

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ZAVOD ZA TELEKOMUNIKACIJE

UPUTE ZA RAD

Pomagala za razvoj programskog proizvoda
IBM Rational RequisitePro i Rose Enterprise

Željka Car

Hrvoje Belani

Zagreb, ožujak 2009.

Ove upute predviđene su za studente 2. semestra diplomskog studija Računarstvo, profil Telekomunikacije i informatika, pri izvođenju dijela zadataka za domaću zadaću iz izbornog kolegija „Razvoj telekomunikacijske programske podrške” na Zavodu za telekomunikacije FER-a.

Sadržaj

Uvod.....	4
1. RequisitePro	5
1.1. Projekti	5
1.2. Vizija	6
1.3. Zahtjevi.....	6
1.4. Lokalno pohranjivanje vizije.....	7
2. Rose Enterprise	8
2.1. Dijagram slučaja uporabe.....	9
2.2. Klasni dijagram	10
2.3. Slijedni dijagram	10
2.4. Kopiranje dijagrama.....	10
2.5. Povezivanje zahtjeva i pripadnih slučajeva uporabe.....	10
3. Literatura	12
Dodatak 1	13
Dodatak 2: Model FURPS+	14

Uvod

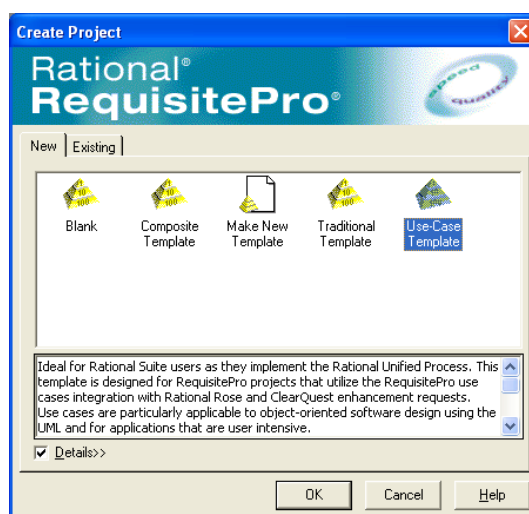
Pomagala za razvoj programskih proizvoda *IBM Rational RequisitePro* i *Rose Enterprise* pripadaju programskom paketu *Rational Suite* kompanije *IBM* i služe za podršku pojedinim fazama razvoja programskog proizvoda definiranim prema procesnoj metodologiji *RUP* (*Rational Unified Process*).

Mjesto rada s pomagalima: Zavod za telekomunikacije (u daljnjem tekstu: ZTEL), 7. kat zgrade C, *laboratorij C7-03* (*laboratorij „Boole“*).

1. RequisitePro

Rational RequisitePro je pomagalo za upravljanje programskim zahtjevima. Zahtjev se definira kao svojstvo, sposobnost ili ograničenje koje se postavlja nad sustavom koji se razvija. Da bi se zahtjevi mogli prikupiti, razjasniti i pratiti potrebno ih je tekstualno dovoljno duboko i jasno opisati. S druge strane, za opisivanje njihovih atributa i međusobnih odnosa te za njihovo katalogiziranje potrebne su mogućnosti baza podataka. Ovo pomagalo ujedinjuje te dvije potrebe svojom integracijom s pomagalom za obradu teksta (*MS Word*) te integracijom s bazama podataka (*MS Access*, *MS SQL Server* ili *Oracle*).

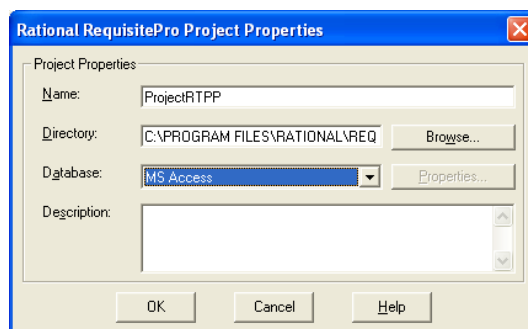
Pomagalo se pokreće odabirom Start → All Programs → Rational Software → Rational RequisitePro. Nakon nekog vremena pojavljuje se prozor Create Project s mogućnostima izbora predloška novog projekta ili otvaranja jednog od već postojećih projekata. Pri otvaranju novog projekta potrebno je označiti predložak *Use-Case Template* i odabrati OK (Slika 1.1).



Slika 1.1 Pokretanje pomagala

1.1. Projekti

Nakon toga se otvora prozor *Rational RequisitePro Project Properties* u kojem se unose redom ime projekta, odredišni direktorij, baza podataka koja će se koristiti te kratak opis projekta (Slika 1.2). Nakon toga treba odabrati OK.



Slika 1.2 Postavke novog projekta

Počinje stvaranje projekta nakon čega se dobiva obavijest *The RequisitePro project was successfully created* i potrebno je odabrati Close.

1.2. Vizija

U *Rational RequisitePro Explorer*-u na lijevoj strani radne površine potrebno je odabrati direktorij *Features and Visions* (otvoriti odabirom oznake + uz ime direktorija), te dvostrukim klikom na *Vision* otvoriti viziju projekta (otvara se u *MS Wordu*). U ponuđenom predlošku vizije potrebno je popuniti sljedeća poglavlja: *Project Name* (upisati dogovoreno ime programskog sustava), *Stakeholder and User Descriptions* (navesti korisnike sustava i interesne sudionike), *Product Overview* (max. dvije rečenice) i *Product Features* (ovdje upisati prepoznate i definirane zahtjeve).

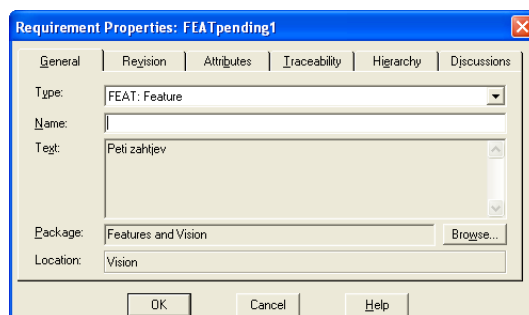
1.3. Zahtjevi

Kako bi se zahtjevi iz popunjene vizije upisali u pomagalo *RequisitePro* potrebno ih je u viziji jedan po jedan označiti („zacrnuti” željeni tekst) i odabrati ikonicu *New Requirement* na traci s alatima *Rational RequisitePro* u *Wordu* (Slika 1.3).



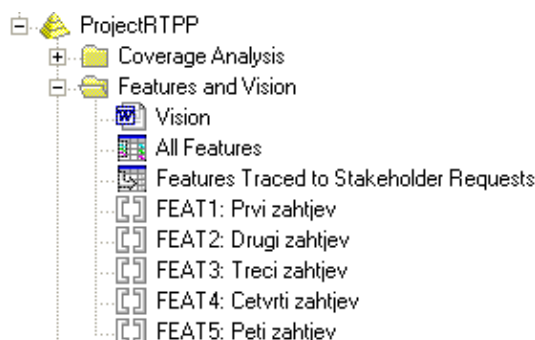
Slika 1.3 Traka s alatima

Za upis svakog novog zahtjeva (Slika 1.4) potrebno je odabrati vrstu zahtjeva (npr. *FEAT*, *STRQ* ili *SUPL*) i upisati ime, a od atributa odrediti prioritet i tip.



Slika 1.4 Novi zahtjev

Nakon toga obavezno je viziju pohraniti odabirom ikonice Save Requirements Document na traci s alatima prikazanoj na slici 2.3. Zatim će svi zahtjevi upisani u pomagalo i vidljivi u *Exploreru* na lijevoj strani (Slika 1.5).



Slika 1.5 Prikaz svih zahtjeva

1.4. Lokalno pohranjivanje vizije

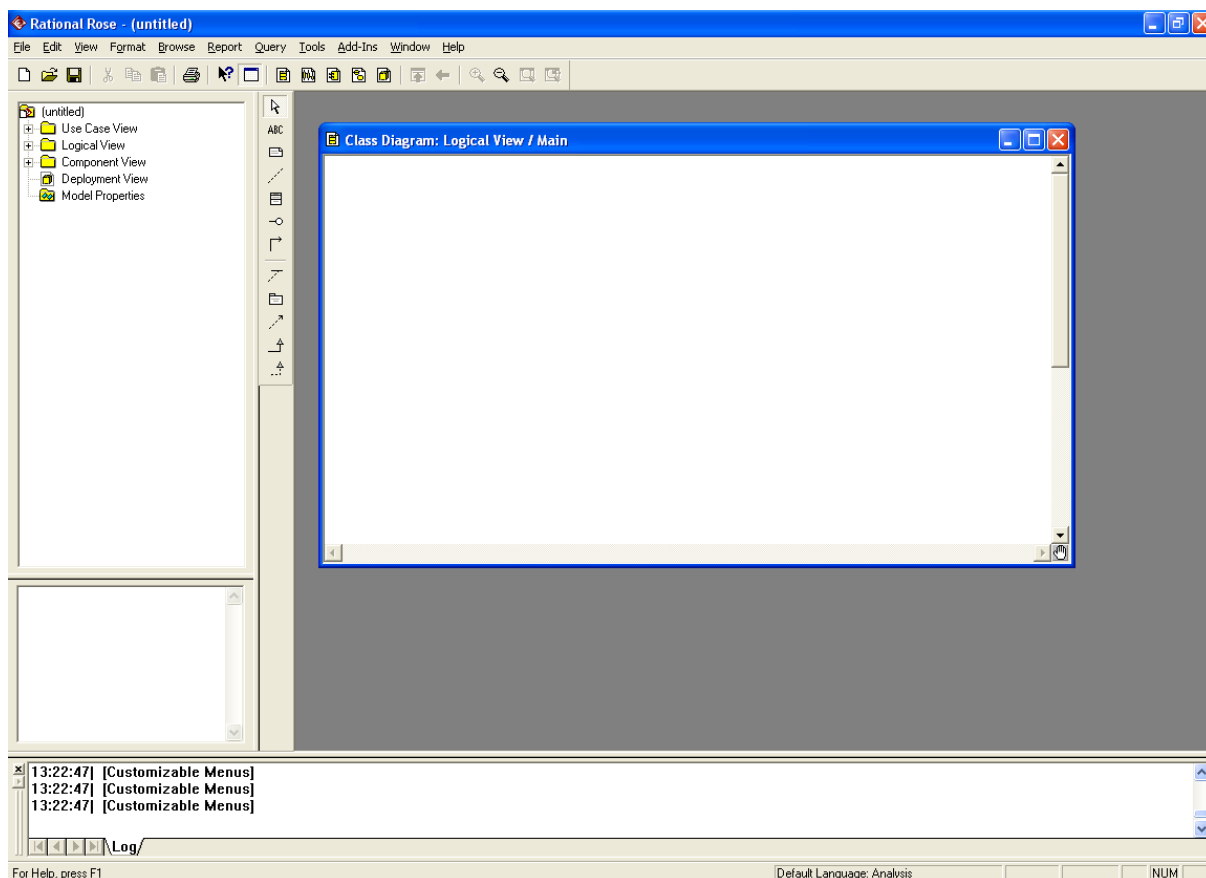
Kako bi se vizija izdvojila iz projekta i pohranila u željeni direktorij, u *Exploreru* je potrebno odabrati Features and Visions zatim desni klik miša na Vision pa odabrati Offline → Take Offline.... Pojavit će se prozor Take Offline Information u kojem treba odabrati korisnički direktorij u kojem će se pohraniti Vizija. Pohranjeni dokument ima nastavak .VIS i moguće ga je otvoriti u *MS Wordu*. Zatim ga se može pohraniti kao novi dokument s nastavkom .doc odabirom File → Save As....

2. Rose Enterprise

Rose Enterprise je pomagalo za dizajn objektno-orijentiranih sustava. Sustavi se opisuju jezikom UML (*Unified Modeling Language*) tj. UML dijagramima. Pomagalo može provjeriti ispravnost dijagrama, stvarati izvještaje, formalnu dokumentaciju, upravljati razvojem projekta te generirati kostur kôda na osnovi formalne specifikacije.

Pomagalo se pokreće odabirom Start → All Programs → Rational Software → Rational Rose Enterprise Edition. Nakon nekog vremena pojavljuje se prozor Create New Model s mogućnostima izbora predloška novog modela ili otvaranja jednog od već postojećih modela. Tu je potrebno odabrati Cancel.

Nakon toga se u pomagalu otvara osnovni prozor (Slika 2.1) u kojem se s lijeve strane nalazi organizacija projekta u različite poglede (*Use Case View*, *Logical View*, *Component View*, *Deployment View*, itd.).



Slika 2.1 Osnovni prozor

Potrebno je modelirati slučajeve uporabe za pojedine zahtjeve definirane pomagalom *RequisitePro*. To se postiže crtanjem dijagrama slučajeva uporabe (*Use Case Diagrams*) te, po potrebi slijednim (*Sequence Diagrams*) i/ili klasnim dijagramima (*Class Diagrams*). Nakon završetka korištenja pomagala model se pohranjuje u datoteku s nastavkom *.mdl* odabirom *File* → *Save*. Pomagalo se zatvara odabirom *File* → *Exit*.

2.1. Dijagram slučaja uporabe

Slučaj uporabe predstavlja slijed operacija koje izvodi sustav čiji rezultat akteru ima smisla. Njime se opisuje funkcija koju sustav nudi te komunikacija između aktera i sustava. Kako bi se dobro definirali slučajevi uporabe, potrebno je pokušati naći odgovore na sljedeća pitanja: Koji su zadaci svakog aktera? Da li neki akter stvara, sprema, mijenja, briše ili čita podatke iz sustava i koji su to slučajevi? Da li akter treba obavijestiti sustav o promjenama izvan sustava? Da li neki akter mora biti obaviješten o nekim događajima u sustavu? Koji slučajevi uporabe služe za održavanje sustava? Mogu li se svi zahtjevi sustava (što treba raditi) napraviti pomoću definiranih slučajeva uporabe?

Dijagrami slučaja uporabe u pomagalu trebaju biti smješteni u *Use Case View*. Diagram se može stvoriti tako da se desnim klikom odabere *Use Case View* → *New* → *Use Case Diagram*. Zatim se unese ime dijagrama i kad se pojavi u *Use Case View* može ga se dvostrukim klikom otvoriti.

Simboli dijagrama nalaze se u srednjem dijelu radne površine, između lijevo smještenog organizacijskog dijela s pogledima i desno smještenog dijela gdje su otvoreni dijagrami. Crtanje simbola izvodi se redom klikom miša na simbol i klikom na željeni dijagram. Zatim se unese ime (aktera ili slučaja uporabe).

Veze među simbolima se crtaju tako da se odabere veza i držanjem klika miša odvuče do odredišnog simbola. Ako je potrebno podesiti dodatne parametre, dvostrukim klikom miša na simbol ili vezu otvara se odgovarajući prozor.

Ako se želi napraviti relacija između aktera u slučaju uporabe koja je dvosmjerna onda se nakon crtanja relacije napravi desni klik na nju i odabere se *Navigable*. Ako se želi napraviti relacija koja ima razlomljenu liniju onda nakon držanja tipke miša pomikemo miš do točke gdje želimo prvi dio razlomljene relacije i tamo pustimo tipku. Svaki sljedeći klik će napraviti novi segment razlomljene relacije dok ne kliknemo na simbol aktera ili slučaja uporabe.

2.2. Klasni dijagram

Klasni dijagram opisuje tipove klase objekata u sustavu te statičke relacije između njih. Njime se prikazuju operacije i attribute klasa, a moguće ih je modelirati od razine koncepta do razine detalja dovoljne za kreiranje kôda.

U ovom pomagalu u klasnom dijagramu mogu se nalaziti i akteri. Oznaka za višestrukost „*“ je promijenjena u „n“. Kompoziciju nije moguće napraviti već se umjesto nje koristi agregacija. Kada se stavlja simbol klase onda se otvori dio u kojem možete odabrati već kreiranu klasu npr. aktera iz prethodnog dijagrama.

Da bi se definirali parametri relacije potrebno je napraviti dvostruki klik na relaciju i onda u otvorenom prozoru podesiti parametre. Višestrukost se definira u Role A Detail ili Role B Detail.

2.3. Slijedni dijagram

Slijedni dijagram opisuje dinamičku suradnju između objekata. Njime je prikazan vremenski slijed poruka koje objekti razmjenjuju te se pokazuje vrijeme života objekta, od stvaranja do uništavanja.

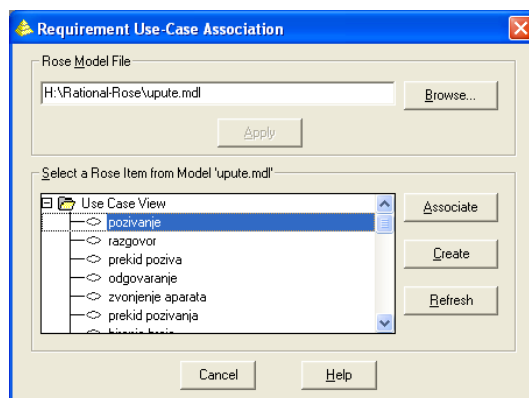
U slijednom dijagramu u ovom pomagalu se isto kao i u klasnom dijagramu može staviti akter koji predstavlja objekt. Ovaj dijagram se crta na sličan način kao i prijašnji dijagrami.

2.4. Kopiranje dijagrama

Trenutno aktivni dijagram moguće je kopirati desnim klikom na površinu dijagrama te odabirom Edit → Copy Active Diagram. Zatim ga treba kopiranje u odredišni Word dokument odabirom Edit → Paste (u Wordu).

2.5. Povezivanje zahtjeva i pripadnih slučajeva uporabe

Nakon modeliranja slučajeva uporabe potrebno je zahtjeve unesene u viziju povezati s odgovarajućim dijagramima slučajeva uporabe. Pojedini zahtjev u pomagalu *RequisitePro* treba najprije označiti, a zatim odabrati opciju Requirement → Associate to Rational Rose Use Case u *RequisitePro*-u.



Slika 3.2. Povezivanje zahtjeva i slučajeva uporabe

Zatim se u prozoru Requirement Use-Case Association (slika 2.6.) odabire postojeći *Rose* model i klikne Apply. U dijelu Select a Rose Item from Model '*model-name*' istog prozora najprije se označi ciljani slučaj uporabe, a zatim odabere Associate. U prozoru Resolve Use-Case Name potrebno je uskladiti imena zahtjeva i slučaja uporabe odabirom jednog od postojećih ili unošenja novog imena. Nakon uspješnog povezivanja pojavljuje se poruka *The requirement has been associated with the use-case 'use-case name'*.

Za otvaranje dijagrama povezanog sa zahtjevom u pomagalu *RequisitePro* potrebno je označiti zahtjev i odabrati Requirement → Open Rational Rose Use Case.

3. Literatura

- [1] Željka Car, *Razvoj telekomunikacijske programske podrške*, materijali s predavanja, FER-ZZT, 2005.
- [2] P. Kruchten, *The Rational Unified Process – An Introduction*, Addison-Wesley, 1999.
- [3] Mario Kušek, *UML – upute za rad u laboratoriju*, skripta iz kolegija *Informacija, logika, jezici*, FER-ZZT, travanj 2004.
- [4] Ghinwa Jalloul, *UML by Example*, Cambridge Press, 2004.
- [5] Specifikacija zahtjeva pomagalima Rational
www.msoe.edu/eecs/cese/resources/Rational/

Dodatak 1

Screenshot cijele radne površine se uzima pritiskom na tipku `Print Screen` kojom se slika pohranjuje u međuspremnik (*clipboard*). *Screenshot* trenutno aktivnog prozora se uzima kombinacijom tipki `Alt + Print Screen`. Zatim ga je moguće kopirati direktno u *Wordov* dokument odabirom `Edit → Paste`.

Dodatak 2: Model FURPS+

Prema RUP-u, zahtjevi se najčešće kategoriziraju modelom *FURPS+* (postoji i model *CRUPIC STMPL*) gdje je *FURPS* akronim sastavljen od sljedećih kategorija zahtjeva:

- funkcionalnost (*functionality*), koja obuhvaća specifikacije funkcijskih zahtjeva, sposobnosti i, ponekad, sigurnosti
- uporabivost (*usability*), koja može obuhvaćati specifikacije online pomoći, čarobnjaka za korisnike, korisničke dokumentacije, materijala za obuku
- pouzdanost (*reliability*), koja obuhvaća specifikacije vezane uz učestalost pada sustava, sposobnost oporavka, robustnost, *MTBF*, i sl.
- performanse (*performance*), koje postavljaju uvjete na funkcijske zahtjeve, poput brzine, točnosti, propusnosti, i sl.
- podrživost (*supportability*), koja može obuhvaćati specifikacije vezane uz ispitljivost, proširljivost, pogodnost za održavanje, i sl.

Znak "+" obuhvaća dodatne zahtjeve poput:

- ograničenja na dizajn (*design constraints*), koja se uvode na sam dizajn sustava,
- zahtjeva na implementaciju (*implementation requirements*), koji postavljaju ograničenja poput programskog jezika, korištenja traženih normi, resursa, politika, i sl.
- zahtjeva na sučelje (*interface requirements*), kojima se specificiraju vanjske entitete u interakciji sa sustavom, te ograničenja poput formata, vremena odaziva, i sl.
- fizičkih zahtjeva (*physical requirements*), koji se mogu koristiti kod specifikacije sklopovlja, npr. fizičke implementacije telekomunikacijske mreže.

Funkcijski zahtjevi određuju akcije koje sustav mora biti sposoban izvršiti, bez uzimanja u obzir fizičkih i drugih ograničenja. Najbolje ih je opisati modelima slučajeva uporabe i pripadnim dijagramima. Funkcijski zahtjevi određuju ulazno i izlazno ponašanje sustava.

Zahtjevi koji nisu funkcijski, a obuhvaćeni su ostatkom akronima *URPS+*, nazivaju se nefunkcijskima. Takvi zahtjevi opisuju samo attribute sustava ili njegove okoline.