

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike

Josip Marić

Optimizacija broja
djelatnika u domaćinstvu

JMBAG: 0303068646, redovit student
Kolegij: Modeliranje i Simulacija
Mentor: doc. dr. sc. Darko Etinger

Pula, lipanj, 2019.

SADRŽAJ

Uvod.....	1
Objašnjenje Simulacije	2
Objašnjenje Koda	3
Analiza Rezultata Simulacije	5
Analiza Rezultata Unaprijedene Simulacije.....	8
Preporuke Sustavu	9
Literatura	10

Uvod

Zbog zahtjeva određenih brendova te ugovora tvrtke sa zaposlenicima potrebno je uskladiti rad.

Riječ je o domaćinstvu u hotelu koje mora zbog zahtjeva nadležnog brenda minimizirati vrijeme čekanja na sobu, te u istom trenutku poštivati kolektivni ugovor zaposlenika koji zahtjeva da zaposlenik mora imati slobodan dan jednom u tjednu, odraditi posao uz minimalan broj zaposlenih i ne preći prag od 60 radnih sati na tjedan po zaposleniku. Također želimo izbjeći preopterećenost radnika.

Korištena simulacija je Discrete-Event a korištena biblioteka te jezik su simmer, R.

OBJAŠNJENJE SIMULACIJE

Naime simulacija je fokusirana samo na dio kako optimalno rasporediti sobarice po katovima kako bi se udovoljilo nadležnom brendu uz pravilo da po sobi uvijek rade minimalno dvije sobarice. U danom primjeru simulira se samo jedan kat zbog jednostavnosti te lakšeg razumijevanja, uključujući vrijeme odlaska i dolaska gostiju. Uz to zanemaruju se činjenice poput kojim gostima pripada koja soba, nego se podrazumijeva najbolji scenarij, odnosno da prva soba nad kojom se počelo čišćenje pripada prvoj skupini gostiju što je vrlo često slučaj i u praksi.

Navedeni ulazni podaci:

- Odlazak Gostiju

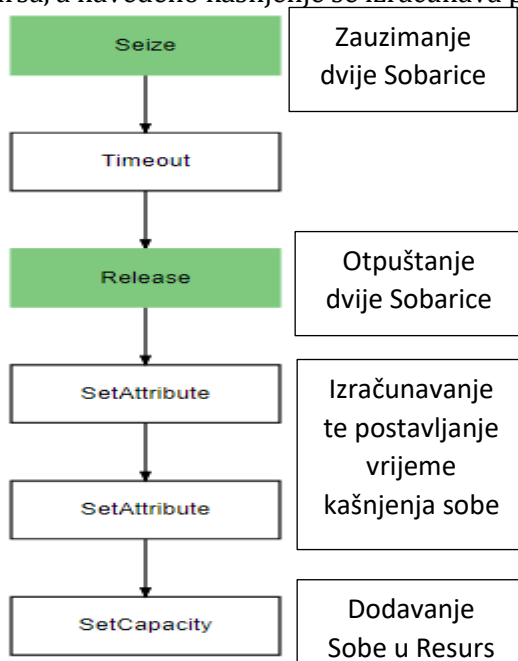
```
data.frame("time" = c(0,0,0,0,0,0,0,0,0,0,0,30,30,30,30,180,180,240,240))
```

- Dolazak Gostiju

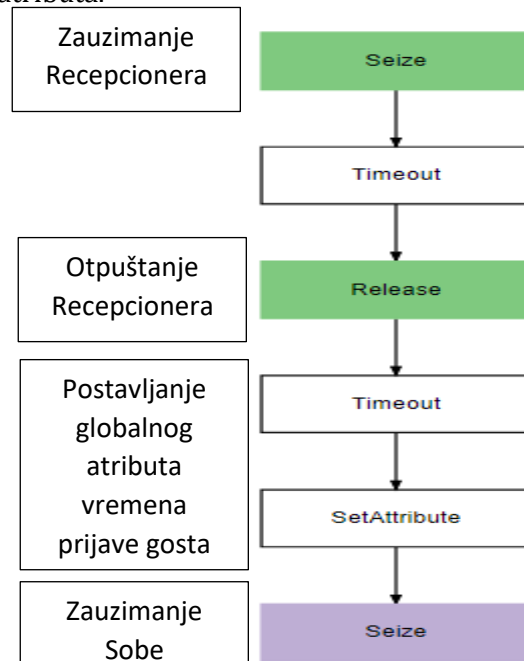
```
data.frame("time" = c(150,150,160,160,160,160,160,240,240,240,380,380,380,380,380,380,380,380))
```

U simulaciji nije uzet u obzir početak radnog vremena, već početak čišćenja soba, odnosno 08:00. Dio definiran sa nulama označuje da je gost napustio sobu prije toga ili u to vrijeme, ostale vrijednosti su izražene u minutama te označavaju koliko vremenski treba proći za njihovu „aktivaciju“. Svaki broj označava jednu sobu ili goste za jednu sobu.

Simulacija ima dva trajectory-a, goste i sobe. Posebnost kod soba je što se one koriste i kao resursi na način da nakon što su sobarice očistile sobu uz pomoć funkcije `set_capacity`, pribraja se resurs soba koju kasnije nakon što gost izvrši prijavu na recepciji zauzme (`seize`). Zbog tog prelaska nemoguće je izračunati `arrival_time` (koliko gost čeka za sobu) nego se soba zauzima iz resursa, a navedeno kašnjenje se izračunava preko atributa.



Slika 1: Soba <- trajectory



Slika 2: Gost <- trajectory

OBJAŠNJENJE KODA

Korištene biblioteke:

- Ggplot2
- Simmer
- Simmer.plot
- Parallel
- Grid
- gridExtra
- stringr

Soba Trajectory (put kojim se soba („arrival“) kreće)

```
1 env1 <- simmer("hotel")
2
3
4 soba <-
5   trajectory("Soba" path") %>%
6
7   seize("sobarica", 2) %>%
8   timeout(function() rnorm(1, 30, 10)) %>%
9   release("sobarica", 2) %>%
10  set_attribute(keys = "gotova_soba", value = function() now(env1))%>%
11  set_attribute(keys = "vrijeme_kasnjenja_sobe", value =
12    function() (now(env1) - get_global(env1, str_extract(get_name(env1),
13    "[:digit:]]+")))%>%
14  set_capacity(resource = "soba", value = 1, mod="+")
15
16
```

Slika 3: Soba Trajectory Kod

Na početku koda inicijaliziramo simmer pod nazivom hotel u varijablu env1 iz razloga jer nam treba kao atribut u funkcijama.

Nakon toga inicijaliziramo sobu sa osnovnim seize, timeout i release funkcijama te nakon toga definiramo dva atributa. Prvi atribut nam služi za provjeru i skupljanje podataka, a sastoji se od ključa „gotova_soba“ i vrijednosti koja se postavlja dinamički prilikom pokretanja simmera(env1), a postavlja se na trenutno vrijeme rada simulacije. Sljedeći atribut je nešto složeniji, sastoji se od ključa „vrijeme_kasnjenja_sobe“, a izračunava se na način da oduzmemo trenutno vrijeme rada simulacije, odnosno završetka čišćenja sobe sa globalnim atributom koji se postavlja u trajectory gost nakon njegove prijave na recepciji. Do točne vremenske vrijednosti, odnosno pravilo da prva očišćena soba pripada prvom gostu dolazimo na način da iz imena objekta („soba0“) izvučemo broj 0 te pod tim ključem će se nalaziti tražena vremenska vrijednost.

Na posljednjoj liniji pribrajam očišćenu sobu u resurs soba.

Gost Trajectory (put kojim se gost („arrival“) kreće)

```
1  gost <-
2    trajectory("Gosti' path") %>%
3
4    seize("recepционер", 1) %>%
5    timeout(function() rnorm(1, 5)) %>%
6    release("recepционер", 1) %>%
7    timeout(function() rnorm(1, 5)) %>%
8
9    set_global(keys = function() str_extract(get_name(env1),
10                                           "[[:digit:]]+"), value = function() now(env1)) %>%
11    seize("soba", 1)
12
```

Slika 4: Gost Trajectory Kod

Slično kao i prije imamo osnovne funkcije te dodatan timeout koji simulira trajanje puta od recepcije do sobe. Na 9 liniji postavljamo globalni atribut sa ključem kojeg izvučemo iz imena objekta („gost0“), a vrijednost postavljamo na vrijeme kada je gost izvršio prijavu na recepciji.

Env1 Simmer (okruženje simulacije)

```
1  env1 %>%
2    add_resource("sobarica", 4) %>%
3    add_resource("recepционер", 2) %>%
4    add_resource("soba", 0) %>%
5    add_dataframe("soba", soba, mon = 2, soba_vrijeme_pripreme, col_time = "time", time = "absolute") %>%
6    add_dataframe("gost", gost, mon = 2, gost_vrijeme_dolaska, col_time = "time", time = "absolute") %>%
7    run(until=480) # izvođenje simulacije 540 minuta; 9 sati
8
9  options(repr.plot.width=10, repr.plot.height=3, repr.plot.res=300)
10 par(mfrow=c(1,2))
11
12
13 env1_arrivals <- get_mon_arrivals(env1)
14 env1_resources <- get_mon_resources(env1)
15 env1_attributes <- get_mon_attributes(env1)
16
17 head(env1_arrivals)
18 head(env1_resources)
19
20
21 plot(env1_resources, metric = "usage", c("sobarica", "soba", "recepционер"),
22      items=c("queue", "server"))
23
24 plot(env1_resources, metric="utilization", c("sobarica", "recepционер"))
25
26 plot(env1_attributes, metric = waiting_time, keys = "vrijeme_kasnjenja_sobe")
27 ~
```

Slika 5: Env1 Simmer Kod

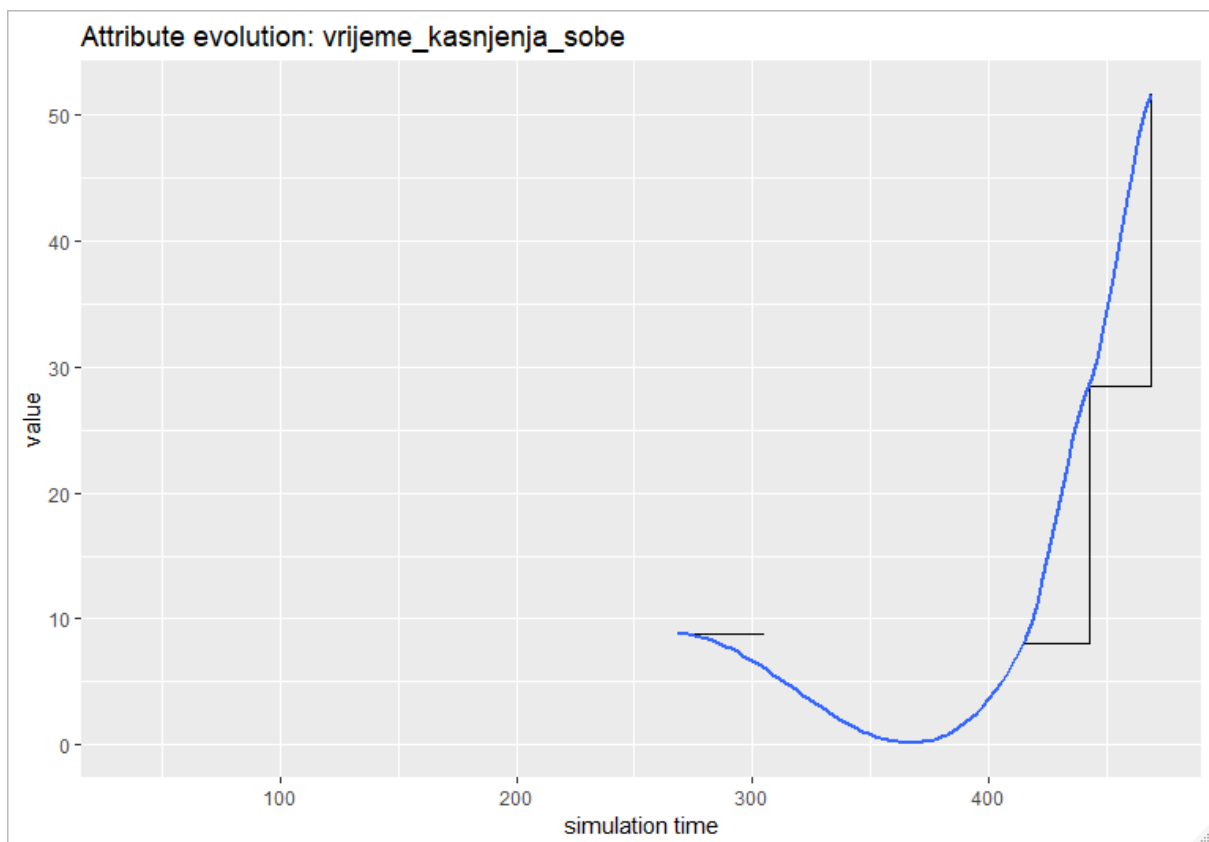
Inicijaliziran u prvom kodu, ovdje postavljamo način rada simulacije. Prvo dodamo 3 resursa, sobarice, recepcionar i soba. Soba je postavljena na 0 jer pristupamo problemu kao da je hotel punog kapaciteta što je vrlo često slučaj (ugovor sa brendom). Na 5 i 6 liniji dodajemo dataframe koji će generirati objekte soba i gost ovisno o ulaznim podacima. Monitoring je u oba slučaja postavljen na 2 iz razloga što kašnjenja zapisujemo u attribute. Simulacije se pokreće i traje 480min (radno vrijeme najkasnije završava u 16:00). U narednim linijama postavljamo opcije za plotting (generiranje grafova) i postavljamo arrivals, resources i attributes varijable. Na 21 liniji generiramo upotrebu resursa ovisno o sobarici, sobi i recepcioneru. Na 24 liniji generiramo iskorištenje resursa sobarica i recepcioner, te na posljednoj generiramo vrijeme čekanja sobe uz pomoć atributa uzeći u obzir jedino vrijednosti veće od 0 (vrijeme zakašnjenja).

ANALIZA REZULTATA SIMULACIJE

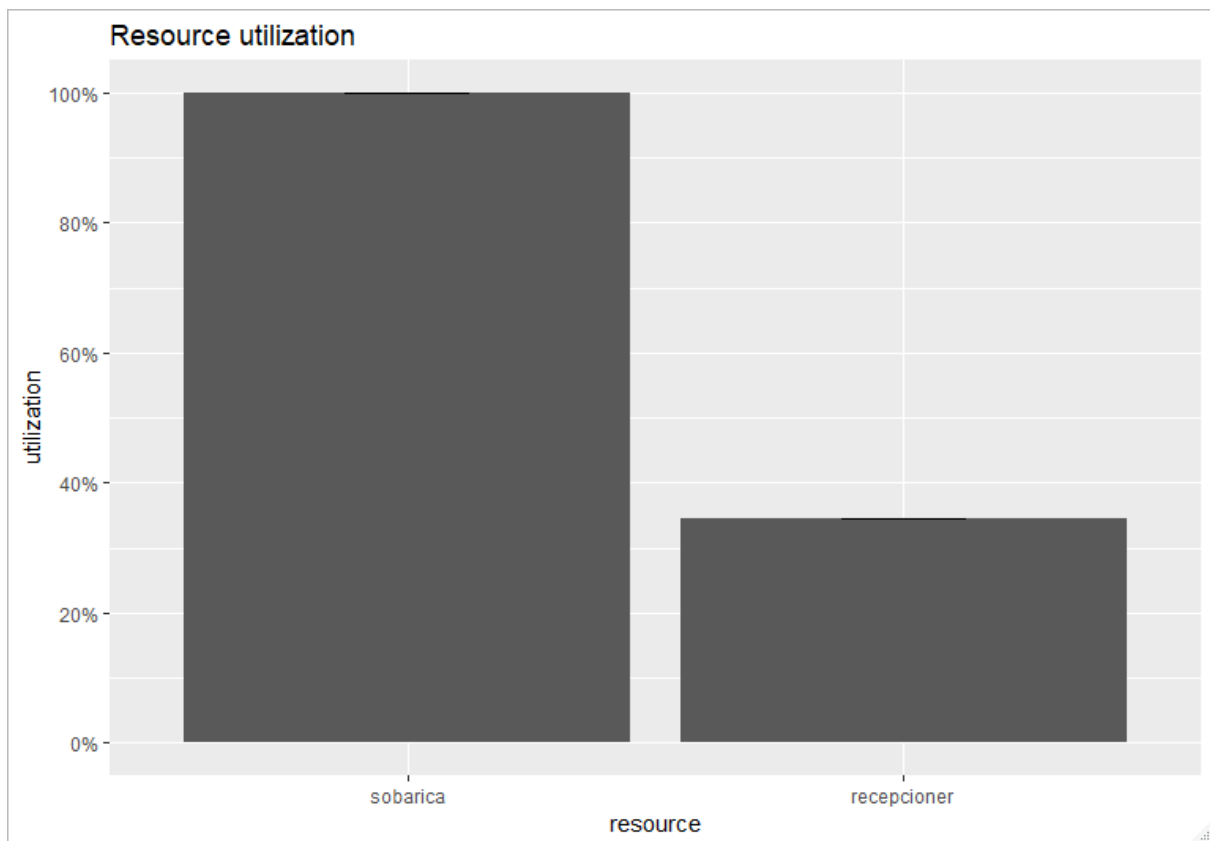
- Sveukupno imamo 18 odlazaka i dolazaka na definiranom katu (dolasci su realni)
- Na katu su pozicionirane dvije sobarice, a na recepciji jedan recepcionar
- Radno vrijeme nad sobama je od 08:00 do 16:00
- Vrijeme čekanja na sobu značajno raste pri kraju vremena, te ne stiže se očistiti svih 18 soba

Ciljevi Simulacije

- Minimizirati vrijeme čekanja na sobu
- Rasteretiti iskorištenost sobarica zbog nedostatka radne snage (u slučaju bolovanja zbog preopterećenosti)
- Osigurati nesmetan rad recepcije
- Završiti prije radnog vremena

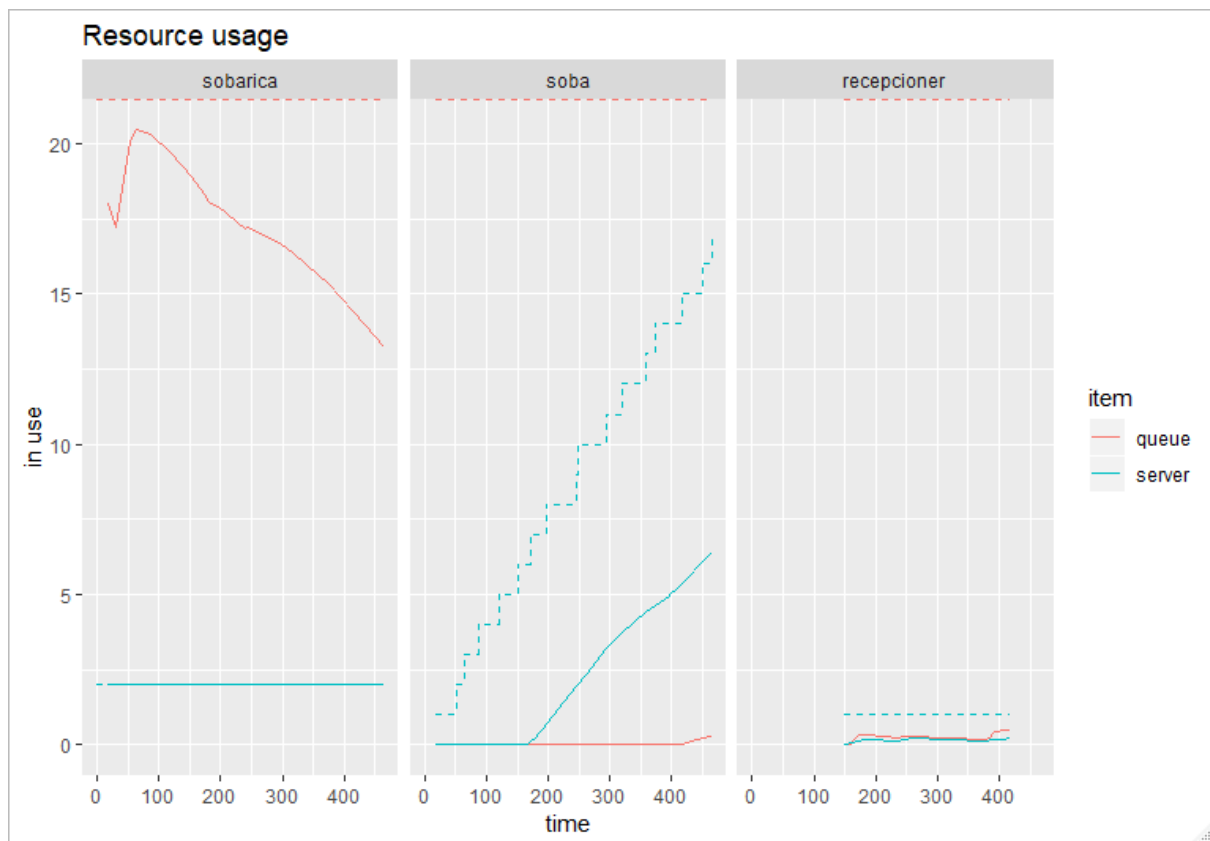


Slika 6: Vrijeme Čekanja na Sobi



Slika 7: Iskorištenost Resursa

Iz gornjeg grafa zaključujemo da nam nedostaje sobarica, iz razloga što je njihova iskorištenost 100% ali tu tvrdnju moramo potvrditi sa grafom upotrebe resursa.



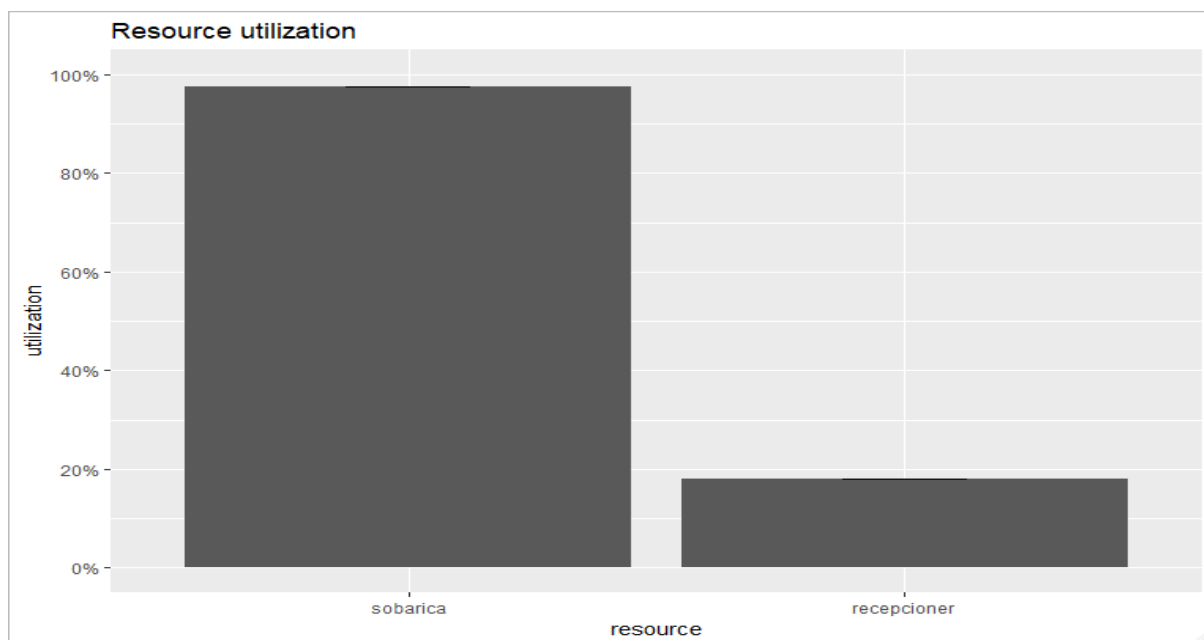
Slika 8: Upotreba Resursa

Sa posljednjeg grafa zaključujemo da nam doista nedostaje najmanje jedan tim sobarica, naime do kraja radnog vremena nisu uspjele očistiti niti pola soba. (Iz neidentificiranog razloga max queue i queue na grafu sobarice je postavljen iznad 20, a nebi smio biti iznad 18).

U navedenom primjeru dovoljan nam je jedan recepcioner ali zbog sigurnosti te neočekivanih zastoja dodati ćemo još jednog u sljedeću simulaciju.

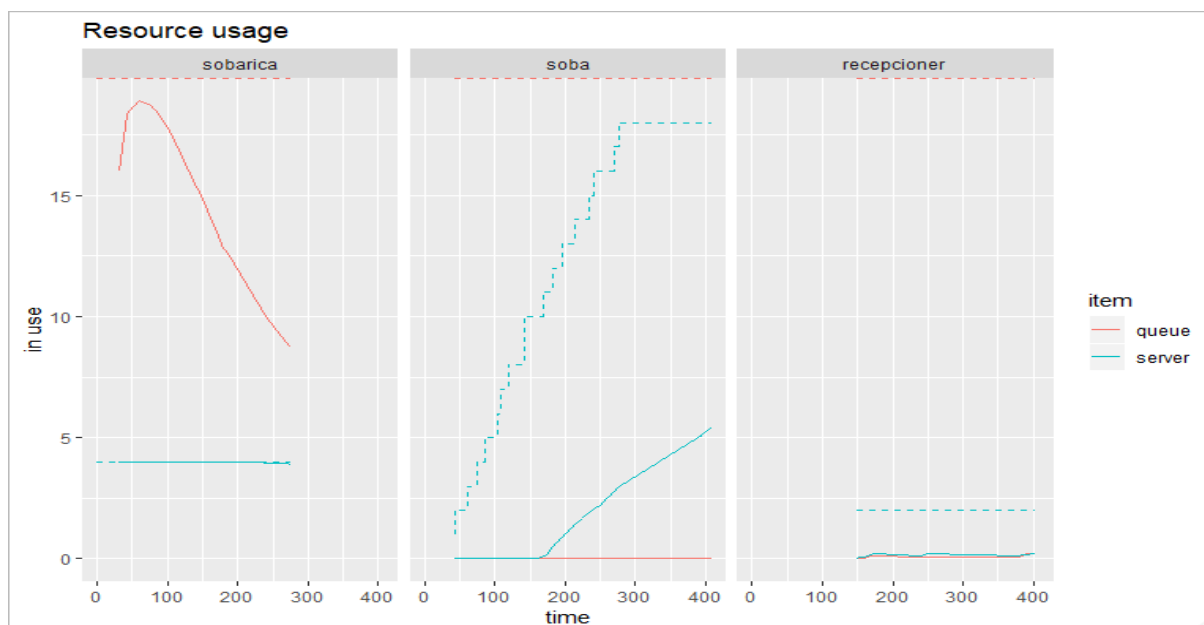
ANALIZA REZULTATA UNAPRIJEĐENE SIMULACIJE

U ovoj primjeru nema čekanja na sobu.



Slika 9: Iskorištenost Resursa

Iskorištenost sobarica je još uvijek visoka što nije dobar znak te će biti potreba većeg rasterećenja ukoliko moguće, ali osigurali smo nesmetan rad recepcionara.

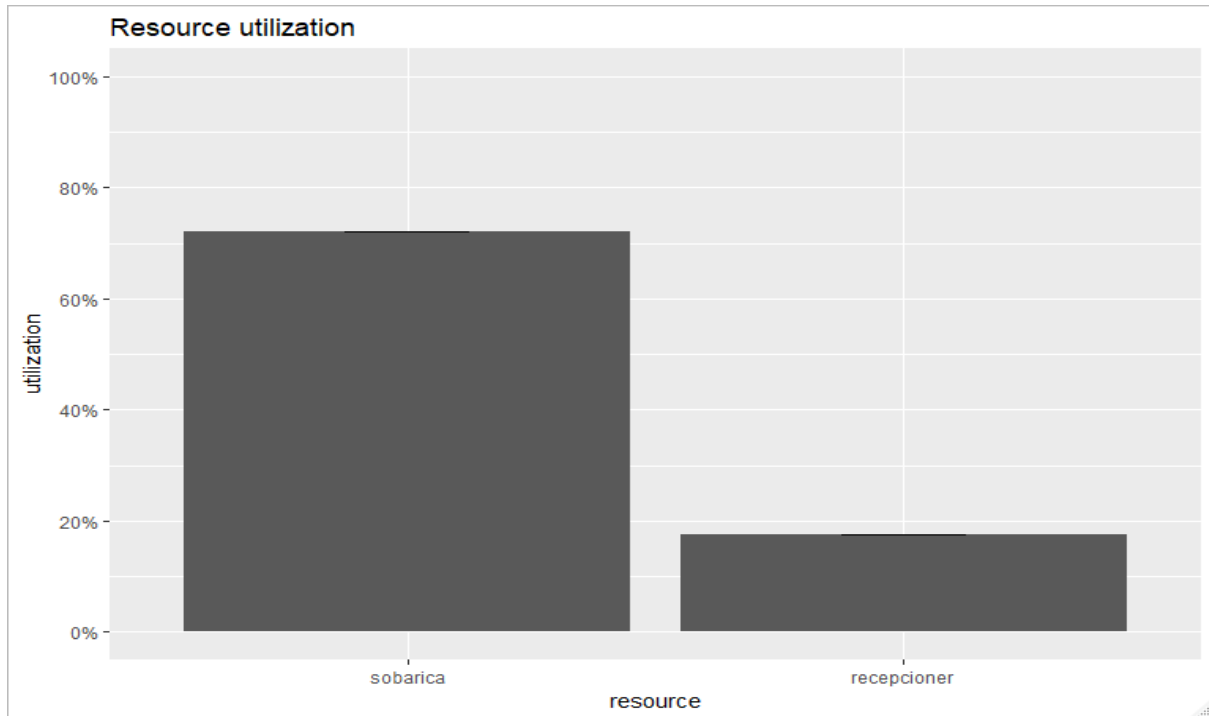


Slika 10: Upotreba Resursa

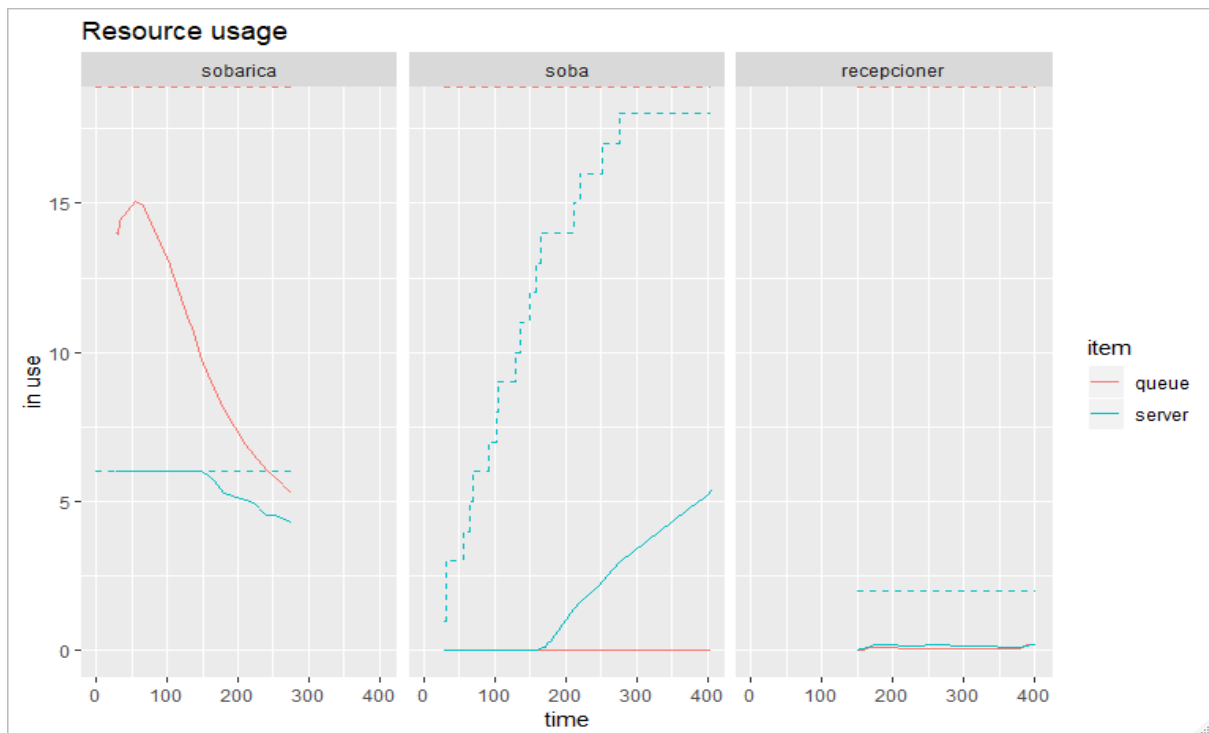
Iz sljedećeg grafa zadovoljili smo četvrtu crticu (završetak prije radnog vremena) što nam omogućuje dodatno čišćenje dnevnih soba.

PREPORUKE SUSTAVU

- Dodati najmanje 2 sobarice, ukoliko moguće 4 sobarice da se zadovolji rasterećenje te mogućnost čišćenja dnevnih soba
- Nije strogo potrebno ali zbog situacija koje nisu uključene u ovu simulaciju, preporuča se dodavanje još jednog recepcionara



Slika 11: Iskorištenost Resursa



Slika 12: Upotreba Resursa

TABLICA SLIKA

Slika 1: Soba <- trajectory	2
Slika 2: Gost <- trajectory.....	2
Slika 3: Soba Trajectory Kod.....	3
Slika 4: Gost Trajectory Kod	4
Slika 5: Env1 Simmer Kod.....	4
Slika 6: Vrijeme Čekanja na Sobu.....	5
Slika 7: Iskorištenost Resursa	6
Slika 8: Upotreba Resursa	7
Slika 9: Iskorištenost Resursa	8
Slika 10: Upotreba Resursa	8
Slika 11: Iskorištenost Resursa.....	9
Slika 12: Upotreba Resursa	9

LITERATURA

<https://www.dropbox.com/s/lke5bwfv4ybh1t2/DES-simmer-KLINIKA.html?dl=0>