



Krunoslav Tržec



Saša Dešić



Gordan Ježić



Mario Kušek

**Krunoslav Tržec, Saša Dešić**

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

**Gordan Ježić, Mario Kušek**

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

**Ključne riječi:**

Agentske tehnologije  
Pokretni agenti  
Višeagentski sustav  
Upravljanje programskom  
podrškom na daljinu  
Grid okolina

**Key words:**

Agent-oriented technologies  
Mobile agents  
Multiagent system  
Remote software  
management  
Grid environment

# Daljinsko upravljanje programskom podrškom pomoću agenata

*Sažetak*

U ovom članku opisana je primjena agentskih tehnologija u području upravljanja programskom podrškom na daljinu u informacijsko-komunikacijskim sustavima. Posebna pažnja posvećena je tehnologiji pokretnih agenata koja je korištena prilikom implementacije višeagentskoga sustava za upravljanje programskom podrškom na daljinu kako bi se uspješno udovoljilo zahtjevima za fleksibilnim i automatiziranim održavanjem programske podrške u novoj generaciji mreža. Opisan je primjer automatiziranoga upravljanja programskom podrškom u *Grid* okolini za koju je karakteristično da su korisnici, programske komponente i računalni resursi raspodijeljeni na velikom broju mrežnih čvorova koji se nalaze u heterogenim i zemljopisno raspršenim telekomunikacijskim mrežama. Višeagentski sustav za upravljanje programskom podrškom na daljinu je nastao kao rezultat istraživačkoga projekta ROPE (*Remote Operation Environment*) kojega je Istraživački odjel Instituta za telekomunikacije u Ericssonu Nikoli Tesli vodio u suradnji sa Zavodom za telekomunikacije Fakulteta elektrotehnike i računarstva u Zagrebu.

## AGENT-BASED REMOTE SOFTWARE MANAGEMENT

*Abstract*

This article deals with usage of agent-oriented technologies in the area of remote software management in information communications systems. Special attention has been paid to mobile agent technology that was applied in implementation of multi-agent system for remote software management. The example of such an automated remote software management is given for the Grid environment that is characterized as an open system in which users, software components and computational resources are distributed across the heterogeneous and geographically dispersed telecommunications networks. The multi-agent system for remote software management is created as a result of the research project

ROPE (Remote Operation Environment), which has been conducted by the Research Department at Ericsson Nikola Tesla's R&D Centre in cooperation with the Department of Telecommunications at the Faculty of Electrical Engineering and Computing, University of Zagreb.

### 1. Uvod

Agentske tehnologije omogućuju razvoj sustava u kojima programski agenti na sasvim autonoman način koordiniraju svoje zadaće kako bi obavili zadatke koje im zadaju korisnici sustava. Sposobnost samostalnoga generiranja i ispunjavanja ciljeva na osnovi danih zadataka predstavlja upravo ono svojstvo koje autonomne agente razlikuje od bilo kojega drugog tipa programskih komponenti (npr., aktivnih objekata). Kako bi agenti mogli samostalno djelovati i koordinirati svoje aktivnosti, moraju biti sposobni inteligentno se ponašati, a u nekim slučajevima biti i pokretljivi. Primjena višeagentskih sustava, koji se sastoje od inteligentnih pokretnih agenata, pokazala se veoma pogodnom za automatizaciju poslovnih procesa davatelja usluga i mrežnih operatora u telekomunikacijskom vrijednosnom lancu [1].

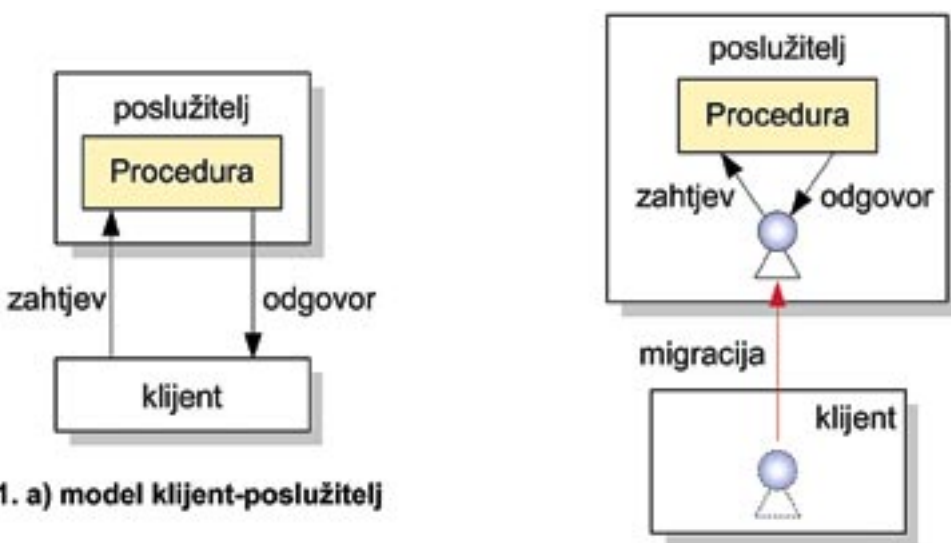
Jedan od takvih poslovnih procesa je upravljanje programskom podrškom na mrežnim čvorovima (poslužiteljima) koji uključuje održavanje i testiranje programskih komponenti na daljinu. Primjenom agentskih tehnologija, taj poslovni proces može se potpuno automatizirati, čime se uvelike smanjuju operativni troškovi davatelja usluga i mrežnih operatora vezani uz održavanje programske podrške. Stoga je u suradnji Ericssona Nikole Tesle i zagrebačkoga Fakulteta elektrotehnike i računarstva pokrenut istraživački pro-

jekt ROPE s ciljem realizacije prototipnoga rješenja za upravljanje programskom podrškom na daljinu koje je zasnovano na primjeni agentskih tehnologija.

### 2. Agentske tehnologije

Programski agenti predstavljaju relativno novu paradigmu u području programskoga inženjerstva koja je vrlo pogodna za rješavanje problema pokretljivosti nastaloga pojavom nove generacije telekomunikacijske mreže za koju je karakteristično objedinjavanje različitih vrsta postojećih mrežnih tehnologija i ostvarivanje novih konvergiranih telekomunikacijskih usluga. Problem pokretljivosti usluge uključuje pružanje usluga korisniku u pokretu te prilagođavanje usluga zahtjevima korisnika i mogućnostima korisničke opreme. Zbog svojih osobina, višeagentski sustavi s pokretnim agentima predstavljaju učinkovito rješenje za ostvarivanje potpune pokretljivosti korisnika i usluga u novim uvjetima u telekomunikacijskoj mreži.

Nagli razvoj programskih agenata započeo je korištenjem IP (*Internet Protocol*) mreža u informacijsko-komunikacijskim sustavima [2]. Do danas su kao rezultat istraživanja agentskih tehnologija razvijene različite agentske platforme te definirani jezici za komunikaciju i protokoli za koordinaciju agenata [2, 3], što je omogućilo različite primjene programskih agenata u postojećim velikim raspodijeljenim sustavima (npr., Grid) [4] i (pokretnim) telekomunikacijskim mrežama (npr. UMTS) [5].



Slika 1. Modeli raspodijeljene aplikacije

## 2.1. Svojstva autonomnih programskih agenata

Osnovno obilježje autonomnih programskih agenata je samostalno rješavanje zadaća koje im zadaju korisnici. Po toj osobini se autonomni agenti razlikuju od ostalih programskih komponenti (npr., aktivnih objekata). Ključni element samostalnoga ponašanja agenata predstavlja njihova sposobnost generiranja ciljeva koje može rezultirati proaktivnim ponašanjem. Samostalnost programskih agenata omogućuje visok stupanj automatizacije rješavanja zadaća, što je itekako važno kada smo suočeni s velikim brojem složenih poslova koje treba obaviti.

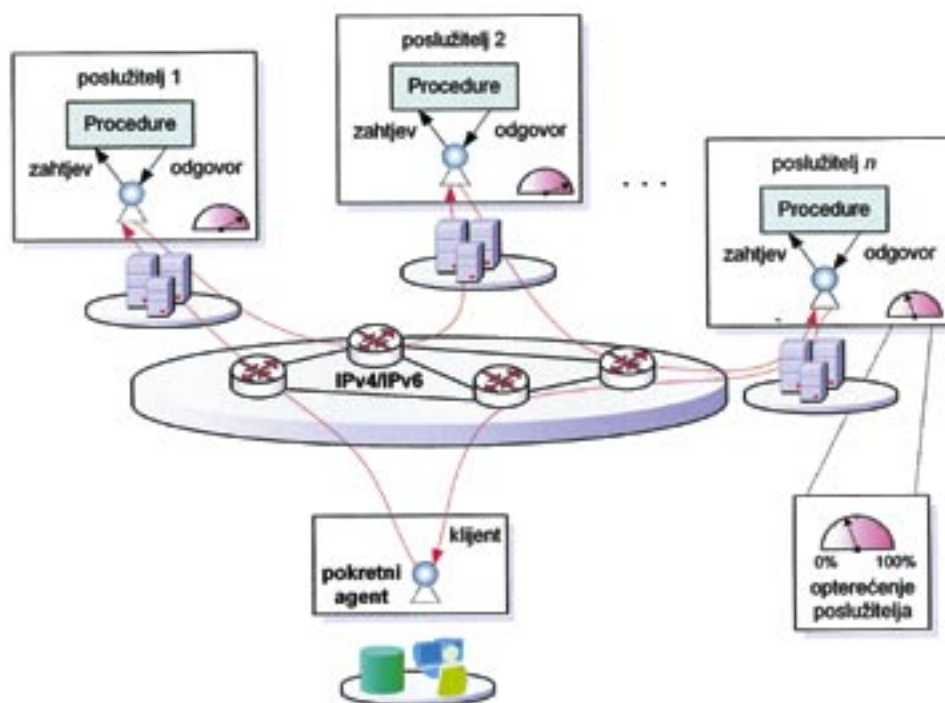
Autonomne programske agente možemo razlikovati prema inteligenciji koju posjeduju, pokretljivosti i mogućnostima međusobne suradnje. Inteligencija podrazumijeva sposobnosti učenja i odlučivanja na osnovi informacija koje agent dobiva iz okoline. Inteligencija je osnovno svojstvo tzv. inteligentnih agenata [6, 7]. Pokretljivost je sposobnost kretanja agenta u mreži i osnovno je obilježje tzv. pokretnih agenata [8]. Za razliku od ostalih tipova pokretnih programskih komponenti (npr., pokretnih objekata) koje ne upravljaju svojom migracijom, pokretni agenti se mogu potpuno samostalno kretati po mreži. Osim toga, pokretni agenti mogu prekidati izvršavanje svojih zadaća i nastaviti ih na nekom drugom računalu. Mogućnost suradnje omogućuje agentu da po potrebi koordinirano komunicira s drugim agentima s ciljem realizacije za-

jedničkoga cilja. Da bi to bilo moguće, potrebno je definirati komunikacijski jezik koji omogućuje razmjenu znanja među agentima [2]. Suradnja je osnovno obilježje tzv. kooperacijskih agenata [9, 10]. Navedena svojstva autonomnih agenata se mogu međusobno preklapati te na taj način dobivamo agente s različitim sposobnostima.

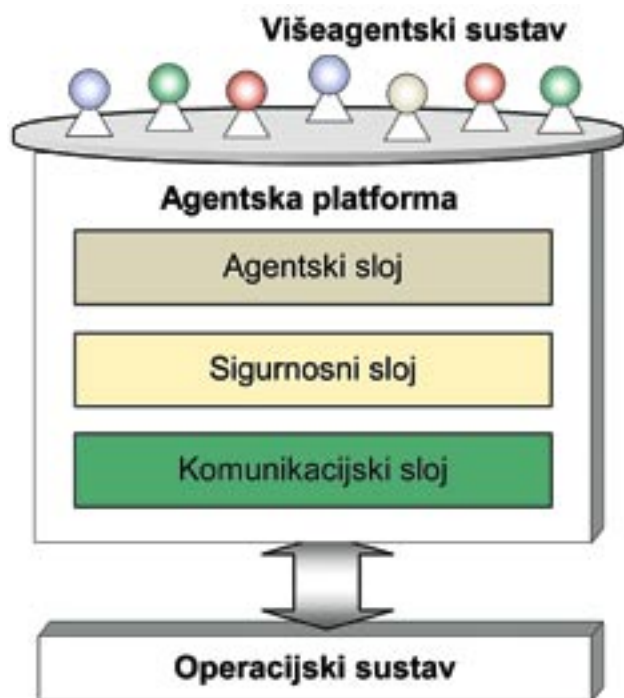
## 2.2. Pokretni agenti

Programski agenti mogu biti pokretni ili stacionarni. Stacionarni programski agenti, za razliku od pokretnih agenata, nemaju mogućnost kretanja u sustavu. Oni stoga svoje aktivnosti u višeagentskom sustavu isključivo koordiniraju primjenom agentskih konverzacijskih protokola [2]. Pokretni agenti mogu iskoristiti svoju pokretljivost kako bi učinkovito koordinirali svoje aktivnosti, uzimajući u obzir čimbenike kao što je, npr., prometno opterećenje mreže te ispad pojedinih čvorova u mreži. Zbog toga su pokretni agenti posebno pogodni za primjenu u telekomunikacijskim sustavima.

Danas se većina raspodijeljenih aplikacija temelji na modelu klijent-poslužitelj. Tipična implementacija podrazumijeva međudjelovanje putem telekomunikacijske mreže u kojemu klijentska aplikacija razmijenjuje s poslužiteljem niz zahtjeva i odgovora (Slika 1. a). Primjenom tehnologije pokretnih agenata omogućena je migracija klijentskoga dijela raspodijeljene aplikacije na poslužiteljsko računalo, čime je udaljeno međudjelovanje zamijenjeno s lokalnim (Slika 1. b).



Slika 2. Migracija pokretnog agenta u IP mreži



Slika 3. Primjer arhitekture agentske platforme

Gledano iz perspektive korisnika raspodijeljene aplikacije, model s pokretnim agentima ima nekoliko prednosti naspram modela klijent-poslužitelj. Prvo, primjenom pokretnih agenata može se znatno smanjiti prometno opterećenje telekomunikacijske mreže onda kada zadaće koje želimo obaviti na udaljenim čvorovima mogu rezultirati intenzivnom komunikacijom između klijenta i poslužitelja, npr., pokretni agent se šalje putem IP mreže na udaljeni mrežni čvor (poslužitelj) te tamo vrši obradu podataka, a nakon završene obrade se vraća s rezultatima. Na taj način izbjegnuta je razmjena velike količine informacija između mrežnih čvorova. Drugo, pokretni agenti se također mogu primijeniti prilikom rješavanja problema ograničenosti računalnih resursa, npr., migracijom u IP mreži i samostalnim pronalaženjem poslužitelja koji ima dovoljno raspoloživih računalnih resursa, pokretni agent može na potpuno automatizirani način obaviti zadaću koju mu zadaje korisnik raspodijeljene aplikacije (Slika 2.).

### 2.3. Agentska platforma

Da bi agenti pristupili i djelovali na nekom čvoru u mreži, mora postojati okružje (platforma) koje će prihvatiti pokretnoga programskoga agenta i omogućiti mu izvođenje. Okružje, odnosno platforma upravlja životnim ciklusom agenata (tj., njihovim stvaranjem i

uništavanjem) te omogućuje njihovu migraciju i komunikaciju [6]. Agentska platforma (agencija) nalazi se na čvoru u mreži (poslužitelju). Skup povezanih agentskih platformi sačinjavaju agentsko okružje unutar kojega se agenti kreću i izvode operacije. Tipična agentska platforma pruža mehanizme za:

- upravljanje agentima;
- sigurnost (povjerljivost i integritet informacija);
- osiguravanje komunikacije među agentima.

Agentska platforma omogućuje realizaciju višeagentskoga sustava koji predstavlja raspodijeljenu aplikaciju zasnovanu na agentima. Većina platformi ima arhitekturu koja sadrži agentski, sigurnosni i komunikacijski sloj (Slika 3.). Agentski sloj je odgovoran za upravljanje agentima (npr., njihovo kreiranje, kretanje, identifikaciju, itd.). Sigurnosni sloj osigurava mehanizme za zaštitu privatnosti informacija i štiti od neovlaštenih aktivnosti u informacijsko-komunikacijskom sustavu. Sigurnosni mehanizmi su potrebni za ostvarivanje sigurne komunikacije i migracije agenata. Komunikacijski sloj se brine o isporuci poruka koje agenti razmjenjuju.

### 2.4. Prednosti pokretnih programskih agenata

Primjena pokretnih agenata pruža čitav niz pogodnosti prilikom razvoja raspodijeljenih aplikacija [8]. Prednosti pokretnih agenata, koje su bile ključne prilikom odabira te tehnologije za razvoj raspodijeljene aplikacije (višeagentskoga sustava) za upravljanje programskom podrškom na daljinu, su:

- manje prometno opterećenje mreže (ušteda u širini prijenosnoga pojasa);
- samostalnost izvršavanja zadaća (autonomnost);
- mogućnost režima rada raspodijeljene aplikacije bez stalne povezanosti na mrežu (tzv. *offline* režim rada);
- smanjenje zahtjeva za korištenjem klijentskih resursa (tj., manje opterećenje klijenta);
- konkurentnost izvođenja (realizirana višeagentskim sustavom).

Manje prometno opterećenje mreže postiže se smanjenjem broja poruka koje se razmjenjuju mrežom, budući da nema stalne komunikacije između klijenta i poslužitelja, već pokretni agent odlazi na udaljeni poslužitelj i sva komunikacija se izvodi lokalno. Mogućnost režima rada raspodijeljene aplikacije bez potrebe za stalnom povezanošću putem mreže moguće je ostvariti zbog pokretljivosti agenata. Agent se šalje poslužitelju i vraća klijentu nakon samostalno obavljenooga posla. Samostalnost se očituje u činjenici da pokretni agent obavlja zadani posao bez dodatne interakcije s klijentom. Pokretni agenti smanjuju zahtjeve za resursima na klijentskoj strani, budući da većinu poslova



obavljaju na strani poslužitelja. U obavljanju složenih poslova, čiji su pojedini dijelovi konkurentni, može se kreirati i zaposliti više agenata koji u tom slučaju poslove izvode paralelno. Paralelnim izvođenjem poslova smanjuje se vrijeme obavljanja ukupnoga posla, a također se optimalno iskorištavaju procesorski kapaciteti računala u mreži. Osim toga, moguće je neki procesorski zahtjevan posao premjestiti sa slabijega računala na računalo s jačim procesorom te tako smanjiti vjerojatnost preopterećenja pojedinih čvorova u mreži.

### 3. Daljinsko upravljanje programskom podrškom

Daljinsko upravljanje programskom podrškom u raspodijeljenim sustavima je zahtjevan posao, osobito kada u takvom sustavu imamo veliki broj čvorova (računala). Administratori takvih sustava moraju obaviti na stotine operacija vezanih uz upravljanje programskom podrškom, kao što su instaliranje, pokretanje ili zaustavljanje programskih komponenti. Osim toga, u mrežama s pokretnim korisnicima i terminalima održavanje programske podrške se dodatno komplicira.

Jedan od velikih problema prilikom upravljanja programskom podrškom u telekomunikacijskim mrežama nove generacije vezan je uz testiranje novih programskih komponenti, budući da rezultati testiranja programske podrške u kontroliranom laboratorijskom okruženju mogu biti različiti od onih u stvarnom okruženju na udaljenom čvoru. Zbog toga je također potrebno izvoditi i udaljeno testiranje na čvoru na kojemu će nova programska komponenta biti instalirana. Taj problem može se učinkovito riješiti primjenom pokretnih agenata [11], pogotovo ako još postoji potreba za testiranjem programskih komponenti na većem broju čvorova u telekomunikacijskoj mreži [12].

#### 3.1. Upravljanje programskom podrškom u telekomunikacijskoj mreži

Konvergencijom usluga i mrežne infrastrukture u novoj generaciji telekomunikacijske mreže davatelji usluga i mrežni operatori sve više osjećaju potrebu za učinkovitom automatizacijom procesa upravljanja programskom podrškom u različitim tipovima mrežnih čvorova. Heterogenost mrežne infrastrukture, terminala i usluga te pokretljivost korisnika predstavljaju nove izazove s kojima su suočeni administratori sustava za upravljanje i nadzor mrežom i uslugama. S ciljem upravljanja programskim komponentama instaliranim po različitim čvorovima (poslužiteljima) u mreži, administratori primjenjuju različite aplikacije za udaljeno održavanje programske podrške. Međutim, većina tih (raspodijeljenih) aplikacija omogućuje

ograničeni stupanj automatizacije procesa upravljanja programskom podrškom na daljinu, pogotovo kada se to upravljanje odnosi na veliki broj (međusobno ovisnih) programskih komponenti koje su raspršene na velikom broju mrežnih čvorova, što je slučaj kod nove generacije telekomunikacijske mreže.

Kako bi se omogućilo održavanje programskih komponenti u novoj generaciji mreže, Istraživački odjel Instituta za telekomunikacije u Ericssonu Nikoli Tesli razvio je zajedno sa Zavodom za telekomunikacije Fakulteta elektrotehnike i računarstva sustav za daljinsko upravljanje programskom podrškom (RMS<sup>1</sup> – *Remote Maintenance Shell*) koji se zasniva na primjeni tehnologije pokretnih agenata. RMS sustav omogućuje potpuno automatizirano udaljeno upravljanje programskom podrškom na velikom broju čvorova u IP mreži [13].

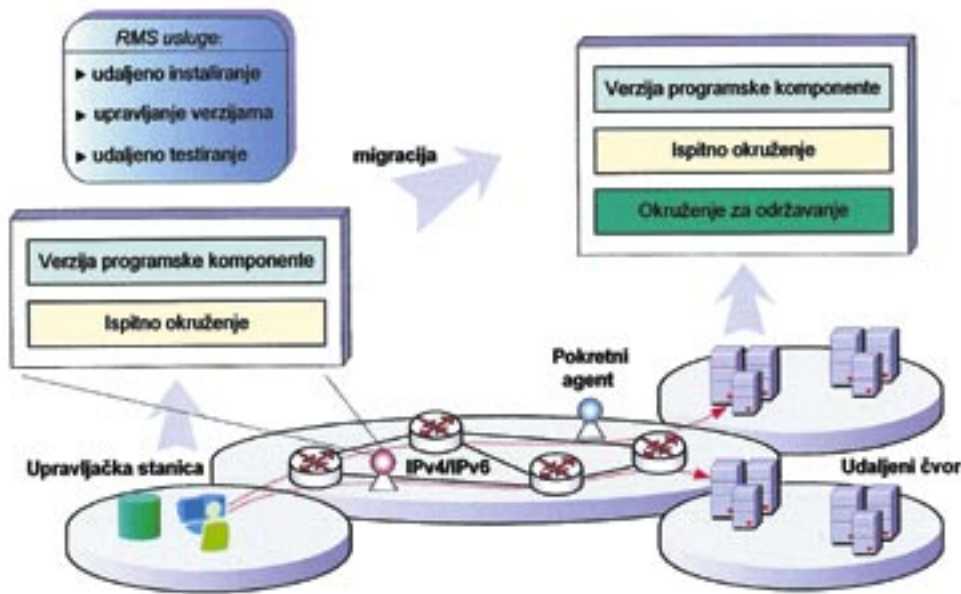
#### 3.2. RMS sustav za upravljanje programskom podrškom na daljinu

Korisniku RMS sustava za upravljanje programskom podrškom na daljinu omogućeno je jednostavno i učinkovito upravljanje programskom podrškom na daljinu primjenom višeagentskog sustava na agentskoj platformi Grasshopper [14], koji sadrži pokretne agente odgovorne za izvođenje različitih operacija vezanih uz udaljeno održavanje i testiranje programskih komponenti. Upravljanje je potpuno automatizirano, pa korisnik samo mora specificirati željenu RMS uslugu vezanu uz upravljanje određenom programskom komponentom (npr., instaliranje, zamjena verzija, itd.), a RMS sustav će realizirati uslugu, uzimajući u obzir prometno opterećenje i pokretljivost korisnika u telekomunikacijskoj mreži.

RMS sustav sastoji se od upravljačke stanice smještene na čvoru (računalu) za upravljanje i od okružja za održavanje smještenih na udaljenim čvorovima u mreži (Slika 4.), [11]. Na upravljačkoj stanici nalazi se konzola s agentima koji putem grafičkoga sučelja brinu o prikupljanju korisničkih zahtjeva i prezentiranju rezultata izvršenih operacija održavanja i testiranja. Okružje za održavanje omogućuje upravljanje programskim komponentama na određenom udaljenom mrežnom čvoru.

Svaka programska komponenta kojom želimo upravljati pomoću RMS sustava mora biti razvijena prema određenim pravilima kako bi se omogućilo njeno udaljeno održavanje i testiranje. To prilagođavanje programske komponente RMS sustavu riješeno je kroz poseban prilagodan sloj, tzv. ispitno okružje. Takav pristup osigurava standardizirano sučelje između programske komponente i sustava.

<sup>1</sup>RMS je patentiran od strane Europskoga patentnog ureda



Slika 4. Koncept rada RMS sustava

Nakon što korisnik putem grafičkoga sučelja specificira željene operacije održavanja ili testiranja nad određenim programskim komponentama na udaljenim čvorovima, agenti u upravljačkoj stanici naprave plan provedbe željenih operacija te kreiraju višeoperacijske pokretne agente kojima dodijele odgovarajuće zadatke definirane planom upravljanja. Kako bi obavili dane zadatke, svaki pokretni agent odlazi na odgovarajući udaljeni čvor u informacijsko-komunikacijskom sustavu (Slika 4.).

RMS sustav nudi korisniku osnovne i napredne mogućnosti upravljanja programskom podrškom na daljinu [11]. Osnovne usluge RMS sustava uključuju udaljeno instaliranje i upravljanje verzijama programskih komponenti, pri čemu su pokretni agenti RMS sustava odgovorni za:

- migraciju određene verzije programske komponente i ispitnog okruženja s upravljačke stanice na udaljene čvorove;
- instalaciju, pokretanje, zaustavljanje i deinstalaci-

ju odgovarajućih verzija programske komponente na udaljenim čvorovima.

Napredne mogućnosti RMS sustava uključuju mogućnosti udaljenoga testiranja i praćenja rada različitih verzija programskih komponenta na udaljenom čvoru. Za potrebe testiranja, RMS sustav može podržavati istovremeni rad više verzija iste programske komponente. Moguća su tri različita načina rada verzija programske komponente:

- uobičajeni rad u kojemu je u svakom trenutku aktivna samo jedna verzija upravljane programske komponente;
- paralelni rad (Slika 5.);
- selektivni rad (Slika 6.).

### 3.2.1 Paralelni rad

Podržavanjem istodobnoga izvođenja dviju verzija (stare i nove) određene programske komponente na udaljenom čvoru, paralelni rad omogućuje udaljeno

Slika 5. Paralelni rad verzija programske komponente





**Slika 6.**  
*Selektivni rad verzija programske komponente*

testiranje nove verzije programske komponente uspoređivanjem njezinih izlaznih podataka s izlaznim podacima stare verzije. To je osobito korisno prilikom provjere rada nove verzije programske komponente, i to na onim dijelovima koji se nisu mijenjali u odnosu na staru verziju. Na taj način je moguće izravno, na udaljenom čvoru, utvrditi kompatibilnost nove sa starom verzijom programske komponente.

Prilikom paralelnoga rada programske komponente, isti ulazni podaci se šalju u obje verzije, nakon čega se uspoređuju dobiveni izlazni podaci. Ukoliko su izlazni podaci stare i nove verzije različiti, tada se izlazni podaci stare verzije uzimaju kao relevantni, a nova verzija se gasi zbog nemogućnosti sinkroniziranja sa starom verzijom koja dalje normalno radi i obrađuje zahtjeve.

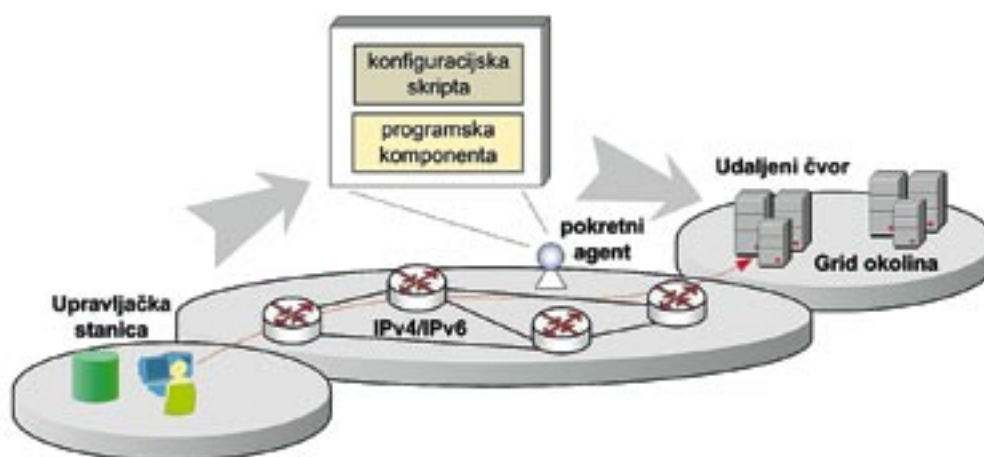
### 3.2.2 Selektivni rad

Nakon nekog vremena uspješnoga paralelnog rada, programska komponenta prelazi u fazu selektivnog rada. U tom režimu rada dvije verzije programske komponente rade tako da jedan dio zahtjeva obrađuje nova verzija, a ostatak stara verzija. Postupnim povećavanjem broja zahtjeva upućenih novoj verziji omogućuje se postupna zamjena stare s novom verzijom programske komponente.

Za razliku od paralelnoga rada programske komponente, gdje stara i nova verzija obrađuju iste zahtjeve, u selektivnom režimu rada stara i nova verzija obrađuju različite zahtjeve, pa su i njihovi izlazni podaci različiti, te stoga RMS sustav osigurava isporuku izlaznih podataka i nove i stare verzije programske komponente.

## 4. Primjena RMS sustava u Grid okolini

RMS prototip je ispitivan kroz upravljanje programskim sustavom MonALISA (*Monitoring Agents in a Large Integrated Services Architecture*), [15] koji predstavlja raspodijeljenu aplikaciju u Grid okolini Europskog istraživačkog centra CERN. Za Grid okolinu je karakteristično da su korisnici, programske komponente i računalni resursi raspodijeljeni na velikom broju mrežnih čvorova koji se nalaze u heterogenim i zemljopisno raspršenim telekomunikacijskim mrežama [16]. Inicijalni rezultati testiranja RMS prototipa u Grid okolini CERN-a pokazali su da je primjena tehnologije pokretnih agenata vrlo pogodna za automatizirano upravljanje programskom podrškom u informacijsko-komunikacijskim sustavima s velikim brojem čvorova, budući da postoji potreba za istodobnim (paralelnim) izvođenjem operacija na više čvorova [10].



**Slika 7.**  
*Premještanje programske komponente pomoću pokretnog agenata*

#### 4.1. Programski sustav MonALISA

Namjena programskoga sustava MonALISA [15] je raspodijeljeno praćenje rada usluga u Grid okolini kako bi se omogućila bolja iskoristivost resursa u tako velikom raspodijeljenom sustavu. MonALISA omogućuje integraciju postojećih aplikacija za praćenje i nadzor raspodijeljenih sustava, kao i procedure za prikupljanje podataka o mrežnim čvorovima i programskim komponentama.

Budući da se MonALISA koristi na velikom broju čvorova u Grid okolini CERN-a, održavanje i nadogradnja takvog programskog sustava, koji sadrži veliki broj međusobno ovisnih programskih komponenti, zahtijeva veliki napor administratora CERN-ovog informacijsko-komunikacijskoga sustava.

Postojeći mehanizam za upravljanje programskom podrškom u MonALISA aplikaciji periodički provjerava postoji li nova verzija programske komponente. Ukoliko se ustanovi postojanje nove verzije određene programske komponente, pokreće se proces nadogradnje koji završava s pokretanjem nove verzije programa. Takav način zamjene verzija programskih komponenti pruža samo djelomičnu automatizaciju procesa upravljanja programskom podrškom u Grid okolini. Stoga je primijenjen RMS sustav kako bi se uvela sljedeća poboljšanja:

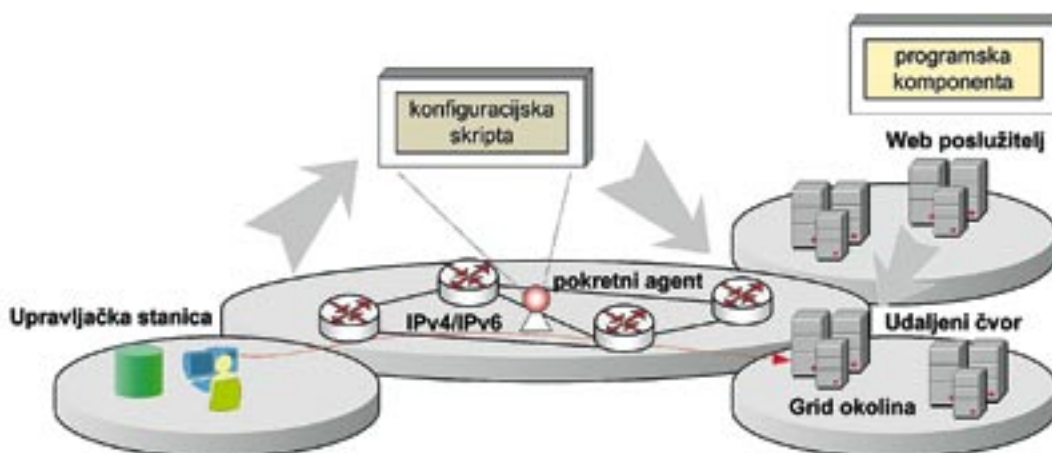
- pokretanje procesa nadogradnje programskih komponenti u trenutku kada su resursi Grid okoline najmanje opterećeni;
- postupno uvođenje novih verzija programskih komponenti;
- automatizirano vraćanje stare verzije programske komponente ukoliko novo instalirana verzija ne zadovoljava.

#### 4.2. Scenarij zamjene verzije programske komponente

Primjenom RMS sustava zasnovanoga na tehnologiji pokretnih agenata, instaliranje i pokretanje određene programske komponente MonALISA aplikacije u Grid okolini je potpuno automatizirano. Administrator informacijsko-komunikacijskoga sustava treba samo definirati na kojim mrežnim čvorovima želi zamijeniti postojeću verziju određene programske komponente s novom verzijom. U nastavku će se razmotriti scenarij zamjene verzije programske komponente na više čvorova u Grid okolini [17].

Programska komponenta instalira se na udaljenom čvoru pomoću instalacijske skripte napisane u deklarativnom jeziku ANT [18] koji omogućuje specificiranje vrlo složenih instalacijskih procedura za različite operacijske sustave. Te se procedure, specificirane od strane administratora informacijsko-komunikacijskoga sustava, pokreću na udaljenom čvoru primjenom pokretnih agenata RMS sustava. Nakon što administrator sustava putem grafičkoga sučelja u upravljačkoj stanici definira na kojim je čvorovima potrebno zamijeniti određenu programsku komponentu, kreiraju se RMS pokretni agenti kako bi obavili zahtijevanu promjenu verzija. Pokretni agent može na udaljeni čvor migrirati i novu verziju programske komponente i konfiguracijsku skriptu potrebnu za njenu instalaciju (Slika 7.), ili pak može migrirati samo konfiguracijsku skriptu, a novu verziju programske komponente dohvatiti s Web poslužitelja (Slika 8.).

Administrator informacijsko-komunikacijskoga sustava može pratiti tijek zamjene programske komponente na određenom udaljenom čvoru putem grafičkoga sučelja u upravljačkoj stanici (Slika 9.). Grafičko sučelje RMS sustava pruža na vrlo pregledan način uvid u sve promjene koje su napravljene nad određenom upravljanom programskom komponentom.



**Slika 8.**  
*Premještanje programske komponente primjenom Web poslužitelja*





Slika 9. Grafičko sučelje RMS sustava

## 5. Zaključak

Primjena agentskih tehnologija omogućuje potpunu automatizaciju procesa upravljanja programskom podrškom na daljinu u informacijsko-komunikacijskim sustavima s velikim brojem mrežnih čvorova, kao što su npr. Grid sustavi, smanjujući tako troškove njihova održavanja. Osim toga, korištenjem pokretnih agenata učinkovito se rješava čitav niz problema vezanih uz udaljeno održavanje i testiranje programskih komponenti u novoj generaciji telekomunikacijske mreže koju karakterizira heterogenost mrežne infrastrukture, terminala i usluga te pokretljivost korisnika.

Davatelji usluga i mrežni operatori mogu primjenom RMS sustava učinkovito upravljati programskom podrškom na daljinu. Korištenjem pokretnih agenata u RMS sustavu omogućena je koordinacija aktivnosti vezanih uz udaljeno održavanje na više čvorova paralelno, uz testiranje programskih komponenti, uzimajući u obzir probleme vezane uz prometno opterećenje i pokretljivost korisnika u telekomunikacijskoj mreži.

## Popis kratica

IP	Internet Protocol
MonALISA	Monitoring Agents in a Large Integrated Services Architecture
RMS	Remote Maintenance Shell
ROPE	Remote Operations Environment
UMTS	Universal Mobile Telecommunications System

## Literatura

- [1] Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sessler, R., Albayrak, S., "Agent-based Telematic Services and Telecom Applications", *Communications of the ACM*, vol. 44, no. 4., pp. 43-48, 2001.
- [2] Omicini, A., Zambonelli, F., Klusch, M., Tolksdorf, R., Eds., *Coordination of Internet Agents*, Springer-Verlag, Berlin, 2001.
- [3] The Foundation for Intelligent Physical Agents Web Site, <http://www.fipa.org/>
- [4] Ježić, G., Kušek, M., Ljubi, I., Marenić, T., Lovrek, I., Dešić, S., Dellas, B., "Mobile-Agent Based Software Management in Grid", *Proceedings of the Workshop on Emerging Technologies for Next Generation GRID*, pp. 345-346, Modena, 2004.
- [5] Gervais, M.-P., Diagne, A., "Enhancing Telecommunication Service Engineering with Mobile Agent Technology and Formal Methods", *IEEE Communications Magazine*, vol. 36, no. 7, pp. 38-43, 1998.
- [6] Knapik, M., Johnson, J., *Developing Intelligent Agents for Distributed Systems: Exploring Architecture, Technologies, and Applications*, McGraw-Hill, New York, 1998.
- [7] Tržec, K., Huljenić, D., "Intelligent Agents for QoS Management", *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, pp. 1405-1412, Bologna, 2002.
- [8] Cockayne, W. R., Zyda, M., *Mobile Agents*, Prentice Hall, Upper Saddle River, New Jersey, 1997.
- [9] D'Inverno, M., Luck, M., *Understanding Agent Systems*, Springer-Verlag, Berlin, 2001.
- [10] Ježić, G., Kušek, M., Dešić, S., Carić, A., Huljenić, D., "Multi-Agent System for Remote Software Operation", *Lecture Notes in Artificial Intelligence*, vol. 2774, pp. 675-682, Springer-Verlag, Berlin, 2003.
- [11] Kušek, M., Ježić, G., Ljubi, I., Mlinarić, K., Lovrek, I., Dešić, S., Labor, O., Carić, A., Huljenić, D., "Mobile Agent Based Software Operation and Maintenance", *Proceedings of the 7th International Conference on Telecommunications (ConTEL 2003)*,

pp. 601-608, Zagreb, 2003.

[12] Marenić, T., Ježić, G., Kušek, M., Dešić, S., "Using Remote Maintenance Shell for Software Testing in the Target Environment", *Proceedings of 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS 2004)*, pp. 19-24, Edinburgh, 2004.

[13] Lovrek, I., Ježić, G., Kušek, M., Ljubi, I., Carić, I., Huljenić, D., Dešić, S., Labor, O., "Improving Software Maintenance by Using Agent-based Remote Maintenance Shell", *IEEE International Conference on Software Maintenance (ICSM 2003)*, pp. 440-449, IEEE Computer Society Press, Amsterdam, 2003.

[14] IKV++ Technologies Web Site, <http://www.grasshopper.de/>

[15] The MonALISA Web Site, <http://monalisa.cacr.caltech.edu/>

[16] Foster, I., Keselman, C., Nick, J., Tuecke, S., "Grid Services for Distributed System Integration", *IEEE Computer*, vol. 35, no. 6, pp. 37-46, 2002.

[17] Ježić, G., Kušek, M., Marenić, T., Ljubi, I., Lovrek, I., Dešić, S., Tržec, K., Dellas, B., "Grid Service Management by Using Remote Maintenance Shell", *Lecture Notes in Computer Science*, vol. 3270, pp. 136-150, Springer-Verlag, Berlin, 2004.

[18] The Apache ANT Project Web Site, <http://ant.apache.org/>

**ADRESE AUTORA:**

**Krunoslav Tržec**

e-mail: [krunoslav.trzec@ericsson.com](mailto:krunoslav.trzec@ericsson.com)  
Ericsson Nikola Tesla d.d.  
Krapinska 45  
p.p. 93  
HR-10002 Zagreb  
Hrvatska

**Gordan Ježić**

e-mail: [gordan.jezic@fer.hr](mailto:gordan.jezic@fer.hr)  
Fakultet elektrotehnike i računarstva  
Sveučilište u Zagrebu  
Unska 3  
HR-10000 Zagreb  
Hrvatska

**Mario Kušek**

e-mail: [mario.kusek@fer.hr](mailto:mario.kusek@fer.hr)  
Fakultet elektrotehnike i računarstva  
Sveučilište u Zagrebu  
Unska 3  
HR-10000 Zagreb  
Hrvatska

**Saša Dešić**

e-mail: [sasa.desic@ericsson.com](mailto:sasa.desic@ericsson.com)  
Ericsson Nikola Tesla d.d.  
Krapinska 45  
p.p. 93  
HR-10002 Zagreb  
Hrvatska

*Uredništvo je primilo rukopis 21. ožujka 2005.*