

S. Radojević<sup>1</sup>, J. Kratica<sup>1</sup>,  
M. Vugdelija<sup>1</sup>, V. Filipović<sup>2</sup>

## OGRANIČAVANJE STVARANJA APLIKATIVNIH OBJEKATA U PROIZVODNIM INFORMACIONIM SISTEMIMA

### Rezime

*Aplikativni objekat ujedinjuje neophodne datoteke za rad metoda i sam metod. Korisnici proizvodnog informacionog sistema stvarajući aplikativni objekat izvršavaju projektom određene radnje na relacionoj bazi podataka. U ovom radu je prikazana jedna metodologija koja stvara osnovu za kontrolisanje rada korisnika u proizvodnom informacionom sistemu.*

### 1. Uvod

Proizvodni informacioni sistem (skraćeno *PIS*) mora posedovati podsistem kojim se kontroliše rad krajnjih korisnika [1],[2]. Ovaj podsistem stvara osnovne pretpostavke za zatvaranje toka informacija u proizvodnom preduzeću. Analize funkcionalnih *PIS* pokazale su neophodnost podele aktivnosti korisnika nad bazom podataka. Ova podela je realizovana stvaranjem kontrolno upravljačkog jezgra baze podataka *PIS*-a. Jezgro je zasnovano na generativnim objektima koji kontrolišu pristup bazi podataka upravljajući stvaranjem aplikativnih objekata. Generativni objekti su deo operativnog sistema na kojima radi *PIS*. Njihovo stvaranje je strogo kontrolisano i određeno. Generativni objekti nemaju mogućnost promene stanje baze podataka *PIS*-a. Oni kontrolišu i obezbeđuju korisnicima pristup bazi podataka preko sopstvene baze podataka. Aplikativni objekat predstavlja jedinstvo datoteka i metoda. Metod je programska reprezentacija grupe transakcija nad određenim datotekama baze podataka. Datoteke koje metod može koristiti u toku izvršavanja nazivamo neophodnim datotekama za rad metoda. Aplikativni objekat stvara korisnik u toku svog rada, da bi promenio stanje baze podatka *PIS*-a. Stvaranje aplikativnih objekata kontroliše deo jezgra *PIS*-a. U nastavku ovog rada prikazana je kontrola stvaranja aplikativnih objekata pomoću generativnog objekta privilegija dela jezgra.

### 2. Generativni objekat privilegija korisnika

Vrednost objektno orijentisane metodologije posebno je izražena u projektovanju baze podataka. Klasa kao definicija objekta, omogućava projektantima

<sup>1</sup> mr Slobodan Radojević, asistent; mr Jozef Kratica, asistent; Milan Vugdelija, asistent pripravnika; Katedra za Matematiku, Mašinski fakultet, Univerzitet u Beogradu, 27. marta 80. 11000 Beograd

<sup>2</sup> Filipović Vladimir, asistent pripravnika, Matematički fakultet, Univerzitet u Beogradu, Studentski trg 16. 11000 Beograd

brži i efikasniji rad. Nasleđivanje i kasno povezivanje su osnova za stvaranje vrlo složenih metoda koji su primenljivi na svim nivoima klasne hijerarhije. Učaurivanjem objekta, dobija se na brzini i sigurnosti programskog sistema.

Klasama se definiše objekat-slog datoteke. One sadrže definiciju podataka i metoda koji su objedinjeni u objektu. Generativna klasa je klasa koja u svojoj definiciji sadrži metod prema kome se stvaraju objekti koje definiše. Objekte ovih klasa nazivamo generativnim objektima. Metodi u generativnim klasama najčešće služe za stvaranje dinamičkih struktura podataka u generativnom objektu. Jedan oblik generativnog objekta privilegija može se definisati na sledeći način: ..

```
stvarno_ime = <CHAR,8>
javno_ime = <CHAR,25>
opis = <CHAR,70>
aplikativni_objekat = <ime : stvarno_ime,
                        opis : opis>
aplikativni_objekti = {aplikativni_objekat}
privilegije = <korisnik : korisnici_PIS,
               objekat : aplikativni_objekti>
privilegije_stvaranja = {privilegije}
privilegija_stvaranja = <korisnik : korisnici_PIS-a,
                        &{privilegije_stvaranja : izaberi korisnik}>
privilegija_stvaranja_korisnika = {privilegija_stvaranja}
```

Za svakog korisnika PIS-a pravi se spisak aplikativnih objekata koje može stvarati. Metod izaberi sa argumentom korisnik iz datoteke klase *privilegije\_stvaranja* odabira za korisnika imena aplikativnih objekata koje može stvoriti. Imena aplikativnih objekata se nalaze u listi pokazivača na objekte datoteke *aplikativni\_objekti*.

Stvaranje generativnih objekata klase *privilegija\_stvaranja\_korisnika* koji sadrže dinamičku listu pokazivača na aplikativne objekte može biti otežana. Generativni objekti su pridruženi operativnom sistemu i njihovo stvaranje direktno utiče na veličinu operativne memorije. Ako je broj korisnika PIS-a veliki, što je najčešće slučaj, to za posledicu ima smanjenje operativne memorije server računara. Posledica je smanjenje efikasnosti celokupnog računarskog sistema. Da bi se ovaj problem rešili uvedimo jednu izmenu u definisanju klase privilegija stvaranja aplikativnih objekata za jednog korisnika.

### 3. Generativni objekat privilegija grupe korisnika

Korisnici PIS-a su organizovani prema organizacionim celinama u preduzeću. Oni mogu imati grupe zajedničkih poslova koje obavljaju na PIS-u. Ako koristimo ovu činjenicu, možemo na sledeći način definisati klasu *privilegije\_stvaranja*.

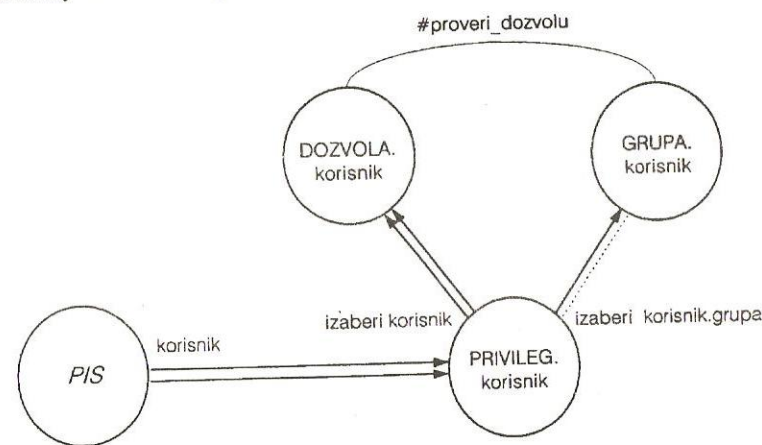
```
grupa = <ime : dužina_niske>
korisnik = <ime : dužina_niske,
           šifra : dužina_niske,
           grupa : grupa>
privilegije_grupe = <grupa : grupa,
                    objekat : aplikativni_objekat>
privilegije_stvaranja_grupe = {privilegije_grupe}
dozvola = <korisnik : korisnici_PIS-a,
           objekat : aplikativni_objekti,
           dozvola : BOOLEAN>
```

4.14

```
dozvole = {dozvola}
privilegija_stvaranja =
  <korisnik : korisnici_PIS-a,
   @(&{privilegije_stvaranja_grupe : izaberi korisnik.grupa})
   #proveri_dozvolu &{dozvole : izaberi korisnik}>
privilegije_stvaranja = {privilegija_stvaranja}
```

Korisnik PIS-a organizaciono pripada samo jednoj grupi korisnika u okviru preduzeća. Zato se i definiše njegova pripadnost grupi. Grupi korisnika se dodeljuju privilegije stvaranja aplikativnih objekata. U odnosu na svoju grupu, korisniku se može dozvoliti stvaranje novih aplikativnih objekata, ili mu se može zabraniti stvaranje nekih objekata. Dozvole i zabrane sadrže objekti datoteke definisane klasom *dozvole*. Objekat klase *privilegije\_stvaranja* je složen i njegovo stvaranje zahteva niz pripremnih aktivnosti. Prvo se mora odrediti grupa kojoj korisnik pripada i odrediti odgovarajuća lista privilegija stvaranja aplikativnih objekata. Na tu listu se funkcionalno dodaju i ograničenja u stvaranju objekata za korisnika. Metod *izaberi* sa argumentom *korisnik.grupa* odabira korisnikovu grupu a dodavanje ograničenja obavlja metod *proveri\_dozvolu*. Kako korisnik može pripadati samo jednoj grupi, za posledicu ima jedinstvenost liste pokazivača na aplikativne objekte koje grupa može stvoriti. Zato se mora proveriti da li je takva lista već formirana. Operator *@* preuzima na sebe proveru i stvaranje navedene liste koja predstavlja objekat sam za sebe.

Analiza objekta klase pokazuje da je on složen objekat koji u suštini upravlja sa dva podobjekta koji su njegov sastavni deo. Na slici 1. je prikazan način komunikacije između ovih podobjekata.



Slika 1.

Stvaranje objekata se naglašava sa dve paralelne usmerene linije. Ukoliko je za stvaranje objekata potrebna i odgovarajuća poruka ona se daje na početku ovih linija. Isprekidana linija označava uslovno stvaranje objekta, sa prethodnom proverom njegove egzistencije. Funkcionalna zavisnost je označena sa krivom linijom koja povezuje zavisne objekte.

4.15



#### 4. Zaključak

Jezgro baze podataka projektuje se i uvodi u *PIS*, da bi se kontrolisale sve aktivnosti korisnika. Posebno one aktivnosti koje menjaju stanje baze podataka u *PIS*-u. Generativni objekti su pružili dobru osnovu za formiranje jezgra baze podataka. Jedna generativnih objekata, koja je opisana u ovom radu, kontroliše stvaranje aplikativnih objekata koristeći njihova imena. Imena aplikativnih objekata zajedno sa grupama korisnika i korisnicima su osnova za dodelu privilegije stvaranja aplikativnih objekata.

Generativni objekti na ovaj način stvoreni, najmanje opterećuju računarski sistem. Omogućavaju kontrolu korisnika i nadgledanje njihovog rada, čime se podiže odgovornost korisnika u *PIS*-u. To za posledicu ima otkrivanje prekida toka informacija u preduzeću.

#### 5. Literatura

- [1] *"Sistem za praćenje aktivnosti korisnika nad objektno orijentisanom bazom podataka u ISPP-u"*  
S. Radojević, M. Milačić  
XX "Jupiter", naučni skup, strane 71-76, septembar 1994, Beograd.
- [2] *"Organizovanje sistema privilegija za pristup u ISPP"*  
S. Radojević, M. Vugdelija, D. Radojević  
"XXI YU-SYM-OP-IS", naučni skup, strane 303-303, oktobar 1994, Kotor

#### Restrict on create applications object in information system for production administering

*Application object has two parts. First is names of necessary files and second is a method. Only user's of information system for production administering can create application object, when they started they jobs. In this paper we present one basic methodology for control user's jobs, controlling application object.*