

Miloš Pejanović,
potpukovnik, dipl. inž.
Uprava za organizaciju, mobilizaciju
i vojnu obavezu GŠ VJ,
Beograd
Dr Svetomir Minić,
pukovnik, dipl. inž.
Tehnička uprava GŠ VJ,
Beograd

METODOLOGIJE RAZVOJA INFORMACIONIH SISTEMA U VOJSCI JUGOSLAVIJE

UDC: 681.32 : 001.83] : 355.1 (497.1)

Rezime:

U radu su prikazane metode za razvoj distribuiranih objektno orijentisanih informacionih sistema, sa posebnim osvrtom na komponentne sisteme. Prikazani su neki metodološki pristupi i standardi u razvoju vojnih informacionih sistema.

Ključne reči: metode, alati, objekti, softverske komponente.

METHODS OF DEVELOPMENT OF THE YUGOSLAV ARMY INFORMATION SYSTEMS

Summary:

This paper presents methods for development of object - oriented information systems with specific emphasis on component systems as well as some methodological approaches and standards of military information systems development.

Key words: methods, tools, objects, components of software.

Uvod

U razvoju informacionih sistema, važnu ulogu ima izbor odgovarajuće metodologije i alata. Metodologije predstavljaju skup postupaka, dok su alati¹ sredstva u procesu razvoja informacionih sistema. Veliki sistemi, kao što je, na primer, vojna organizacija, potrebno je da imaju standardizovane postupke i metodologije za razvoj informacionih sistema. Postoje različiti metodološki pristupi u njihovom razvoju: sistemski integralni pristup razvoju (od projektovanja i implementacije do održavanja) i razvoj informacionih sistema kao tehničko-tehnološke strukture (standardna

mreža računara i servisa). Kombinacija ovih pristupa vodi efikasnom rešenju u razvoju informacionih sistema, pri čemu treba imati u vidu koncert „otvorenih sistema“ koji omogućavaju da različiti informatički standardi i servisi mogu da funkcionišu na tim sistemima, uključujući operative sisteme, baze podataka, računarske komunikacije i korisničke interfejsse. Brz tehnološki razvoj, posebno pojava Internet tehnologije, uzrokuje određene promene u navedenim metodološkim pristupima, što je od velikog značaja u definisanju metodoloških postupaka za razvoj savremenih informacionih sistema u vojnoj organizaciji. U ovom radu prikazani su trendovi, neke od mogućnosti i pristupi u razvoju savremenih informacionih sistema.

¹ Na primer: dijagrami toka podataka, dijagrami objekti-veze, dijagrami klasa, Use Case, itd.

Metodologije razvoja informativnih sistema

Osnovni problem u razvoju složenih informativnih sistema je savladavanje njihove složenosti. Pri tome se analizira dva aspekta: podela celokupnog razvoja na faze i dekompozicija sistema.

Podela razvoja na faze može se posmatrati kroz poznati konvencionalni životni ciklus informativnih sistema: 1. planiranje razvoja, 2. analizu i specifikaciju zahteva, 3. projektovanje, 4. implementaciju i 5. održavanje. To je „detaljna“ metodologija projektovanja i potrebno je razmotriti kada na njoj treba insistirati. Radi toga često se uvodi „prototipski razvoj“ odnosno korisnik, može vrlo brzo da vidi softver koji „radi spolja“, gde osnovni problemi nisu rešeni. Sa tog aspekta prototipovi su odbacivi ili nadgradivi. Za razvoj prototipova danas postoji veliki broj alata. Prototipski razvoj je primenjiv pri razvoju velikih i složenih sistema kod pojedinačnih aplikacija. Ovakav razvoj karakteriše nešto lošiju dokumentaciju.

Kada je moguće prototipski razvoj treba automatski prevesti u aplikaciju, odnosno kada je specifikacija izvršiva na računaru, tada je to operacionalni ili transformacioni pristup. Transformacioni pristup omogućava najjeftinije održavanje informativnih sistema. Ovakva „izvršiva specifikacija“ u veštačkoj inteligenciji naziva se „automatsko programiranje“, odnosno podizanje programskih jezika na viši nivo.

Do sredine osamdesetih godina dominantni su bili metodološki pristupi zasnovani na specifikaciji funkcija sistema,

a krajem osamdesetih pojavile su se objektnoorijentisane metode za razvoj informativnih sistema. Ove metode polaze od činjenice da sistem predstavlja skup međusobno povezanih objekata. Model sistema zasnovan je na specifikaciji objekata sistema. Svako stanje sistema određeno je stanjima objekata posmatranog sistema. Funkcije sistema realizovane su kao operacije nad objektima sistema koje mogu da menjaju stanje tih objekata. Izmene u sistemu predstavljaju izmene nad objektima sistema.

Radi savladavanja složenosti pristupa se dekompoziciji sistema na manje i lakše savladive celine ili podsisteme. Pristup može biti konvencionalni ili objektni. Konvencionalni pristup obuhvata funkcionalnu dekompoziciju i odvajanje apstrakcije podataka od proceduralnih apstrakcija (modeli podataka kao kontekst za funkcionalnu dekompoziciju). Funkcionalna dekompozicija opisuje ulazno-izlazne karakteristike bez obzira na suštinu sistema i daje „algoritamske apstrakcije“, a ne apstrakcije podataka. Baze podataka i datoteke su nusproizvod, odnosno „privatno vlasništvo“ programa bez kojeg nemaju smisla. Funkcije se često menjaju i nestabilna su osnova projektovanja, što uzrokuje teško održavanje. Odvajanje apstrakcije podataka od proceduralnih apstrakcija predstavlja odvojene baze podataka od programa. Baze podataka su nezavisan resurs. Izdvaja se model podataka i projektuje baza podataka, nakon čega se izrađuju programi nad tom bazom podataka. Kod složenijih sistema mogu nastati određeni problemi u odvajanju programa i baze podataka.

Zbog toga se pristupa objektnoj dekompoziciji uvođenjem koncepta objekta. Objektno orijentisani pristup proističe iz koncepta apstraktnog tipa podataka. To znači da projekat treba da bude apstraktan. Tip podataka je skup vrednosti i skup operacija nad njima. Za razliku od funkcionalnog pristupa gde su datoteke ulaz i izlaz, a sistem čista funkcija koja se realizuje algoritamski, objektni pristup podrazumeva da objekat (apstraktni tip podataka) sadrži i apstrakcije podataka i proceduralne apstrakcije koje su „učaurene“ i predstavljaju njegove unutrašnje karakteristike. Realni sistem predstavlja skup objekata i operacija nad objektima. Razvojem informacionih sistema želi se postići da tipovi podataka budu bliski realnom sistemu, odnosno da se dobije formalna specifikacija koja će omogućiti što manju „semantičku prazninu“.

Pojedinačne metodologije koriste skup alata, tehnika i aktivnosti za rešavanje problema u razvoju informacionih sistema. Najznačajniji savremeni alati su: dijagrami toka podataka (strukturna sistem analiza – SSA), dijagrami objekti – veze, objektni dijagrami (dijagrami klasa), dijagrami slučajeva korišćenja (Use Case), dijagrami tragova događaja (scenarija), dijagrami promene stanja, Petrijeve mreže itd.

U razvoju informacionih sistema, razmatra se mogućnost da baza podataka nije samo statički već i dinamički odziv na ažuriranje. Tada se govori o aktivnim sistemima, gde se njihova stanja proširuju sa ulazima i izlazima, a dinamika opisuje preko pravila integriteta. Sistem se opisuje neproceduralno, a ne algoritamski.

Aplikativno najpogodniji je objektni pristup. Kod različitih pristupa postoji problem izbora metoda projektovanja, odnosno stvaranja optimalnog metodološkog pristupa razvoja informacionih sistema. Takođe, postoji problem razvoja zbog parcijalne automatizacije u uslovima masovnog uvođenja personalnih računara. U takvim uslovima često se pristupa razvoju informacionih sistema kao infrastrukture i definisanju standarda, ne čekajući globalni projekat. Nakon izvesnog vremena postoji mogućnost da dođe do redundanci i haosa, zbog čega je potrebno ostvariti kontrolu i upravljati razvojem. Koordinacija nad lokalnim podsystemima, pored standarda, podrazumeva rečnik podataka, što znači postojanje „informacionog sistema o informacionim sistemima“. Rečnik podataka može se dobiti primenom poznatih metodoloških pristupa i alata kao što su objektni i funkcionalni pristup. Ovakvom decentralizacijom dobijaju se federativni sistemi sa bazama podataka distribuiranim na specifičan način. Metodologija je sadržana u rečniku podataka, ali ne znači da mora biti nametnuta. Ovakav pristup u razvoju informacionih sistema zahteva dosta znanja.

Postoje različite objektnoorijentisane metode koje predlažu različite pristupe objektnoorijentisanoj analizi i projektovanju informacionih sistema. Među njima su najpoznatije OMT metod (Object Modeling Technique), Boochov metod i OOSE metod (Object-Oriented Software Engineering).

U OMT metodu posmatraju se tri različita modela: objektni, dinamički i funkcionalni, koji predstavljaju različite poglede na sistem i međusobno su komplementarni. Ovaj metod daje pred-

nost pristupu u kome analiza sistema započinje definisanjem entiteta i formiranjem objektnog modela. Ovakav pristup naziva se pristup orijentisan podacima (data driven approach).

U Boochovom metodu definisani su različiti dijagrami: dijagram klasa, dijagram objekata, dijagram interakcije objekata i dijagram prelaza stanja. Njima su predstavljene statičke i dinamičke karakteristike sistema. U ovom metodu posebno je naglašen onaj deo procesa razvoja sistema koji se odnosi na projektovanje i implementaciju sistema.

U OOSE metodu definisane su tri vrste objekata. To su objekti entiteta (entity objects), interfejs objekti (interface objects) i upravljački objekti (control objects). Za opis dinamike sistema uvedeni su slučajevi korišćenja (Use Cases). Ovaj metod daje prednost pristupu u kome definisanje zahteva i analiza sistema započinju definisanjem slučajeva korišćenja i formiranjem modela slučajeva korišćenja. Ovaj pristup naziva se pristup orijentisan slučajevima korišćenja (Use Case approach).

Uvidom u navedene objektnoorijentisane metode dolazi se do zaključka da sve te metode imaju jedan zajednički skup elemenata koji se koristi u modeliranju sistema. Zbog toga su razvijeni elementi modela objedinjenog jezika modeliranja (Unified Modeling Language – UML), koji je nastao sredinom devedesetih,² prvenstveno objedinjavanjem OMT metode, Boochove metode i OOSE metode. Analizom koncepata obuhvaćenih u

objektnoorijentisanim metodama došlo se do skupa modela i elemenata modela koji se koriste pri analizi, projektovanju i dokumentovanju elemenata sistema.

Razvoj distribuiranih objektnoorijentisanih vojnih informacionih sistema

Informacioni sistemi koji imaju karakteristike distribuiranog hardvera, i/ili distribuiranog upravljanja, i/ili distribuiranih podataka, predstavljaju distribuirane informacione sisteme. Distribuirani informacioni sistemi³ obično se dele na: snažno spregnute sisteme (multiprocesorski sistemi), umereno spregnute sisteme (lokalna računarska mreža – LAN sa klijentserver arhitekturom), slabo spregnute sisteme (potpuno autonomni računari sa mogućnošću komuniciranja LAN i WAN).

U užem smislu, može se reći da su samo umereno spregnuti pravi distribuirani sistemi. Osnovni tehnološki razlog za razvoj distribuiranih informacionih sistema je povoljniji odnos cena – performans za mrežu mikroracunarskih sistema, nego za veliki sistem.

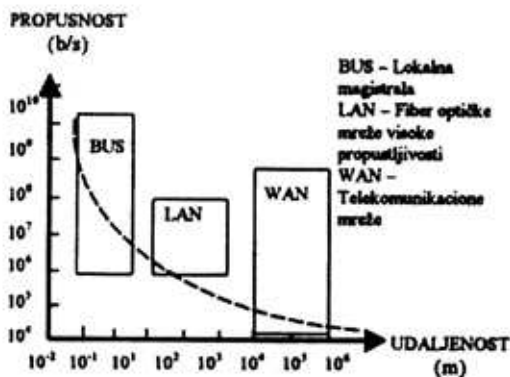
Pored toga, primena je važna u deljenju resursa. U tom smislu postoji težnja da se u razvoju postigne povećanje propusnosti ovih sistema i smanjenje vremena odziva.

Odnos između propusnosti i udaljenosti resursa prikazan je na slici 1.

Referentni model za povezivanje otvorenih sistema zasnovan je na konceptu hijerarhijske organizacije komunikacione arhitekture u sedam diskretnih slojeva. Time se ceo problem komuni-

² Prva verzija UML data je 1995. godine pod nazivom Unified Method autora G. Booch-a i J. Rumbaugh-a. Novije verzije prihvatile su i vodeće svetske kompanije u proizvodnji softvera kao što su ORACLE, IBM, Digital Equipment, Microsoft i drugi.

³ Na osnovu modela ENSLOW-a.



Sl. 1 - Propusnost u distribuiranim informacionim sistemima

ranja razdvaja na manje, zaokružene celine. Svaki sloj obezbeđuje višem sloju definisani servis, dok najviši sloj svoj servis pruža korisničkoj aplikaciji koja je izvan OSI⁴ okruženja.

Objektno orijentisani distribuirani informacioni sistemi zasnovani su na konceptu objekta (apstraktnog tipa podataka). U suštini, to su računarske mreže sa informacionim sistemima koji imaju karakteristike distribuiranih resursa. Razvoj globalne računarske mreže bazirane na Internet tehnologiji i klijent-server arhitekturi, kao i njena masovna upotreba, bitno utiču na redefinisane standarda za razvoj objektnoorijentisanih distribuiranih informacionih sistema.

U odnosu na ukupne troškove razvoja informacionih sistema klasičnom metodologijom, troškovi razvoja softvera su veći od troškova razvoja hardvera. To znači da faktički dolazi do napuštanja takvog pristupa razvoju. Pored toga, različite softverske aplikacije u takvom sistemu ne komuniciraju međusobno ili je to

minimalno izraženo. To znači da cena razvoja i održavanja softvera ostvaruje veliki skok.

Prelazak sa strukturalnog i modularnog programiranja u objektnoorijentisano programiranje reducira cenu softvera. Pored toga, softverske komponente se integrišu sa aplikacijama, a aplikacije u celoviti sistem. Pri tome je potrebno da softverske komponente pri pojavi „događaja“ međusobno komuniciraju i sa drugim softverskim komponentama, koristeći postojeće mehanizme i otvorene (open) interfejsne. Sve kompleksniji hardver i softver uzrokuje složeniju komunikaciju između različitih komponenti.

Novi trendovi u poslovanju – Internet poslovanje, postavili su nove zahteve pred Web servere. Oni moraju na neki način podržati transakcije u novom okruženju. Proširivanje uloge Web servera vodi njihovoj evoluciji u kompleksnije aplikativne servere.

Najopštije, transakciju možemo definisati kao interakciju (uobičajeno dva aktera) koja menja stanje sistema. Potreba za modelovanjem poslovnih i drugih realnih sistema u kojima se najveći deo dinamike sistema odvija kroz mnoštvo transakcija, uslovljena je praviljenjem mehanizma koji će ih u projektovanim IS na pravi način podražavati.

Transakcije se mogu izvršavati direktno ili svrstavajući se u red čekanja. U direktnim se uspostavlja sinhronizovana komunikacija između klijenta i servera, čime se dobija utisak da je uspostavljena direktna veza.

Tipičan primer ovakvog izvršavanja transakcija uočava se u CGI aplikacijama koje podržavaju Web serveri. Noviji pri-

⁴ ISO/OSI referentni model je, pre svega, namenjen za telekomunikacione mreže, zbog čega se ne koristi u LAN (IEEE802)

stup koji se oslanja na CORBA⁵ model i Object Request Broker (ORB) još jedan je primer direktnog izvršenja transakcija.

Transakcije koje se mogu kompletno izvršiti na jednom čvoru mreže nazivaju se lokalnim transakcijama. Priroda distribuiranih sistema uslovlila je pojavu distribuiranih transakcija, za čije je izvršenje potrebno više čvorova mreže. One su kompleksnije i teže za implementaciju. Za izvršenje je potrebna saradnja sa nekim alatom za koordinaciju distribuiranih transakcija (DTS – eng. distribution transaction coordinators). Jedan takav servis definisala je OMG, i to je Object Transaction Service (OTS). Drugo popularno rešenje daje JavaSoft u vidu Java Transaction Servicea (JTS).

Implementacija logike posla aplikacije zahteva od razvojnih timova da brinu o upravljanju deljivim resursima servera (kao što je memorija ili procesorsko vreme). Takve transakcije nazivaju se programirane (eng. Programmatic transactions). Aplikacije koje ih koriste su teške i skupe za održavanje. To je jedan od razloga zašto je razvoj komponentnih modela postao popularan i doživeo veliki napredak poslednjih godina. Komponente kojima je implementiran mehanizam izvršavanja transakcija na ovaj način se mogu koristiti u pravljenju novih aplikacija. Komponentni model omogućava lakše izmene i održavanje. S druge strane, skraćuje se vreme potrebno za implementaciju, jer postoje različite gotove komponente koje se mogu iskoristiti.

⁵ CORBA (Common Object Request Broker Architecture) jeste konceptualno „softverska magistrala“ (softvere bus) koja dozvoljava aplikacijama da komuniciraju međusobno, u zavisnosti od toga ko projektuje i razvija platforme na kojima se izvršavaju, jezik na kojem je pisan program i gde se izvršava.

Ovakav pristup implementaciji transakcija naziva se deklarativnim pristupom. Programeri su oslobođeni brige o realizaciji mehanizama za kontrolu izvršenja transakcija. Trenutno su najpopularniji komponentni modeli COM+ (eng. Component Object Model), EJB (eng. Enterprise Java Beans) i CORBA.

Microsoftovo rešenje COM definiše standardni interfejs za binarne objekte u Windowsu. Izvršni kod ovako napisanih objekata mogu izvršavati i drugi objekti. Kako se radi o izvršnom kodu, na taj način mogu komunicirati objekti napisani u različitim programskim jezicima. Komunikacija se vrši preko utvrđenog interfejsa koji je, u stvari, kolekcija naziva virtuelnih funkcija. Dublje upuštanje u funkcionisanje COM modela je izvan okvira ovog rada, pa će se zadržati na zaključku da je na ovaj način omogućeno korišćenje već napisanih i testiranih izvršnih modula koji obavljaju neke složene poslove. ActiveX kontrole zasnovane su na COM tehnologiji. To su komponente koje imaju omotač koji browseru omogućava da ih prikaže.

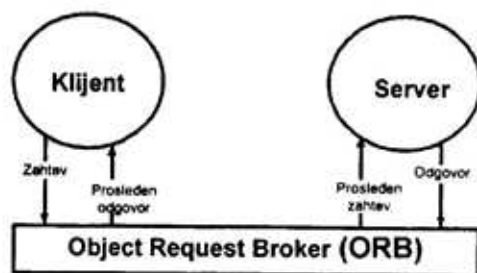
CORBA je industrijski standard za distribuirane aplikacije koji je definisao OMG (Object Management Group). OMG⁶ pokušava da pretvori u stvarnost ideju o mogućnosti da se komponente softvera mogu koristiti ako su već jednom napisane ili da se mogu kupiti gotove od drugih proizvođača. Ovaj standard definiše interfejs koji bi trebalo da omogućiti da komponente pisane u različitim jezicima međusobno saraduju. To je ostvareno definisanjem seta metoda koje

⁶ OMG (Object Management Group) formirana je 1989. godine.

su vidljive za ostale komponente. Od početka je zamišljeno da komponente mogu da budu postavljene na različite čvorove u mreži. Bez obzira na to da li su na istom serveru ili su distribuirane kroz mrežu, komponente komuniciraju zahvaljujući uslugama ORB (eng. Object Request Broker). ORB ima mehanizam za poziv udaljenih procedura (RPC), kojima za komponente potpuno maskira mrežu i sve komunikacione tehnologije, čime omogućava da se pozivi funkcija vrše kao da su lokalni bez obzira na to gde se nalaze. Ni klijent ni server na taj način više ne moraju brinuti o lokaciji, jeziku na kojem je komponenta pisana, niti o načinu na koji se vrši transport podataka kroz mrežu.

ORB može biti implementiran kroz proces na hostu kojem klijent pristupa, kao proces na nekom centralnom računaru kroz koji klijent i server komuniciraju ili kao servis operativnog sistema. U komunikaciji se koristi IIOP protokol (eng. Internet Inter-ORB Protocol).

Klijent i server CORBA, komponente slobodno komuniciraju kroz mrežu i pozivaju udaljene funkcije. To je upravo ono što je neophodno za ostvarenje koncepta aplikativnog servera kao srednjeg sloja u kome izvršenje posla aplikacije zavisi od ispunjenja zahteva upuće-



Sl. 2 – Klijent-server komunikacija kroz ORB

nog serveru koji je zadužen za logiku aplikacije. CORBA, dakle, predstavlja model koji potpuno podržava ideju troslojne arhitekture, koja po prirodi ima aplikaciju distribuiranu između klijenta, aplikativnog servera i servera baze podataka.

EJB je relativno nov model koji je vrlo brzo postao veoma popularan. EJB je Java program spakovan kao komponenta koja se koristi na strani servera. Za razliku od JavaBeanova koji se koriste prvenstveno za interpretaciju korisničkog interfejsa u okviru browsera, EJB pristupa Java klasama i kroz mrežu pruža usluge izvršavanja transakcija i obezbeđivanja sigurnosti.

U organizacionim naukama vojna organizacija se posmatra kao dinamički skup entiteta (ljudi i sredstava) između kojih se u toku dinamike aktivnosti formiraju i razrešavaju specifični odnosi (relacije) koji su organizovani na specifičan način i sa jasnim ciljem. Zato je veliki značaj razvoja vojnih informacionih sistema u obezbeđenju informacija za funkcionisanje vojne organizacije. U ovom radu nisu detaljnije razmatrani vojna organizaciona struktura i funkcije. Na primer, u ovom području američka vojska ima standard (dokument) za razvoj svojih tehničkih sistema i informacionih sistema pod nazivom „Joint Technical Architecture“ (JTA).

JTA treba da obezbedi efikasno izvršavanje vojnih operacija sa različitim snagama, bilo gde u svetu i u bilo kojem trenutku. Kao kritična potreba javlja se mogućnost sistema informacionih tehnologija u obezbeđenju sadejstva i razmene informacija. Proučavanjem konflikata i

operacija došlo je do nove vizije u američkom ministarstvu za odbranu (DoD). Kao konceptualni obrazac javlja se „Joint Vision 2010 (JV 2010)“. JTA je ključni dokument ministarstva za odbranu u ostvarenju JV 2010. Obezbeđuje osnovu za neprekidnu i dobro povezanu međuoperativnost sistema DoD, definiše područja servisa, interfejsa i standarde (JTA elemente) primenjive u svim DoD podsystemima, pri čemu su primenjivi u upravljanju, razvoju i lociranju novih ili postojećih sistema kroz DoD.

JTA sadrži dva glavna dela: JTA jezgro, koje sadrži minimalni skup JTA elemenata primenjivih u svim Do podsystemima za obezbeđenje međuoperativnosti i JTA dodatak⁷ koji sadrži dodatne JTA elemente primenjive u specifičnim funkcionalnim domenima (familijama sistema).

JTA je komplementaran i konzistentan sa drugim DoD programima i inicijativama u razvoju „efektiva“ i međuoperativnih informacionih sistema. To je „otvoreni“ dokument koji sadrži razvojne tehnologije i objedinjene bazične standarde. Stalno ga usavršavaju i otvoren je za pristup svim zainteresovanima. Ovaj dokument predstavlja primer jednog savremenog integralnog pristupa u razvoju tehničkih sistema, uključujući i informacione sisteme.

⁷ Na primer, verzija 3.1. ovog dokumenta uključuje dodatak za C4ISR domene (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance), zatim Combat Support domain, Modeling and Simulation domain i Weapon Systems domain.

Zaključak

Posledica ubrzanog razvoja tehnologije i porasta informacija jeste uticaj na metodološke pristupe i koncepte u razvoju informacionih sistema. U poslednje vreme došlo je do razvoja računarskih mreža i Internet tehnologije, što je od velikog značaja i za Vojsku Jugoslavije. Potrebno je obratiti pažnju na značaj TCP/IP protokola i telekomunikacione infrastrukture sa multiservisnim sposobnostima, koji će predstavljati jezgro budućih distribuiranih informacionih sistema. Zbog postojanja sopstvene telekomunikacione infrastrukture, vojni informacioni sistemi će se, verovatno, bazirati na takvoj osnovi.

Navedena razmatranja pokazuju da je osnova uspešnog razvoja vojnih informacionih sistema praćenje i uvođenje savremenih metodoloških pristupa, tehnika i alata.

Literatura:

- [1] John Robert Beishline, Ph. D. Colonel, General Staff Corps, U. S. Army, MILITARY MANAGEMENT for NATIONAL DEFENSE, Vojno delo, Beograd, 1952.
- [2] VJINFO2001, zbornik radova sa seminara o primeni informatike u Vojsci, Beograd, april 2001.
- [3] OMB Unified Modeling Language Specification, Version 1.3, June 1999.
- [4] Arun Rai, T. Ravichandran and Subhashish Samaddor, How to anticipate the Internet's global diffusion, communications of the ACM, October 1998/vol. 41. № 10.
- [5] Harlan McGhan, Mike O'Connor, PicoJava: A Direct Execution Engine For Java Bytecode, Computer, October 1998.
- [6] Communications of the ACM, October 1998/vol. 41. №1.
- [7] Joint Technical Architecture ver. 3.1, Department of Defense, March 2000.
- [8] Pejanović, M.: Internet tehnologija kao okruženje za razvoj informacionih sistema, Vojnotehnički glasnik, 6/2000.
- [9] Pejanović, M., Minić, S.: Automatizovani sistemi za podršku odlučivanju u vojnoj organizaciji, SYM-OP-IS 2000, Beograd.
- [10] Pejanović, M., Minić, S.: Metodološki pristupi u razvoju vojnih informacionih sistema, SYM-OP-IS 2001, Beograd.