnjem korisniku programirljivo internetsko okružje nudi mogućnost

programiranja raspodijeljenih aplikacija povezivanjem usluga smještenih na tim računalima, kao da se te usluge nalaze na korisnikovom stolnom računalu.

## 2. CRO-GRID

Prateći trendove u svjetskom raspodijeljenom računarstvu, tijekom 2003. godine nekoliko različitih istraživačkih grupa (FER i ETK, IRB, SRCE) prijavilo je tehnološke projekte tematski vezane uz GRID područje. Gotovo istodobno, Ministarstvo znanosti, obrazovanja i športa je također prepoznalo važnost GRID tehnologija u daljnjoj informatizaciji Hrvatske. Kako su projekti bili međusobno komplementarni, predloženo je njihovo okrupnjavanje u poliprojekt CRO-GRID koji bi predstavljao hrvatsku nacionalnu GRID inicijativu, istovjetnu nacionalnim GRID projektima drugih zemalja.

Krajem 2003. g. Ministarstvo znanosti obrazovanja i športa je prihvatilo poliprojekt CRO-GRID kao složeni tehnološki projekt unutar programa HITRA (Hrvatski Inovacijski Tehnologijski RAZvitak). Program HITRA je poseban oblik integracije znanstvene i tehnološke politike usmjerene na povezivanje javnog znanstveno-istraživačkog sektora i gospodarstva koju provodi Ministarstvo u cilju promicanja gospodarstva utemeljenog na znanju. Uz Ericssonu Nikolu Teslu koji je aktivno sudjelovao u poliprojektu CRO-GRID, u pojedine podprojekte bile su uključene i mnoge druge hrvatske tvrtke različitih djelatnosti, npr. brodogradilište u Rijeci, Hrvatska pošta, Klara, i dr.



**Slika 1. Ustroj CRO-GRID poliprojekta u okviru programa HITRA**

Slika 1. prikazuje ustroj CRO-GRID poliprojekta u okviru programa HITRA. Poliprojekt se sastojao od tri projekta: Infrastruktura, Posrednik i Aplikacije. Voditelji pojedinih projekata izvještavali su Upravljačku grupu o napretku i rezultatima projekata. Upravljačka grupa nadzirala je i usmjeravala cjelokupan poliprojekt, a sastojala se od predstavnika Ministarstva znanosti, obrazovanja i športa, Hrvatskoga instituta za tehnologiju (HIT), četiri hrvatska

sveučilišta i gospodarstvenih čimbenika uključenih u poliprojekt. Upravljačka grupa izvještavala je o radu i napretku cjelokupnog poliprojekta Hrvatski institut za tehnologiju koje nadzire realizaciju programa HITRA. Hrvatski institut za tehnologiju nadalje je izvještavao Ministarstvo znanosti, obrazovanja i športa o stanju poliprojekta i cjelokupnog programa HITRA.

Ciljeve poliprojekta moguće je podijeliti na tehnološke i netehnološke. Osnovni tehnološki cilj bio je izgraditi nacionalnu GRID infrastrukturu. Pritom se ne misli samo na sklopovlje, već i na programsku potporu potrebnu za rad GRID sustava i pripadnih aplikacija u stvarnom vremenu. Glavni netehnološki zadatak bio je povezati hrvatsku akademsku zajednicu u jedinstven hrvatski GRID sustav koji bi bio stavljen na raspolaganje akademskoj i gospodarskoj zajednici. Iz ovog cilja proizašla je potreba za zapošljavanjem i obrazovanjem mladih kadrova koji će svoja stečena znanja i iskustva prenijeti hrvatskim znanstvenicima te nastaviti daljnje ostvarivanje hrvatskog GRID sustava. Također, uočena je nužnost povezivanja CRO-GRID-a u europske i svjetske GRID inicijative, odnosno, povezivanja hrvatskog GRID sustava s postojećim GRID sustavima.

Uključivanjem u CRO-GRID Ericsson Nikola Tesla je nastavio i proširio dosadašnju uspješnu suradnju s hrvatskom akademskom zajednicom. Tehnički izazovi na koje smo naišli za vrijeme realizacije poliprojekta, prilikom ostvarivanja različitih prototipova predstavljaju vrijedno iskustvo i znanje primjenjivo u idućoj generaciji (ne samo GRID) mreža. Korištenje različitih eksperimentalnih tehnologija omogućilo nam je procjenu korisnosti i isplativosti nadolazećih tehnologija.

## 2.1. Projekt Infrastruktura

Projektom Infrastruktura upravljao je Sveučilišni računski centar (SRCE). Osnovni zadatak toga projekta bio je izgradnja hrvatske GRID infrastrukture koja povezuje sva hrvatska sveučilišta u jedinstveni GRID sustav. Uz akademsku zajednicu, korištenje nacionalnog GRID sustava omogućeno je i gospodarskim čimbenicima. Usporedo s ostvarivanjem infrastrukture, SRCE je sa svojim partnerima obrazovalo znanstvenike za korištenje novih GRID tehnologija.

Zahvaljujući korektno odrađenim poslovima, hrvatski GRID sustav 2006. g. primljen je u srednjeeuropsku federaciju GRID sustava te međunarodni projekt EGEE-II (*Enabling Grids for E-Science II*) gdje su dogovorene aktivnosti i obaveze do 2008. g. U okviru tih aktivnosti Hrvatska će imati mogućnost aktivnog sudjelovanja u daljnjem definiranju i razvoju GRID koncepata te pristup zadnjim dostignućima na tom području.

## 2.2. Projekt Posrednik

Fakultet elektrotehnike i računarstva i Ericsson Nikola Tesla surađivali su na projektu Posrednik. Najvažniji cilj projekta bio je ostvarivanje programske potpore za izgradnju i povezivanje međusobno neovisnih i lokacijski odijeljenih usluga u raspodijeljenu aplikaciju. Posebna pažnja usmjerena je približavanju koncepata raspodijeljenog računarstva krajnjem korisniku. Razvijeno korisničko sučelje, odnosno, programirljivo internetsko okružje, i programska podrška koja se nalazi iza njega, detaljnije su predstavljene u poglavlju 3.

Sam projekt i postignuti rezultati predstavljeni su na mnogobrojnim radionicama i konferencijama. Posebno priznanje projektu predstavljaju pozvana predavanja u američkim tvrtkama (Intel corp., Google inc., i dr.) te sveučilištima (Berkeley, Santa Clara, Irvine i dr.).

## 2.3. Projekt Aplikacije

Projekt Aplikacije sastoji se od četiri podprojekta kojima je cilj bio izgraditi četiri prototipne GRID aplikacije. Vodeće institucije bile su Institut Ruđer Bošković, Fakultet prometnih znanosti u Zagrebu, Prirodoslovno-matematički fakultet u Zagrebu te Fakultet elektrotehnike, strojarstva i brodogradnje u Splitu. Uz akademsku zajednicu na pojedinim aplikacijama sudjelovale su i druge hrvatske tvrtke.

U sklopu ovoga projekta izgrađene su sljedeće aplikacije: Optimizacija organizacije transporta, Indeksiranje web/lokalnih mreža, Modeliranje i simulacija savijanja proteina te CRO GRID portal.

Optimizacija organizacije transporta rješava problem optimiranja ruta i broja dostavnih vozila na određenom gradskom području uzevši u obzir parametre gradskog prometa. U okviru ove aplikacije ostvarena je suradnja nekoliko tvrtki, a ovdje navodimo samo neke - Hrvatske pošte, Zagrebački električni tramvaj, Tisak, Čistoća i dr. Indeksiranje web/lokalnih mreža je najjednostavnije opisati kao hrvatsku varijantu Google tražilice koja uzima u obzir sve specifičnosti hrvatskog jezika i gramatike. Modeliranje i simulacija savijanja proteina je primjerna znanstvena aplikacija izuzetno složenih matematičkih proračuna u cilju razumijevanja ponašanja i prostorne strukture proteina, s potencijalnom primjenom u farmaceutskoj industriji. CRO GRID portal je namijenjen znanstvenicima koji nisu stručnjaci u području raspodijeljenog računarstva i GRID-a. On im omogućava korištenje različitih GRID aplikacija.

Sve te aplikacije poslužile su kao smjernice za razumijevanje i primjenu novih tehnologija te definiranje načina programiranja raspodijeljenih aplikacija u skoroj budućnosti.

Uz navedene aplikacije u okviru ovog projekta uspostavljena je i suradnja s CERN-om te projektom SEE GRID (*South-Eastern Europe GRID*) koji okuplja zemlje jugoistočne Europe u zasebnu GRID federaciju.

### 3. Programirljivo internetsko okružje kao programski sustav

Programirljivo internetsko okružje u svojstvu programskoga sustava omogućuje brzo i učinkovito oblikovanje raspodijeljenih aplikacija povezivanjem dostupnih usluga. Na primjer, u GRID okolini programirljivo internetsko okružje omogućuje povezivanje usluga postojećih GRID podsustava te gradnju i izvođenje raspodijeljenih aplikacija korištenjem mrežnog kapaciteta i računalne snage ponuđene GRID uslugama. Slično, u okolini telekomunikacijskih operatora programirljivo internetsko okružje može se iskoristiti za gradnju aplikacija koje integriraju usluge različitih podsustava operatora, poput sustava brige o korisnicima (*customer care*), sustava naplate te sigurnosnog podsustava mreže operatora.

Raspodijeljene aplikacije izgrađene putem programirljivog internetskog okružja sastoje se od tri tipa usluga: aplikacijskih usluga, usluga sinkronizacije i komunikacije te poveznih programa. Aplikacijske usluge ostvaruju aplikacijski specifičnu logiku raspodijeljene aplikacije. Usluge sinkronizacije i suradnje omogućuju različite modele komunikacije i sinkronizacije aplikacijskih usluga, npr., povezivanje poštanskim pretincem ili modelom objavi-pretplati (*publish-subscribe*). Povezni programi ostvaruju logiku povezivanja aplikacijskih usluga i usluga sinkronizacije i komunikacije u potpunu raspodijeljenu aplikaciju.

Postupak izgradnje raspodijeljene aplikacije od pojedinačnih usluga putem programirljivog internetskog okružja ostvaruje se u pet koraka: (1) uključivanje računala

u mrežu ravnopravnih sudionika u programirljivom internetskom okružju, (2) pronalaženje i postavljanje aplikacijskih usluga, (3) postavljanje usluga komunikacije i sinkronizacije, (4) pisanje poveznih programa i (5) postavljanje sustava prevođenja i prevođenje poveznih programa.

U prvom koraku korisnik koji želi koristiti programirljivo internetsko okružje uključuje vlastito računalo u mrežu ravnopravnih sudionika u programirljivom internetskom okružju (*Slika 2.a*). To se izvodi dohvatom i postavljanjem programske potpore za programirljivo internetsko okružje na računalo. Prvi korak nije nužan prilikom izgradnje svake aplikacije, već se za svako računalo mora izvesti samo jednom.

U drugom koraku korisnik pristupa izgradnji raspodijeljene aplikacije. Aplikacija se gradi povezivanjem skupa aplikacijskih usluga u slijed izvođenja (*workflow*). Ako aplikacijske usluge potrebne za izvođenje aplikacije već postoje, korisnik ih pronalazi unutar ili izvan mreže programirljivog internetskog okružja. Ako aplikacijske usluge ne postoje, korisnik ih može sam izgraditi i postaviti u mrežu. Na primjer, na *Slici 2.b* korisnik zamišlja aplikaciju sastavljenu od šest usluga. Korisnik pronalazi usluge A, B i C pretraživanjem programirljivog internetskog okružja. Uslugu D pronalazi ostvarenu drugdje na Internetu, a usluge ulaz i izlaz koje predstavljaju ulaz i izlaz nove aplikacije korisnik ostvaruje sam i instalira na vlastito računalo.

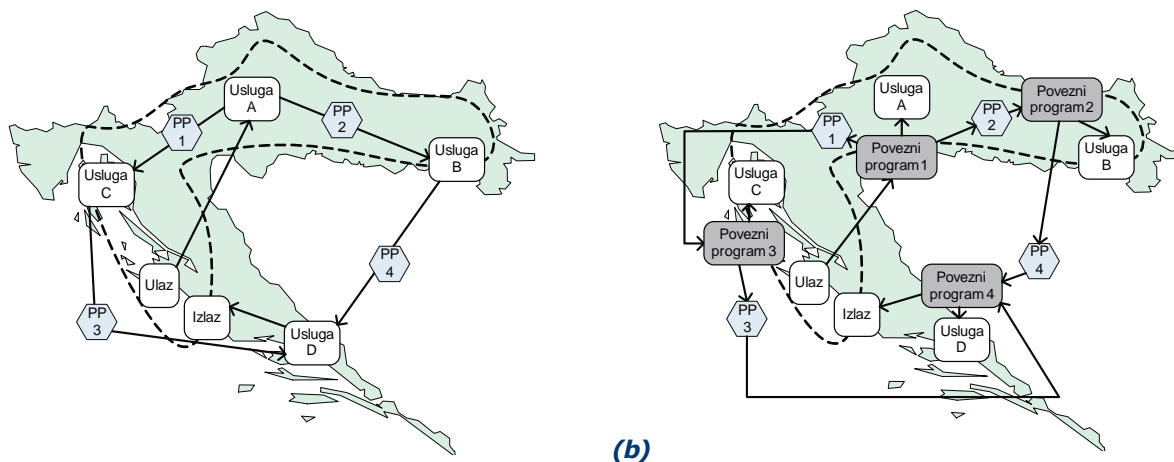
Cilj trećeg koraka je osmisлити načine komunikacije i usklađivanja rada usluga raspodijeljene aplikacije. Programirljivo internetsko okružje podržava četiri načina povezivanja i usklađivanja rada usluga – povezivanje i usklađivanje izravnim pozivima usluga te povezivanje i usklađivanje semaforom, poštanskim pretincem ili usmjernikom događaja. Svaki način povezivanja usluga



(a)

(b)

**Slika 2. Prvi i drugi korak izgradnje raspodijeljene aplikacije zasnovane na programirljivom internetskom okružju**



(a) (b)  
**Slika 3. Treći i četvrti korak izgradnje raspodijeljene aplikacije zasnovane na programirljivom internetskom okružju**

ostvaruje drugačiji model komunikacije između usluga. Dok je izravni poziv usluga podržan standardima za internetske usluge, druga tri komunikacijska modela zahtijevaju postavljenje dodatnih komunikacijskih i sinkronizacijskih usluga. Pošto su te dodatne usluge standardne te se ponavljaju u svim aplikacijama, programirljivo internetsko okružje ih ima unaprijed izgrađene. Dodatno, taj sustav omogućuje napredno upravljanje navedenim uslugama komunikacije i sinkronizacije. Npr., na *Slici 3.a* korisnik je odlučio ostvariti komunikaciju između usluga A te usluga D i izlaz izravnim pozivima usluga, dok se sva ostala komunikacija između usluga ostvaruje poštanskim pretincima.

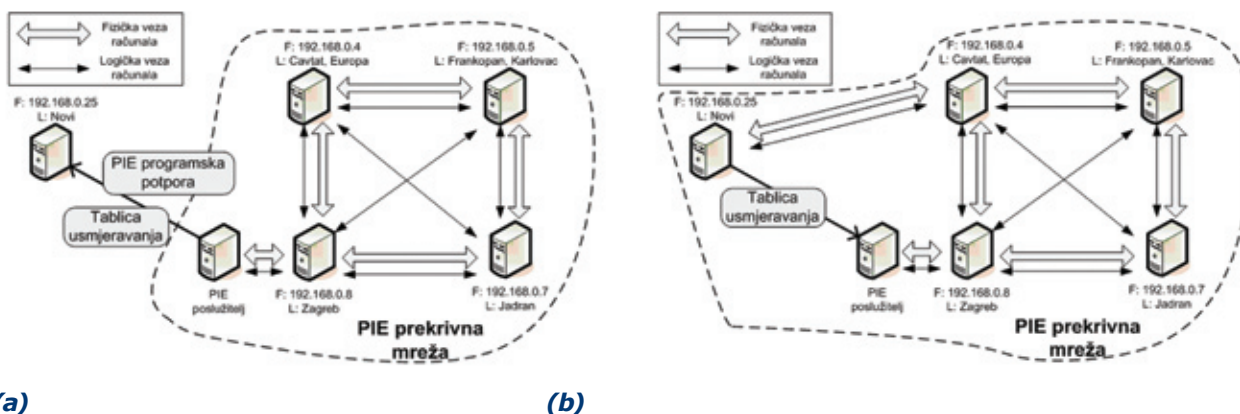
U četvrtom koraku, *Slika 3.b* korisnik stvara povezne programe koji ostvaruju prenošenje vrijednosti između usluga. Prilikom prenošenja vrijednosti povezni programi koriste prethodno postavljene usluge komunikacije i sinkronizacije. Povezni programi su nužni jer su aplikacijske usluge izgrađene neovisno jedne od drugih i od usluga komunikacije i sinkronizacije. Odnosno, aplikacijske usluge nemaju ugrađenu logiku povezivanja. Tipično, povezni programi dohvaćaju ulazne vrijednosti iz usluga sinkronizacije i komunikacije, pozivaju aplikacijske usluge te izlazne vrijednosti prosljeđuju drugoj usluzi sinkronizacije i komunikacije. Npr., logika poveznog programa 1 sa *Slike 3.b* je sljedeća: Nakon primitka izravnog poziv od usluge ulaz, povezni program 1 prosljeđuje ulazne parametre usluzi A, te prima odgovor. Dobiveni odgovor prosljeđuje se u poštanske pretince 1 i 2, od kud ih dalje obrađuju druge usluge i povezni programi. Dodatno, povezni programi mogu ostvarivati i transformiranje izlaznih vrijednosti parametara jedne usluge u ulazne parametre druge usluge. U programirljivom internetskom okružju povezni programi definiraju se korištenjem CL ili SSCL jezika. CL

jezik zasnovan je na XML-u i sličan je WS-BPEL jeziku, standardnom jeziku za definiranje povezivanja usluga. SSCL jezik je jednostavni skriptni jezik prilagođen uporabi u programirljivom internetskom okružju. Pisanje poveznih programa korištenjem SSCL jezika jednostavnije je i brže, dok korištenje CL jezika omogućava definiranje složenijih odnosa usluga. Dodatno, programirljivo internetsko okružje pruža potporu stvaranju SSCL poveznih programa vođenim Web sučeljem.

Posljednji korak prilikom gradnje aplikacije je definiranje sustava prevođenja poveznih programa. Naime, pošto su povezni programi napisani SSCL jezikom visoke razine, prije izvođenja potrebno ih je prevesti u oblik pogodan za izvođenje na računalima programirljivog internetskog okružja. Prevođenje i interpretiranje poveznih programa izvodi poseban podsustav koji se može konfigurirati i time prilagoditi svakoj izgrađenoj aplikaciji. Prevođenje i interpretacija poveznih programa odvija se u stvarnom vremenu i omogućuje razmjerni rast raspodijeljenih aplikacija na velik broj računala. U nastavku poglavlja detaljnije je opisan svaki od navedenih koraka ostvarenja raspodijeljene aplikacije putem programirljivog internetskog okružja.

### 3.1. Uključivanje u prekrivnu mrežu programirljivog internetskog okružja

Osnova upravljanja računalnom infrastrukturom u razvojnoj okolini programirljivog internetskog okružja je sustav prekrivne mreže (*overlay network*). Prekrivna mreža omogućuje komunikaciju računala programirljivog internetskog okružja unutar sigurne komunikacijske okoline logički izdvojene od ostatka javne mreže. Dodatno, prekrivna mreža omogućuje adresiranje računala programirljivog internetskog okružja logičkim adresama.



**Slika 4. Uključivanje novog računala u prekrivnu mrežu programirljivog internetskog okružja**

Uporaba logičkih adresa prekrivne mreže kao osnove komunikacije omogućuje izgradnju prilagodljivog sustava usmjeravanja, dinamičko dodavanje i uklanjanje računala iz sustava te povećanu sigurnost i pouzdanost sustava. Slika 4. prikazuje postupak uključivanja novog računala u prekrivnu mrežu programirljivog internetskog okružja. Slika 4.a prikazuje primjer prekrivne mreže koja se sastoji od četiri računala. Svakom računalo pridijeljena je jedna fizička adresa (F) te jedna ili više logičkih adresa (L). Komunikacija između računala PIE sustava odvija se primjenom logičkih adresa.

Prilikom uključivanja računala u prekrivnu mrežu (Slika 4.a), korisnik računala koje se uključuje pristupa poslužitelju programirljivog internetskog okružja te dohvaća programsku potporu i postavlja je na vlastito računalo. Programska potpora programirljivog internetskog okružja sadrži podsustav za razrješavanje

(*resolving*) logičkih adresa te usmjeravanje poziva usluga u prekrivnoj mreži. Dodatno, dohvaća se i trenutna važeća tablica usmjeravanja prekrivne mreže. Nakon što se programska potpora postavi na računalo (Slika 4.b), računalo mijenja tablicu usmjeravanja prekrivne mreže i prosljeđuje ju poslužitelju programirljivog internetskog okružja. Poslužitelj osvježava tablice usmjeravanja ostalih računala prekrivne mreže podacima o novopriključenom računalo.

Grafičko internetsko sučelje poslužitelja programirljivog internetskog okružja omogućuje jednostavno pregledavanje strukture prekrivne mreže (Slika 5a.), odnosno postojećih računala, logičkih i fizičkih adresa te tablica usmjeravanja. Slika 5.b prikazuje internetsko sučelje za dodavanje novog računala i pridruživanje logičke adrese računalo te uklanjanje stare logičke adrese ili cijelog računala iz prekrivne mreže.

### Virtual Network Management

List of members of Virtual Network

Search node by name:  Search

Search node by physical address:  Search

Logical Node	Physical Address
Marge	damrod.zemnis.fer.hr
Bart	gwaihir.zemnis.fer.hr
Homer	mablung.zemnis.fer.hr
Lisa	161.53.65.187
MrBurns	161.53.65.186
Itchy	161.53.65.222

(a) (b)  
**Slika 5. Internetska sučelja prekrivne mreže: (a) pregled postojećih čvorova i tablice usmjeravanja, (b) dodavanje i uklanjanje računala iz prekrivne mreže**

### Virtual Network Management

#### Adding Local Computer to the Virtual Network

Your computer will become a part of the Virtual Network. To join your computer to the Virtual Network, you have to do the following steps:

- 1) Download the Virtual Network Setup Application
- 2) Run the Virtual Network Setup Application from local computer
- 3) Specify the name of your node within the Virtual Network and root URL of your computer, and then click the **Join** button

Node name:

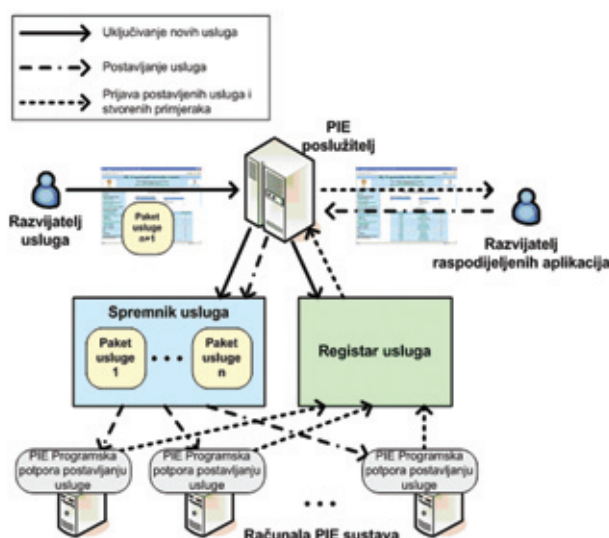
Root URL:

Contacting your computer...  
 Creating logical node on your computer...  
 Exchanging virtual network topology information with your computer...

### 3.2. Upravljanje uslugama programirljivog internetskog okružja

Upravljanje uslugama programirljivog internetskog okružja sastoji se od niza funkcionalnosti koje omogućuju uključivanje novih usluga u mrežu, postavljanje i uklanjanje usluga s računala u programirljivom internetskom okružju i uništavanje primjeraka usluga na računalima, omogućavanje pristupa uslugama izvan programirljivog internetskog okružja te pretraživanje usluga. Slika 6. prikazuje arhitekturu sustava za upravljanje uslugama. Arhitektura se sastoji od spremnika usluga, registra usluga i programske potpore za postavljanje usluga na računala.

Uloga spremnika usluga je pohranjivanje postavljačkih paketa (*installation package*) usluga dostupnih u programirljivom internetskom okružju. Uloga registra usluga je bilježenje lokacija postavljenih usluga, opisa usluga te lokacija postojećih primjerka usluga. Spremnik i registar usluga infrastrukturni su podsustavi programirljivog internetskog okružja te su postavljeni na poslužitelju. Programska potpora za postavljanje usluga koristi upravljačke pakete te na osnovi njih postavlja usluge na računala programirljivog internetskog okružja. Programska potpora za postavljanje usluga dio je programske potpore programirljivog internetskog okružja koja se postavlja na računala prilikom uključivanja računala u okružje.



Slika 6. Arhitektura sustava za upravljanje uslugama

#### 3.2.1. Uključivanje, postavljanje i pokretanje usluga

Novе usluge u programirljivo internetsko okružje uključuju razvijatelji usluga koristeći internetsko sučelje prikazano na Slici 7.a. U internetskom upitniku razvijatelji usluga dostavljaju datoteke izvedivog programskog koda nove usluge, opis sučelja nove usluge WSDL jezikom te skripte za postavljanje nove usluge na računala<sup>1</sup>. Na osnovu dostavljenih datoteka gradi se postavljački paket (*installation package*) nove usluge i pohranjuje u spremnik usluga. Dodatno, registru usluga se dojavljuje postojanje nove usluge koja je spremna za postavljanje. Postavljački paket usluge pohranjuje se u spremnik samo jednom te se naknadno prilikom svakog postavljanja usluge na računala dohvaća iz spremnika i prenosi do računala na kojem se usluga postavlja.

Usluge koje su pohranjene u spremniku usluga moguće je postaviti na računala prekrivne mreže programirljivog internetskog okružja. Postavljanje usluge ostvaruje se pomoću sučelja prikazanog na Slici 7.b. Usluga se postavlja odabirom logičke adrese računala i naziva usluge. Postupak postavljanja u potpunosti je automatiziran te se sastoji od dohvata postavljačkog paketa iz spremnika usluga i slanja paketa programskoj potpori za postavljanje usluga udaljenog računala. Programska potpora za postavljanje usluge na osnovi dostavljenog paketa postavlja i konfigurira uslugu na računalu te novopostavljenu uslugu prijavljuje registru usluga.

#### 3.2.2. Upravljanje uslugama izvan programirljivog internetskog okružja

Programirljivo internetsko okružje omogućuje i upravljanje vanjskim uslugama. Riječ je o uslugama ostvarenim na računalima koja nisu dio programirljivog internetskog okružja, ali su dostupne putem javne mreže. Posljedično, vanjske usluge nije moguće postavljati na računala programirljivog internetskog okružja, već ih se može samo pozivati. Postoji cijeli niz inicijativa za gradnju usluga koje će se moći prijavljivati kao vanjske usluge. Npr., Parlay X skup standarda definira internetske usluge za korištenje usluga telekomunikacijske mreže, Ericssonov istraživački projekt Com2MonSense definira internetske usluge za pristup mrežama senzora, dok Globus GRID inicijativa definira internetske usluge za pristup i upravljanje GRID podsustavima.

<sup>1</sup>Trenutačna izvedba PIE sustava omogućuje postavljanje usluga ostvarenih .NET programskim okvirom te ciljanih za IIS poslužitelje



## Service Repository Management

Storing new service into Service Repository

Node of repository to store your service:

Marge

ZIP file of your service:

WSDL file of your service:

Installation script file of your service:

Name of service to be stored:

(a)

**Slika 7. Internetska sučelja programirljivog internetskog okružja: (a) uključivanje usluge u sustav, (b) postavljanje usluga na računala sustava**

Za prijavu vanjske usluge dovoljno je pristupiti internetskom sučelju prikazanom na *Slici 8.* te dostaviti WSDL opis sučelja vanjske usluge.

Opis sučelja usluge WSDL jezikom sadrži sve podatke potrebne za pristup usluzi. Prijavljena vanjska usluga te njoj pridruženi WSDL opis upisuju se u registar usluga programirljivog internetskog okružja.

### 3.2.3. Pretraživanje usluga

Tijekom gradnje raspodijeljenih aplikacija korisnici programirljivog internetskog okružja unutar sustava traže usluge koje su im potrebne za ostvarivanje aplikacije. Pretraživanje usluga obavlja se putem posebnog internetskog sučelja koje omogućuje uvid u registar usluga programirljivog internetskog okružja. Ako postojeće usluge

## External Service Registration

### Registration of new external service

To register external service, you should provide its WSDL description.

Use the form below to upload the WSDL file of the service you want to register.

You should use an out-of-band method to retrieve the service WSDL. For example, you can visit official web site of the required service and download the service WSDL.

WSDL file:

Optionally, you can use the form below to set the link to the detailed service description.

Link to service description:

**Slika 8. Sučelje za prijavu nove vanjske usluge**

## Service Deployment & Installation

Select the criterium for listing:

All nodes  
Marge  
Bart  
Homer  
Lisa

All services  
MailBox  
BinarySemaphore  
CountingSemaphore  
MailFiller

Logical Node	Service Name	Installed Status
Marge	MailBox	<input checked="" type="checkbox"/>
Marge	BinarySemaphore	<input type="checkbox"/>
Marge	CountingSemaphore	<input type="checkbox"/>
Marge	MailFiller	<input type="checkbox"/>
Marge	ProgramInstaller	<input checked="" type="checkbox"/>
Marge	ConstantProvider	<input type="checkbox"/>
Marge	StockRepresentative	<input type="checkbox"/>
Marge	MailServer	<input type="checkbox"/>
Marge	XMLChecker	<input type="checkbox"/>
Marge	VratiAdrese	<input type="checkbox"/>
Marge	SSCLTranslator	<input checked="" type="checkbox"/>
Marge	ProgramInterpreter	<input checked="" type="checkbox"/>
Marge	ProgramTranslator	<input checked="" type="checkbox"/>
Marge	EventChannel	<input type="checkbox"/>
Marge	MailAdress	<input type="checkbox"/>
Marge	TaskSystemSupervisor	<input checked="" type="checkbox"/>
Marge	GenerateStockClientAdress	<input type="checkbox"/>

(b)

ne zadovoljavaju potrebe korisnika, oni putem prethodno opisanih sučelja mogu razviti novu uslugu te je uključiti i postaviti u programirljivo internetsko okružje, ili prijaviti vanjsku uslugu.

### 3.3. Upravljanje komunikacijskim i sinkronizacijskim uslugama

Razvojna okolina programirljivog internetskog okružja sadrži skup općenitih komunikacijskih i sinkronizacijskih usluga. Komunikacijske i sinkronizacijske usluge služe za ostvarivanje različitih modela usklađivanja i komunikacije između usluga raspodijeljene aplikacije. Trenutačna inačica programirljivog internetskog okružja ostvaruje tri mehanizma sinkronizacije i komunikacije: poštanski pretinac, usmjernik događaja i semafor. Poštanski pretinac omogućuje asinkronu razmjenu poruka između usluga. Usmjernik događaja omogućuje komunikaciju usluga na osnovi objavi-pretplati mehanizma (*publish-subscribe*). Semafor omogućuje usklađivanje izvođenja više usluga.

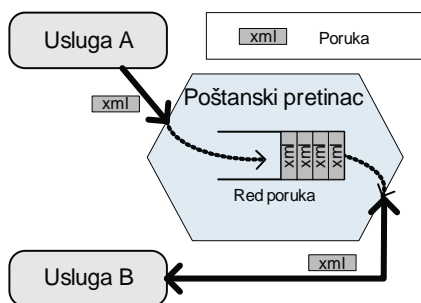
Sve ostvarene usluge komunikacije i sinkronizacije omogućuju stvaranje i uništavanje više primjeraka usluge na jednom računalu (*service instance*). Pojedine primjerke usluga sinkronizacije i komunikacije moguće je koristiti istovremeno i međusobno neovisno u različitim aplikacijama.

Usluge sinkronizacije i komunikacije ostvarene su primjenom osnovnih standarda za internetske usluge te standarda *WS-ResourceFramework* i *WS-*

*Addressing.* Ostvarenje tih usluga kao unutrašnjih usluga programirljivog internetskog okruženja omogućava njihovo postavljanje na sva računala u sustavu.

### 3.3.1. Poštanski pretinac

Poštanski pretinac je standardna usluga programirljivog internetskog okruženja koja omogućuje razmjenu poruka između povezanih programa i usluga aplikacije. Komunikacija primjenom poštanskog pretinca vremenski je nepovezana jer poštanski pretinac pohranjuje poslano poruke sve dok ih primatelj ne preuzme. Tom uslugom razmjenjuju se poruke pisane XML jezikom. Prednosti korištenja poruka u XML obliku su mogućnost prosljeđivanja podatkovnih struktura proizvoljne složenosti te neovisnost o sklopovskoj, programskoj i razvojnoj platformi usluga koje razmjenjuju poruke. Prilikom prosljeđivanja poruka poštanski pretinac ne analizira



**Slika 9. Primjena poštanskog pretinca u komunikaciji usluga**

sadržaj poruka koje prenosi, već poruke samo prosljeđuje.

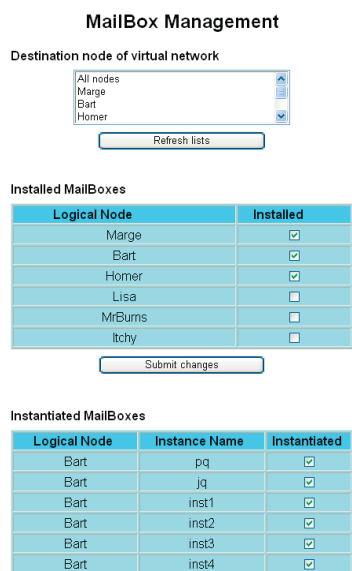
Slika 9. prikazuje primjer razmjene poruka korištenjem poštanskog pretinca. Usluga A prosljeđuje XML poruke u poštanski pretinac, a usluga B dohvaća poruke iz poštanskog pretinca. Ako u trenutku pristupa usluge B poštanskom pretincu, ne postoji poruka u redu poruka poštanskog pretinca usluga B ulazi u stanje čekanja. Kada se na poštanskom pretincu pojavi poruka, poštanski pretinac automatski je prosljeđuje usluzi B koja nastavlja s radom.

Programirljivo internetsko okruženje ostvaruje grafičko internetsko sučelje za upravljanje poštanskim pretincima (Slika 10.). Sučelje za upravljanje poštanskim pretincima omogućuje postavljanje i uklanjanje usluge poštanskog

pretinca s pojedinih računala programirljivog internetskog okruženja te stvaranje i uništavanje primjeraka usluge na pojedinim računalima mreže. Time je graditelju raspodijeljene aplikacije omogućeno postavljanje i pokretanje vlastitih primjeraka usluge poštanski pretinac na računalima u mreži.

### 3.3.2. Usmjernik događaja

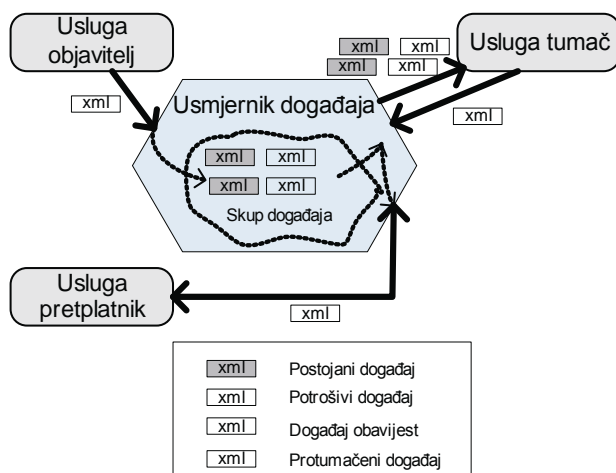
Usmjernik događaja najsloženija je usluga sinkronizacije i komunikacije ostvarena trenutačnom inačicom programirljivog internetskog okruženja. Usmjernik događaja ostvaruje napredni oblik komunikacijskog modela objavi-pretplati. Napredni model komunikacijskog modela objavi-pretplati zasniva se na XML dokumentima koji predstavljaju događaje te uslugama pretplatnicima, objaviteljima i tumačima. Usluge objavitelji objavljuju



**Slika 10. Sučelje za upravljanje poštanskim pretincima**

nove događaje u usmjernik događaja. Usluge pretplatnici pretplaćuju se na događaje usmjerniku događaja te odabiru uslugu koju koriste za tumačenje događaja. Uloga usluge tumača je da na osnovi trenutačno objavljenih događaja utvrdi je li potrebno obavijestiti uslugu pretplatnika. Ako je uslugu pretplatnika potrebno obavijestiti, usluga tumač stvara protumačeni događaj koji se prosljeđuje usluzi pretplatniku.

Usmjernik događaja različito postupa s različitim tipovima događaja. Usmjernik događaja raspoznaje četiri tipa događaja: postojani događaji, potrošivi događaji, događaji obavijesti te protumačeni događaji (Slika 11.). Postojane događaje, potrošive događaje i događaje obavijesti objavljuju usluge objavitelji. Nakon objave, postojani



**Slika 11. Primjena usmjernika događaja u komunikaciji usluga**

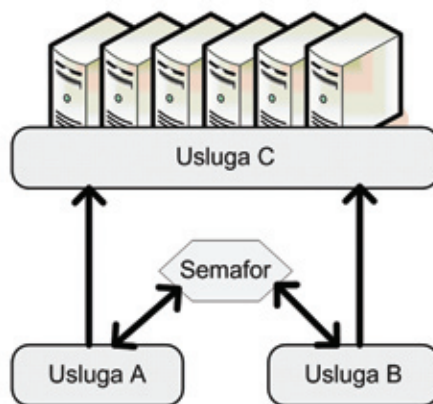
događaj se umeće u skup događaja usmjernika i ostaje u skupu sve dok ih objavitelj događaja ne povuče. Potrošivi dokumenti umeću se u skup objavljenih događaja, ali se iz skupa automatski uklanjaju kada prva usluga tumač potvrdi ispunjenje uvjeta pretplate. Događaji obavijesti jednokratno se upisuju u skup događaja i šalju svim uslugama tumačima. Protumačeni događaji su događaji koje stvaraju usluge tumači na osnovu skupa događaja usmjernika kada su ispunjeni uvjeti pretplate.

Nakon svake objave događaja (postojanog, potrošivog ili obavijesti) usmjernik poziva sve pridružene usluge tumače i prosljeđuje im skup događaja koji trenutačno postoje u usmjerniku. Usluge tumači potom odlučuju ispunjava li trenutačni skup događaja uvjete pretplate. Ako neki tumač događaja utvrdi da su uvjeti pretplate ispunjeni, on stvara protumačeni događaj te ga vraća usmjerniku. Primljeni protumačeni događaj usmjernik prosljeđuje onoj usluzi pretplatnik čiji tumač je stvorio protumačeni događaj.

Usmjernicima događaja upravlja se grafičkim internetskim sučeljem. Grafičko sučelje omogućuje postavljanje i uklanjanje usmjernika događaja s računala programirljivog internetskog okružja te pokretanje i gašenje primjeraka usluge usmjernik događaja na tim računalima. Grafičko internetsko sučelje za upravljanje usmjernicima događaja istovjetno je sučelju za upravljanje poštanskim pretincima.

### 3.3.3. Binarni i opći semafor

Treća vrsta sinkronizacijske i komunikacijske usluge su semafori. Semafori su primarno namijenjeni sinkronizaciji izvođenja usluga i povezanih programa. Ostvarena su dva tipa semafora: binarni i opći. Binarni semafor omogućuje međusobno isključivanje povezanih programa i usluga jer u svakom trenutku dozvoljava najviše jednoj usluzi ili poveznom programu da ga zauzme. Nakon zauzimanja,



**Slika 12. Primjena binarnog semafora u usklađivanju rada usluga**

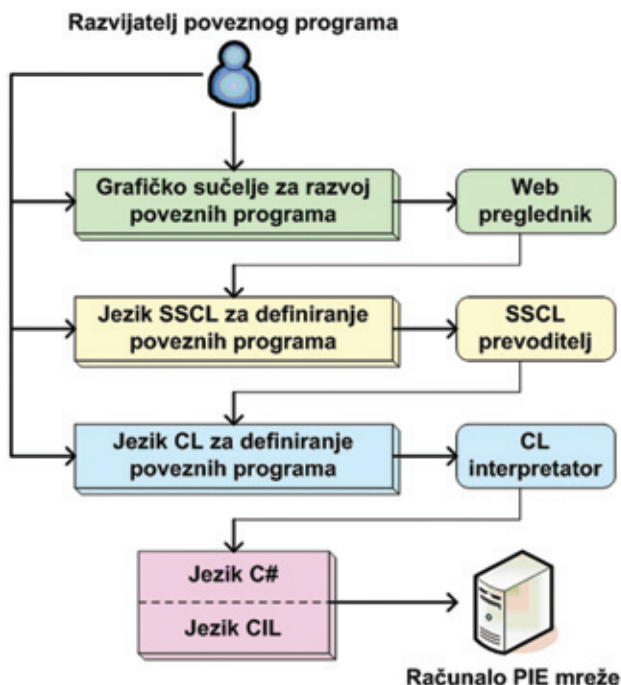
usluga ili povezni program mora osloboditi binarni semafor kako bi ga druga usluga ili povezni program mogao zauzeti. Opći semafor je proširenje binarnog semafora i namijenjen je zaštiti sredstava (*resources*). U odnosu na binarni semafor, više poveznih programa ili usluga može istovremeno zauzeti opći semafor. Maksimalni broj usluga koje mogu istovremeno zauzeti opći semafor zadaje se prilikom stvaranja primjerka općeg semafora.

Najčešća primjena semafora u razvojnoj okolini PIE je ograničavanje istovremenog pristupa računalnom sredstvu na definirani broj raspodijeljenih programa. Slika 12. prikazuje primjer korištenja binarnog semafora za ograničavanje istovremenog pristupa usluzi C koja ostvaruje pristup i korištenje skupa računala. Prije pristupa usluzi C, usluga A pristupa semaforu i traži dozvolu za pristup usluzi C. Pošto semafor nije zauzet, usluzi A dozvoljava se pristup usluzi C te usluga A pristupa usluzi C. Kada usluga B zatraži pristup semaforu, pristup se odbija, jer je sluga A već zauzela semafor. Nakon završetka korištenja usluge C, usluga A oslobađa semafor. Nakon što je oslobođen, semafor javlja usluzi B da je pristup sredstvu dozvoljen te usluga B može koristiti uslugu C.

Binarnim i općim semaforima upravlja se grafičkim internetskim sučeljem istovjetnim sučelju za upravljanje poštanskim pretincem. To sučelje omogućuje postavljanje i uklanjanje usluge semafora s računala programirljivog internetskog okružja te pokretanje i gašenje primjeraka semafora na tim računalima.

### 3.4. Gradnja povezanih programa

Povezni programi povezuju unutrašnje i vanjske usluge programirljivog internetskog okružja te usluge sinkronizacije i komunikacije u jedinstvenu raspodijeljenu



**Slika 13. Hijerarhija razvojnih jezika u razvojnoj okolini PIE**

aplikaciju. Povezne programe moguće je ostvariti pomoću nekoliko hijerarhijski povezanih jezika. Slika 13. prikazuje hijerarhiju jezika dostupnih u programirljivom internetskom okruženju. Na vrhu hijerarhije nalazi se grafičko internetsko sučelje za razvoj poveznih programa. Grafičko sučelje omogućuje jednostavan razvoj poveznih programa korištenjem internetskoga preglednika. Razvoj poveznih programa grafičkim sučeljem je brz i jednostavan, ali je i mogućnost gradnje složenih poveznih programa mala.

Grafičko sučelje za razvoj poveznih programa stvara programski kôd u jeziku SSCL (*Simple Service Composition Language*). Jezik SSCL jednostavan je skriptni jezik prilagođen razvoju poveznih programa namijenjenih povezivanju aplikacijskih usluga korištenjem usluga komunikacije i sinkronizacije. Osim automatskog stvaranja SSCL naredbi korištenjem grafičkog internetskoga sučelja, iskusniji programeri mogu jezik SSCL koristiti izravno te tako ostvariti bolji nadzor nad strukturom poveznog programa.

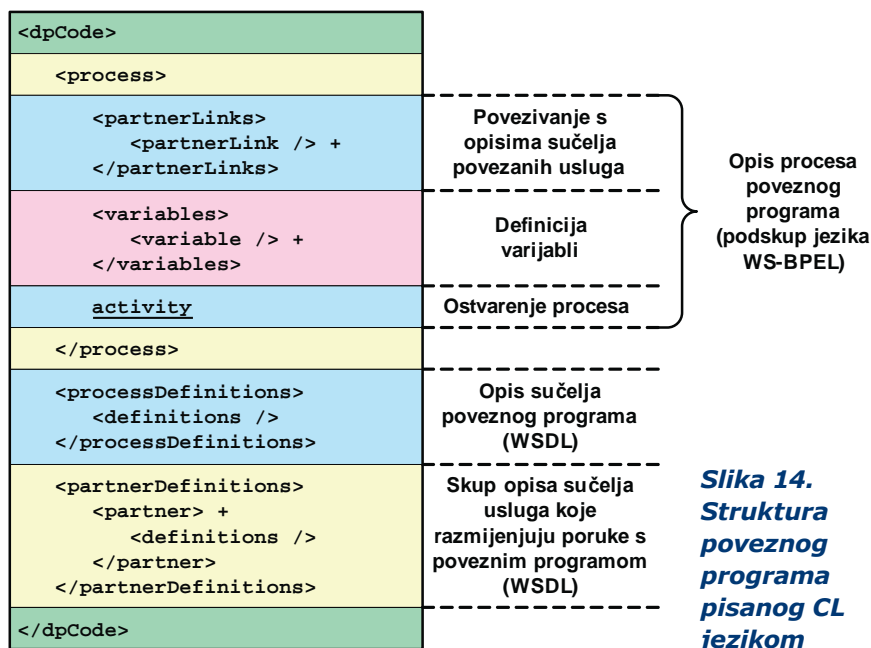
Jezik SSCL prevodi se u jezik CL (*Coopetition Language*)

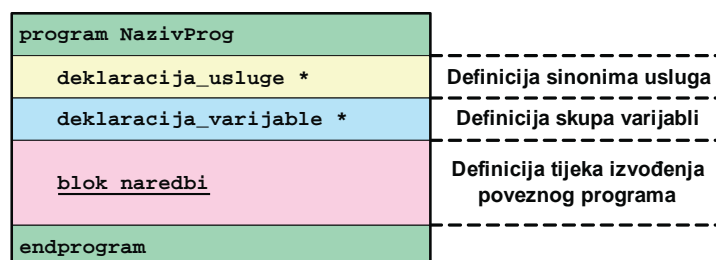
korištenjem SSCL prevoditelja. Osnovne prednosti jezika CL su sukladnost sa standardima za internetske usluge te mogućnost jednostavnog prenošenja i izvođenja na svim računalima programirljivog internetskoga okruženja. Jezik CL definira upravljanje pozivima usluga na razini SOAP poruka te upravljanje podacima na razini XML podatkovnih struktura. Navedena svojstva omogućuju potpunu kontrolu nad izvođenjem poveznih programa. Međutim, zbog opširnosti i složenosti XML strukture jezika, jezik CL nije prikladan za izravnu uporabu. Jezik CL stoga se najčešće koristi kao izlaz SSCL prevoditelja, a u rijetkim primjenama omogućuje i manje zahvate iskusnim programerima.

U praktičnoj izvedbi programirljivog internetskoga okruženja pomoću .NET radnog okvira jezik CL prevodi se dvostupanjskim procesom CL prevoditelja u jezik CIL. U prvom stupnju procesa prevodenja, jezik CL prevodi se u jezik C#, jedan od osnovnih jezika razvoja usluga u .NET radnom okviru (*.NET framework*). U drugom stupnju prevodenja, jezik C# prevodi se u jezik CIL (engl. *Common Intermediate Language*) primjenom ugrađenog prevoditelja .NET radnog okvira. Programi pisani jezikom CIL izravno se izvode pomoću virtualnog stroja .NET okvira.

### 3.4.1. Jezik CL

Jezikom CL definiraju se povezni programi na razini upravljanja SOAP porukama koje prolaze kroz povezni program. SOAP poruke slijede XML sintaksu i predstavljaju osnovnu jedinicu komunikacije između internetskih usluga. Jezikom CL određuje se što povezni program mora učiniti nakon primitka određene SOAP poruke, odnosno, koje transformacije mora učiniti nad porukom te kamo poruku mora proslijediti.





Slika 15. Struktura programa pisanih u jeziku SSCL

Struktura poveznog programa pisanog CL jezikom prikazana je na Slici 14. CL jezik zasnovan je na XML strukturi i sastoji se od opisa procesa poveznog programa, opisa sučelja poveznog programa te skupa opisa sučelja usluga s kojima je povezni program razmjenjuje SOAP poruke.

Opis sučelja poveznog programa i opisi sučelja svih usluga s kojima povezni program razmjenjuje SOAP poruke izvedeni su standardnim jezikom za opisivanje sučelja za internetske usluge – WSDL jezikom. Opis procesa izveden je iz jezika WS-BPEL, ali je podržan samo sažeti podskup svih funkcionalnosti jezika WS-BPEL. Opis procesa sadrži definicije svih poruka koje se razmjenjuju između poveznog programa i drugih usluga. Dodatno, definira se skup lokalnih varijabli koje mogu pohranjivati međuvrijednosti parametara poziva prilikom pozivanja više usluga. Naposljetku, definira se niz aktivnosti koje određuju proces. Tipičan slijed aktivnosti je prihvaćanje SOAP poruke, prosljeđivanje poruke drugoj usluzi (pozivanje druge usluge), dohvaćanje povratne vrijednosti te njeno prosljeđivanje trećoj usluzi.

### 3.4.2. Jezik SSCL

Jezik SSCL razvijen je uslijed složenosti jezika CL. Struktura jezika SSCL nije zasnovana na XML-u, već je slična skriptnim jezicima. Jezične konstrukcije jezika SSCL oblikovane su prema skupu jezičnih konstrukcija jezika WS-BPEL, no znatno su jednostavnije sintakse. Jezik SSCL od korisnika skriva složenost upravljanja XML podacima, definiranja veza s WSDL sučeljima te upravljanja SOAP porukama. Dodatno, jezik SSCL sadrži posebno prilagođene jezične konstrukcije koje dodatno olakšavaju definiranje međudjelovanja poveznog programa s uslugama sinkronizacije i komunikacije. Zbog toga se primjenom jezika SSCL grade povezni programi čiji je kôd znatno kraći, jednostavniji i razumljiviji čovjeku.

Na Slici 15. prikazana je struktura programa pisanih jezikom SSCL. Tijelo programa podijeljeno je u tri osnovna dijela. Prvi dio čine deklaracije usluga koje se pozivaju iz programa. Deklaracijama usluga definiraju se pseudonimi (*alias*) kojima će se pozivati usluge u poveznom programu. Deklaracije usluga koriste imena usluga koja su navedena

u registru usluga programirljivog internetskog okružja, čime se izbjegava složeni unos WSDL opisa sučelja usluge. Drugi dio tijela programa čini deklaracija varijabli u kojoj se navodi skup varijabli koje povezni program koristi. Posljednji dio čini blok naredbi za pozivanje usluga i upravljanje tijekom izvođenja poveznog programa.

Slika 16. ilustrira razliku u definiranju poveznih programa jezicima SSCL i CL na primjeru prevođenja jedne naredbe jezika SSCL u istoznačnu složenu XML strukturu jezika CL.

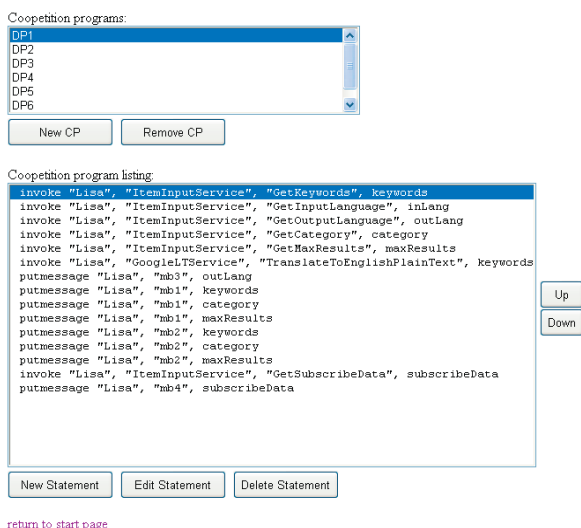
### 3.4.3. Grafičko sučelje za razvoj poveznih programa

Grafičko sučelje vodi korisnika kroz razvoj poveznih programa ostvarujući poluautomatsko uređivanje programskih datoteka poveznih programa. Poluautomatski postupak korisniku pomaže tijekom izgradnje programa nudeći izbor jezičnih konstrukta jezika SSCL koje korisnik može upotrijebiti u programu. Nakon što korisnik odabere jezični konstrukt, grafičko sučelje automatski gradi sintaksnu strukturu, a korisnik samo unosi vrijednosti

Izvedba poziva usluge jezikom SSCL
invoke serviceURL, operation, inputVar, outputVar
Istoznačna izvedba poziva usluge jezikom CL
<pre> &lt;assign&gt;   &lt;copy&gt;     &lt;from&gt;       ...     &lt;/from&gt;     &lt;to partnerLink="ServicePL" /&gt;   &lt;/copy&gt; &lt;/assign&gt; &lt;assign&gt;   &lt;copy&gt;     &lt;from&gt;       ...     &lt;/from&gt;     &lt;to variable="srcvRqstMsg" part="params" /&gt;   &lt;/copy&gt; &lt;/assign&gt; &lt;invoke partnerLink="ServicePL" portType="..."   operation="..." inputVariable="srcvRqstMsg"   outputVariable="srcvRsltMsg" /&gt; </pre>

Slika 16. Primjer SSCL naredbe invoke i istoznačnog CL kôda

Application Name: BookSearch



[return to start page](#)

**Slika 17. Sučelje za razvoj povezih programa**

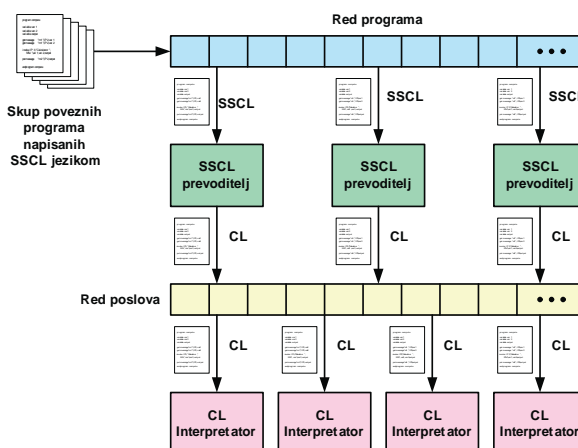
pojedinih parametara. Dodatno, sučelje za razvoj programa objedinjeno je s podacima iz registra usluga te prilikom definiranja parametara jezičnih konstrukta predlaže moguće vrijednosti imena usluga i primjeraka usluga.

Raspodijeljena aplikacija sastoji se od skupa povezih programa koji se grade neovisno jedan o drugome. Izgradnja povezih programa zasnovana je na sučelju prikazanom na *Slici 17*. Sučelje omogućuje uređivanje poveznog programa dodavanjem novih naredbi te izmjenama i brisanjem postojećih naredbi. Izgrađeno grafičko sučelje razvojne okoline programirljivog internetskog okružja omogućuje jednostavan i brz razvoj povezih programa i raspodijeljenih aplikacija.

### 3.5. Prevođenje povezih programa

Raspodijeljena aplikacija sastoji se od skupa aplikacijskih usluga, skupa usluga sinkronizacije i komunikacije te skupa povezih programa. Aplikacijske usluge i usluge sinkronizacije i komunikacije postavljaju se na računala primjenom sustava za upravljanje uslugama. Povezni programi ostvaruju tijek izvođenja aplikacije pozivanjem aplikacijskih usluga te međusobnom komunikacijom i sinkronizacijom. Izvođenje povezih programa, a time i raspodijeljene aplikacije, u programirljivom internetskom okružju zasniva se na dvoslojnoj arhitekturi prevođenja, raspoređivanja i izvođenja prikazanoj *Slikom 18*.

Dvoslojna arhitektura prevođenja, raspoređivanja i izvođenja sastoji se od reda programa, reda poslova, prevoditelja jezika SSCL i prevoditelja jezika CL. Redovi programa i poslova ostvareni su korištenjem usluge poštanski pretinac. Nakon pokretanja aplikacije, svi povezni programi koji čine aplikaciju umeću se u red



**Slika 18. Dvoslojna arhitektura prevođenja, raspoređivanja i izvođenja**

programa. Prevoditelji jezika SSCL dohvaćaju povezne programe napisane SSCL jezikom iz reda programa, prevode ih u povezne programe napisane CL jezikom te postavljaju u red poslova. Svaki SSCL prevoditelj radi u ciklusu tijekom kojeg dohvaća jedan SSCL program iz reda programa, prevodi ga u jezik CL i dodaje u red poslova. Prevoditelji SSCL ponavljaju opisani ciklus sve dok ih korisnik ne zaustavi.

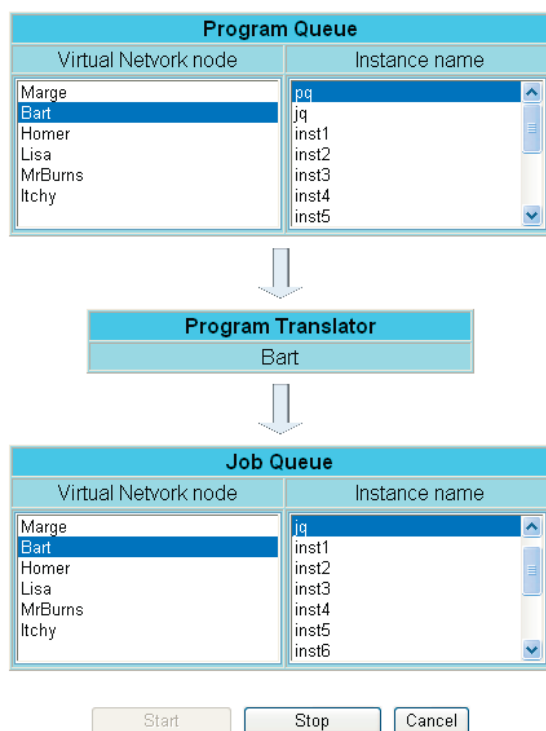
U redu poslova nalaze se povezni programi pisani CL jezikom. CL prevoditelji postavljeni su na računala programirljivog internetskog okružja. Svaki prevoditelj povezuje se s redom poslova, dohvaća jedan CL povezni program i izvodi ga na računalu na koje je postavljen. Opisani ciklus dohvaćanja i izvođenja ponavlja se sve dok korisnik ne zaustavi izvođenje prevoditelja.

#### 3.5.1. Upravljanje postupkom prevođenja

Upravljanje dvoslojnom arhitekturom za prevođenje, raspoređivanje i izvođenje povezih programa u razvojnoj okolini programirljivog internetskog okružja podijeljeno je na upravljanje SSCL prevoditeljima i upravljanje CL prevoditeljima. Upravljanje SSCL prevoditeljima omogućuje postavljanje i uklanjanje SSCL prevoditelja s računala PIE mreže te upravljanje načinom povezivanja SSCL prevoditelja s redovima programa i poslova. Upravljanje CL prevoditeljima omogućuje postavljanje i uklanjanje CL prevoditelja s računala programirljivog internetskog okružja te definiranje reda poslova iz kojeg pojedini prevoditelj dohvaća povezne programe.

*Slika 19.a* prikazuje sučelje za upravljanje povezivanjem SSCL prevoditelja s redovima programa i poslova, dok

## Program Translator Management



(a)

**Slika 19. Definiranje prevođenja povezanih programa: (a) povezivanje prevoditelja s redom programa i redom poslova, (b) povezivanje prevoditelja s redom poslova**

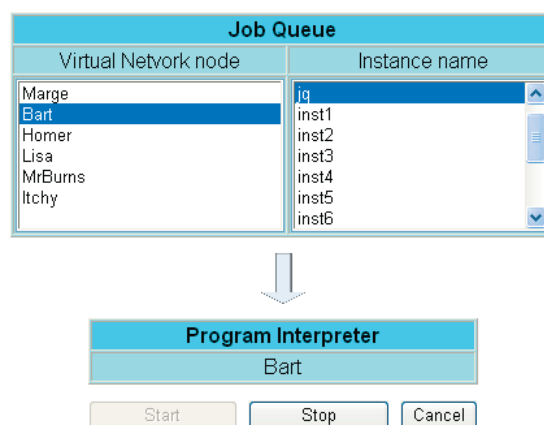
Slika 19.b prikazuje sučelje za definiranje povezivanja CL prevoditelja s redom poslova. Za svaki SSCL prevoditelj određuje se poštanski pretinac s kojega prevoditelj dohvaća SSCL programe te poštanski pretinac na koji prosljeđuje CL programe. Za svaki CL prevoditelj definira se poštanski pretinac s kojeg prevoditelj dohvaća CL programe koje će izvoditi.

Zahvaljujući primjeni redova i mogućnosti dinamičkog konfiguriranja SSCL prevoditelja i CL prevoditelja, izgrađena arhitektura prevođenja, raspoređivanja i izvođenja omogućuje razmjerni rast raspodijeljene aplikacije i visok stupanj prilagodljivosti potrebama korisnika.

## 4. Zaključak

U ovom članku predstavlja se sustav programirljivog internetskog okružja (*Programmable Internet Environment*). On je namijenjen povezivanju javno dostupnih usluga međusobno udaljenih računala u složenu raspodijeljenu aplikaciju. Usluge mogu nuditi različite funkcionalnosti, od korištenja različitih resursa računala i mreža, do pristupa znanju institucija. Sustav programirljivog internetskog okružja razvijen je kao dio projekta CRO-GRID posrednik, u okviru kojeg je

## Program Interpreter Management



(b)

predstavljen na nizu konferencija i radionica te elitnim američkim sveučilištima i tvrtkama. Cjelokupnim poliprojektom CRO-GRID stvorena je hrvatska nacionalna GRID mreža koja je uključena u istovjetne europske GRID sustave. Razvijene su prototipne aplikacije te su pokazane mogućnosti njihove komercijalne primjene.

Pomoću programirljivog internetskog okružja postavlja se mreža ravnopravnih sudionika koja povezuje udaljene institucije te stvara sigurnu i odvojenu okolinu za izvođenje raspodijeljene aplikacije. Nadalje, programirljivim internetskim okružjem postavljaju se posebne usluge sinkronizacije i komunikacije koje omogućuju različite modele usklađivanja i razmjena poruka između usluga smještenih u udaljenim institucijama. Dodatno, programirljivim internetskim okružjem grade se povezni programi koji povezuju usluge udaljenih institucija pomoću usluga sinkronizacije i komunikacije u jedinstvenu smislenu aplikaciju. Upravljanje svim funkcionalnostima programirljivog internetskog okružja te stvaranje novih raspodijeljenih aplikacija ostvaruje se korištenjem internetskoga preglednika i pristupanjem poslužitelju programirljivog internetskog okružja. Korištenje jednostavnog internetskog sučelja omogućuje jednostavno i korisniku prilagođeno nadziranje sustava i gradnju raspodijeljenih aplikacija.

## 5. Kratice

<b>SOC</b>	Service Oriented Computing
<b>IP</b>	Internet Protocol
<b>XML</b>	Extensible Markup Language
<b>WSDL</b>	Web Service Description Language
<b>SOAP</b>	Simple Object Access Protocol
<b>UDDI</b>	Universal Description, Discovery and Integration
<b>WS-BPEL</b>	Web Services – Business Process Execution Language
<b>WS-Addressing</b>	Web Services – Addressing
<b>WS-RF</b>	Web Services – Resource Frame- work
<b>PIE</b>	Programmable Internet Environment
<b>CL</b>	Composition Language
<b>SSCL</b>	Simple Service Composition Language
<b>HIT</b>	Hrvatski institut za tehnologiju
<b>HITRA</b>	Hrvatski inovacijski tehnološki razvitak

## 6. Literatura

[1] D. Škvorc: „Prividna mreža računalnih sustava zasnovanih na uslugama“, Magistarski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, listopad 2005.

[2] M. Podravec: „Otkrivanje i postavljanje usluga u sustavima zasnovanim na uslugama“, Magistarski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2005.

[3] A. Milanović, S. Srbljić, D. Skrobo, D. Čapalija, S. Rešković: „Coopetition Mechanisms for Service-Oriented Distributed Systems“, Proceedings of the CCCT 2005 (The 3rd International Conference on Computing, Communications and Control Technologies), Austin, Texas, USA, July 2005, Vol. 1, pp. 118-123.

[4] S. Rešković: „Sinkronizacijski i komunikacijski mehanizmi za sustave zasnovane na uslugama“, Diplomski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, rujan 2005.

[5] D. Čapalija: „Objava/pretplata mehanizmi za ostvarivanje mreža zasnovanih na sadržaju“, Diplomski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, svibanj

2005.

[6] D. Skrobo: „Raspodijeljeno usporedno interpretiranje programa u arhitekturama zasnovanim na uslugama“, Magistarski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, listopad 2005.

[7] I. Gavran: „Korisnički jezik programskog modela zasnovanog na uslugama“, Magistarski rad u izradi, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2005.

[8] D. Skrobo, A. Milanović, S. Srbljić: “Distributed Program Interpretation in Service-Oriented Architectures”, Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2005), Orlando, Florida, USA, July 2005, Vol. 4, pp. 193-197.

[9] A. Milanović: „Programski model zasnovan na uslugama“, doktorska disertacija, FER, Sveučilište u Zagrebu, 2006.

[10] World Wide Web consortium: „Extensible Markup Language (XML) 1.0“, <http://www.w3.org/TR/REC-xml/>

[11] World Wide Web consortium: „SOAP Version 1.2“, <http://www.w3.org/TR/soap/>

[12] World Wide Web consortium: „Web Services Description Language (WSDL) 1.1“, <http://www.w3.org/TR/wsdl>

[13] OASIS: „UDDI Spec Technical Committee Draft“, [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)

[14] OASIS: „Business Process Execution Language for Web Services“, <http://www.oasis-open.org/committees/wsbpel/charter.php>

[15] World Wide Web consortium: „Web Services Addressing (WS-Addressing)“, <http://www.w3.org/Submission/ws-addressing/>

[16] OASIS: „Web Services Resource 1.2 (WS-Resource)“, [http://docs.oasis-open.org/wsrif/wsrif-ws\\_resource-1.2-spec-os.pdf](http://docs.oasis-open.org/wsrif/wsrif-ws_resource-1.2-spec-os.pdf)

[17] The Parlay Group: „Parlay X specifications“, <http://www.parlay.org/en/specifications/pxws.asp>



**Adrese autora:**

**Siniša Sribljčić**

**e-mail: [sinisa.sribljic@fer.hr](mailto:sinisa.sribljic@fer.hr)**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu

Unska 3

HR-10000 Zagreb

Hrvatska

**Andro Milanović**

**e-mail: [andro.milanovic@combis.hr](mailto:andro.milanovic@combis.hr)**

Combis d.o.o.

Baštijanova 52a

HR-10000 Zagreb

Hrvatska

**Dejan Škvorc**

**e-mail: [dejan.skvorc@fer.hr](mailto:dejan.skvorc@fer.hr)**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu

Unska 3

HR-10000 Zagreb

Hrvatska

**Daniel Skrobo**

**e-mail: [daniel.skrobo@fer.hr](mailto:daniel.skrobo@fer.hr)**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu

Unska 3

HR-10000 Zagreb

Hrvatska

**Miroslav Popović**

**e-mail: [miroslav.popovic@fer.hr](mailto:miroslav.popovic@fer.hr)**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu

Unska 3

HR-10000 Zagreb

Hrvatska

**Ivan Gavran**

**e-mail: [ivan.gavran@fer.hr](mailto:ivan.gavran@fer.hr)**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu

Unska 3

HR-10000 Zagreb

Hrvatska

**Matija Podravec**

**e-mail: [matija.podravec@fer.hr](mailto:matija.podravec@fer.hr)**

Hrvatske telekomunikacije d.d.

Savska cesta 32

HR-10000 Zagreb

Hrvatska

**Ivan Žužak**

**e-mail: [ivan.zuzak@fer.hr](mailto:ivan.zuzak@fer.hr)**

Fakultet elektrotehnike i računarstva,  
Sveučilište u Zagrebu

Unska 3

HR-10000 Zagreb

Hrvatska

**Ivan Skuliber**

**e-mail: [ivan.skuliber@ericsson.com](mailto:ivan.skuliber@ericsson.com)**

Ericsson Nikola Tesla d.d.

Krapinska 45

p.p. 93

HR-10002 Zagreb

Hrvatska

**Ivan Benc**

**e-mail: [ivan.benc@ericsson.com](mailto:ivan.benc@ericsson.com)**

Ericsson Nikola Tesla d.d.

Krapinska 45

p.p. 93

HR-10002 Zagreb

Hrvatska

*Uredništvo je primilo rukopis 3. svibnja 2007.*