



Upravljanje poslovnim procesima

6. Osobine poslovnih procesa

Katedra za informatiku
nastavnik: Zarić dr Miroslav



Sadržaj

- Uvod
- Međuzavisnost podataka
- Usklađenost sa životnim ciklusom objekata
- Strukturna ispravnost (korektnost)
- Ispravnost - Korektnost
- Relaksirani kriterijum korektnosti
- Slabi kriterijum korektnosti
- Lenji kriterijum korektnosti
- Pregled kriterijuma korektnosti



Uvod

- Razmatranje osobina modela poslovnog procesa je bitan aspekt upravljanja poslovnim procesom
- Ako neka osobina dokazano postoji u modelu tada će i sve instance procesa ispoljavati tu osobinu
- Strukturne zavisnosti
- Zavisnost podataka koji se procesiraju tokom izvršavanja procesa

Međuzavisnost podataka

- Aplikativni podaci – integralni deo poslovnog procesa
 - Mogu nastajati, biti menjani i brisani tokom izvršavanja procesa
- Dva aspekta:
 - Podaci nad kojima aktivnosti vrše neku manipulaciju (pozivanjem aplikacija ili servisa)
 - Međuzavisnost podataka između različitih aktivnosti u procesu

Podaci nad kojima aktivnosti vrše neku manipulaciju (pozivanjem aplikacija ili servisa)

- Ovaj aspekt se rešava na operativnom nivou
- Kod servisno orijentisnaih arhitektura parametri koji se koriste za pozivanje servisa i njihov tip moraju biti potpuno specificirani

Međuzavisnost podataka između različitih aktivnosti u procesu

- Rešava se na nivou procesa i tipično se opisuje tokovima podataka
- Aktivnosti procesa manipulišu podacima koji opisuju jedan slučaj obrade

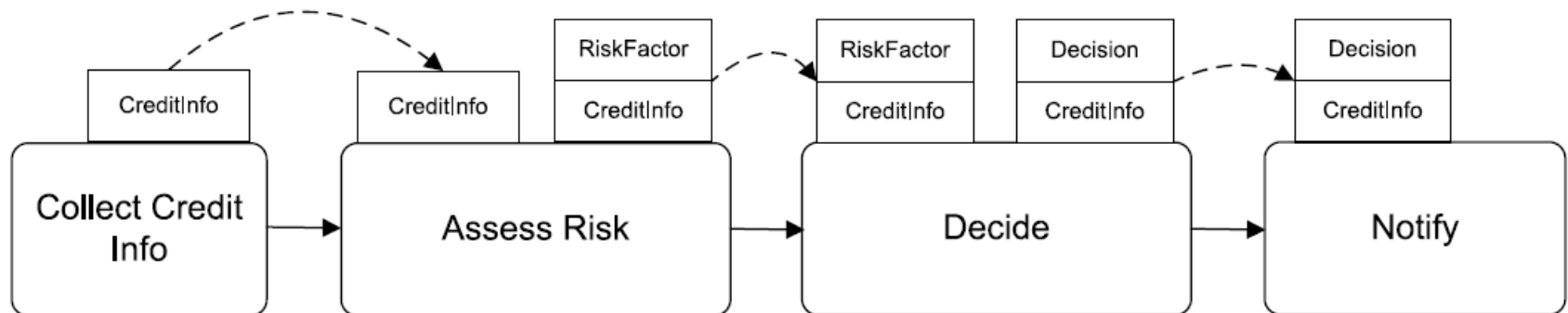


Fig. 6.1. Data dependencies imply activity orderings

Međuzavisnost podataka između različitih aktivnosti u procesu (2)

- U primeru, postoje međuzavisnosti između podataka koji se koriste u određenim aktivnostima
- Redosled aktivnosti je u ovom slučaju čvrsto vezan uz podatke koji se prosleđuju od jedne do druge aktivnosti
 - *AssesRisk* može biti aktivirana tek kada je dostupna informacija o zahtevanom kreditu – a taj podataka je izlazni parametar aktivnosti *Collect Credit Info*
- Međuzavisnost podataka je određena ovim tokovima podataka i kontrole toka moraju da prate ove tokove podataka

Međuzavisnost podataka između različitih aktivnosti u procesu (3)

- Ukoliko kontrola toka procesa ne prati tok podataka, instance procesa bi došle u blokirano stanje
 - Ova pretpostavka važi ako aktivnost zahteva ulazne parametre u momentu kada se pokreće
 - Ukoliko je moguće da aktivnost ulazne podatke dobije i nakon što je pokrenuta, ili da *Assess Risk* aktivnost izlazne podatke proizvede dok se još izvršava onda do *deadlock-a* neće doći

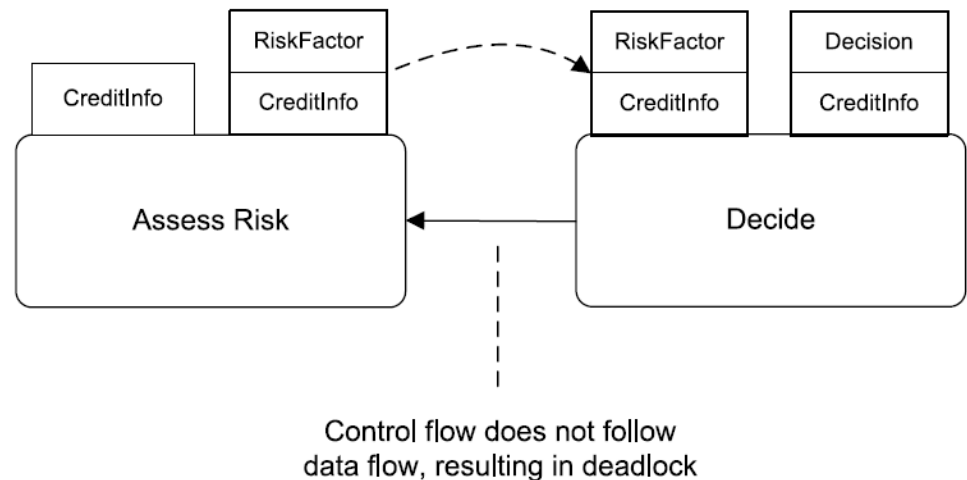


Fig. 6.2. Data flow violated by control flow, resulting in deadlock

Usklađenost sa životnim ciklusom objekata

- Svi poslovni procesi tokom izvršavanja generišu ili manipulišu nad određenim podacima
- Objektno orijentisani pristup omogućava ekspresivne metode za organizovanja objekata podataka, kako u pogledu strukture tako i u pogledu njihovog ponašanja
 - Klase, njihove veze, ponašanje opisano stanjem objekta i tranzicijama stanja (izvršavanjem metoda datog objekta)
 - Životni ciklus objekta

Životni ciklus objekata

- Primer: stanja objekta koji predstavlja predloženu cenu

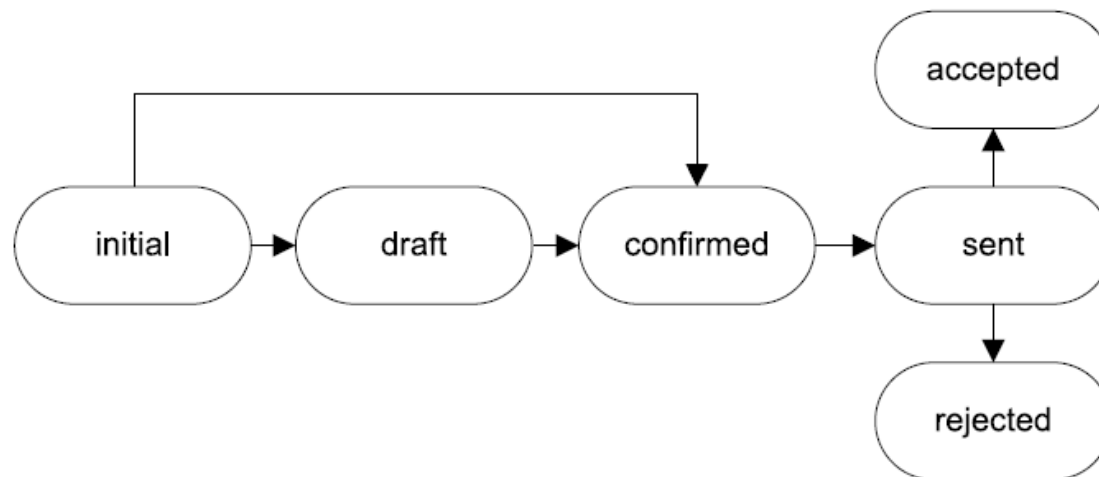


Fig. 6.3. Object lifecycle of quote object

- Prelazi stanja su inicirani aktivnostima u procesu

Životni ciklus objekata (2)

- Pošto model procesa definiše redosled aktivnosti, a aktivnosti izazivaju promenu stanja objekata, proces i životni ciklus objekata moraju biti usklađeni

Proces koristi samo one promene stanja objekta koje su definisane njegovim životnim ciklusom

object lifecycle conformant proces

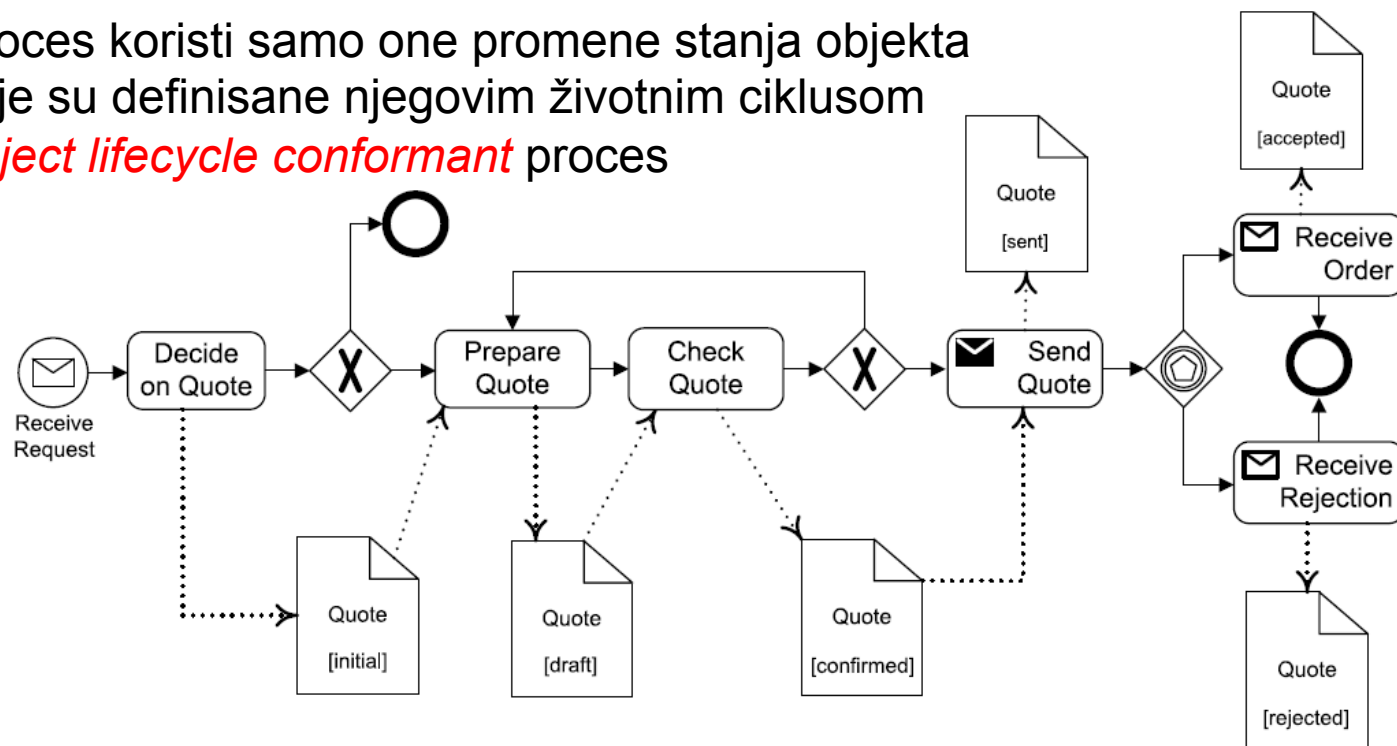


Fig. 6.4. Business process operating on a quote data object

Životni ciklus objekata (3)

- Proces koji nije usklađen sa životnim ciklusom objekta
 - Proces salje ponudu u *draft* stanju
 - Takav prelaz (iz *draft* u *sent*) nije definisan kao moguć za dati objekat

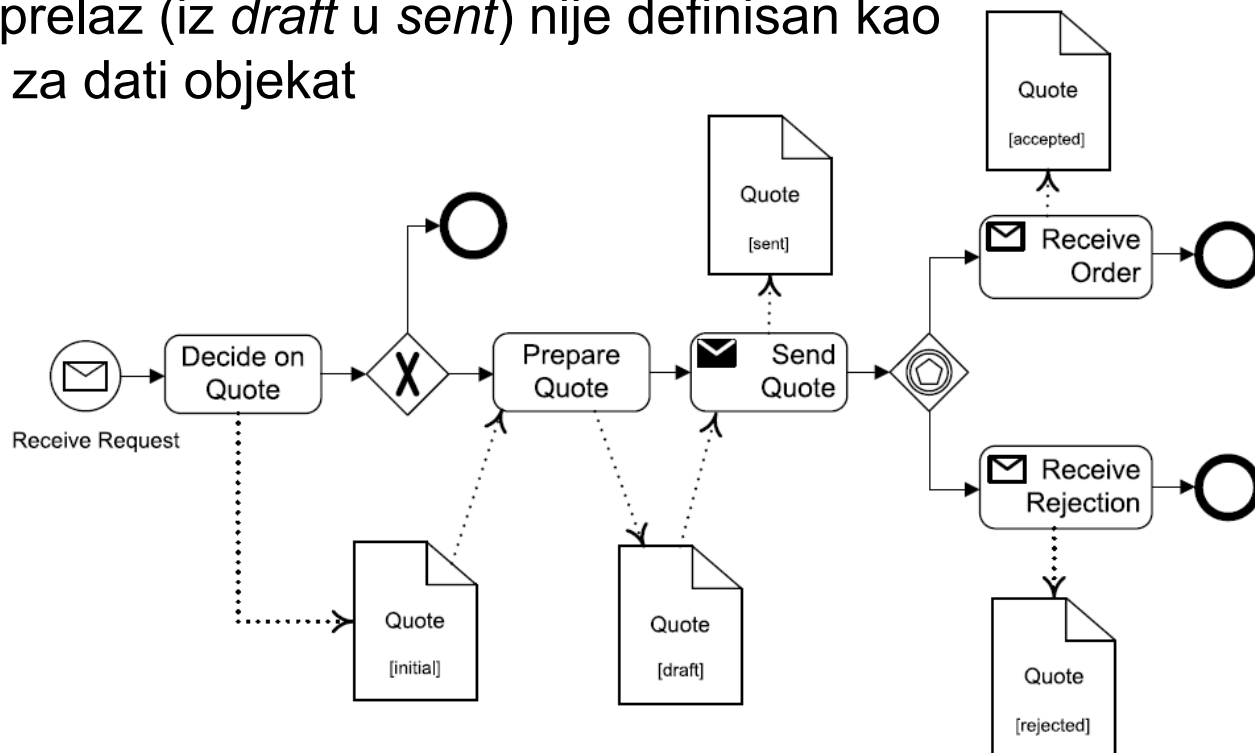


Fig. 6.5. Business process that does not conform to the quote object's lifecycle

Strukturna ispravnost (korektnost)

- Iako je primenjivo i na druge modele poslovnih procesa - koristimo Petri mreže (*Workflow* mreže) za utvrđivanje strukturnih grešaka – dozvoljavaju formalnu reprezentaciju strukturne korektnosti modela

Strukturna ispravnost (korektnost)

- Strukturna greška – “viseća” mesta ili tranzicije tj. tranzicije bez odgovarajućih ulaznih i izlaznih mesta u Petri mreži
 - Sledeća mreža je primer (nije *Workflow* mreža jer ima više od jednog mesta bez ulazne grane, i nisu svi čvorovi na putanji od *i* ka *o*)
 - Kada token stigne u mesto o *t5* bi se možda još izvršavala, tako da pristizanje tokena u *o* ne signalizira završetak obrade
 - Ovakve greške se izbegavaju kod *Workflow* mreža

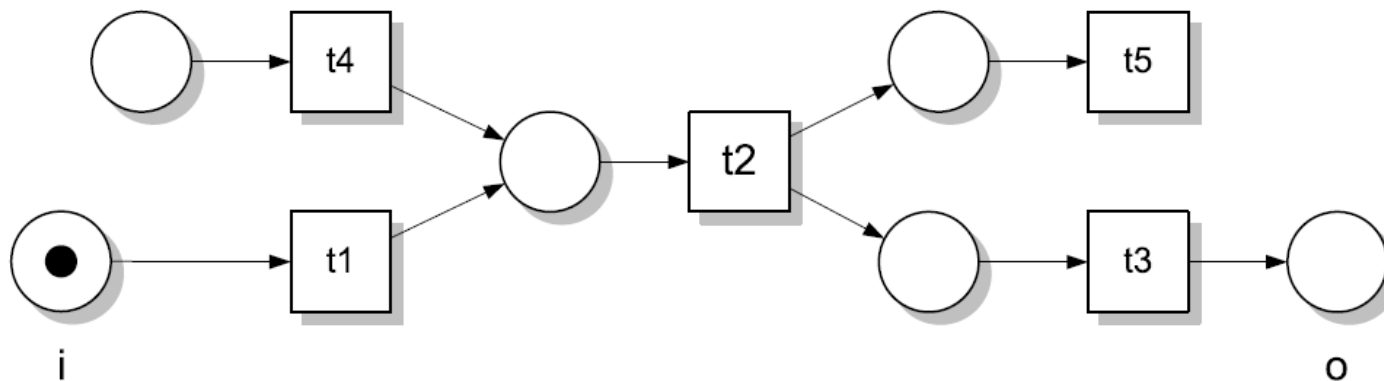


Fig. 6.6. Petri net with dangling places and dangling transitions

Strukturna ispravnost - definicija

- Model procesa je strukturno ispravan ako:
 - Postoji tačno jedan početni čvor, koji je jedini koji nema ulaznu granu
 - Postoji tačno jedan krajnji čvor, koji je jedini koji nema izlazne grane
 - Svi ostali čvorovi u mreži se nalaze na putanji od početnog ka krajnjem čvoru
- Većina jezika za modelovanje (i BPMN) forsiraju strukturnu ispravnost
 - Ipak dizajner modela može koristeći konstrukte jezika formirati model koji nije strukturno ispravan

Ispravnost

- Prvi kriterijum za procenu ispravnosti modela definisao Wil van der Aalst
- Motivacija za utvrđivanje ispravnosti modela
 - Primeri mreža sa greškama
 - *deadlock – xor split, and join* i
 - *livelock – beskonačne petlje*
 - mogu biti uzrokovane pogrešnim uslovima koji se uvek evaluiraju kao tačni, što nije moguće uvek utvrditi na osnovu modela procesa već može da se desi tokom izvršavanja
 - Pogrešna struktura procesa (u primeru ulazak u petlju je formiran *and* grananjem, umesto *xor*)
 - Ni izlazno mesto više ne signalizira stvarni kraj procesa

Ispravnost – primeri grešaka

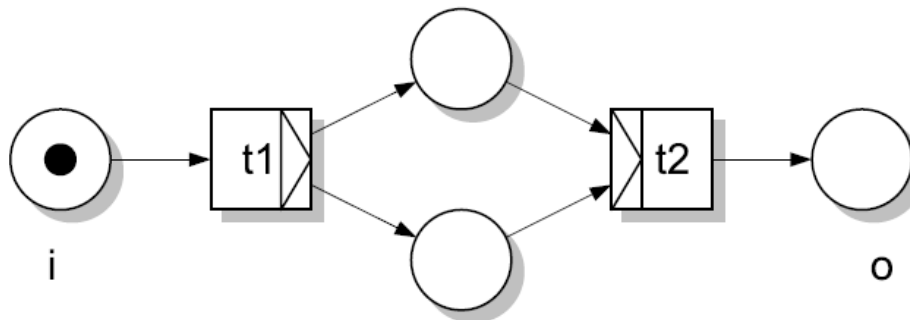


Fig. 6.7. Workflow net with deadlock

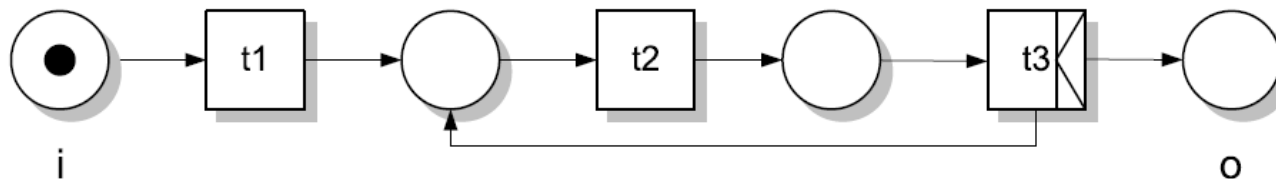


Fig. 6.8. Workflow net with livelock

Ispravnost – primeri grešaka (2)

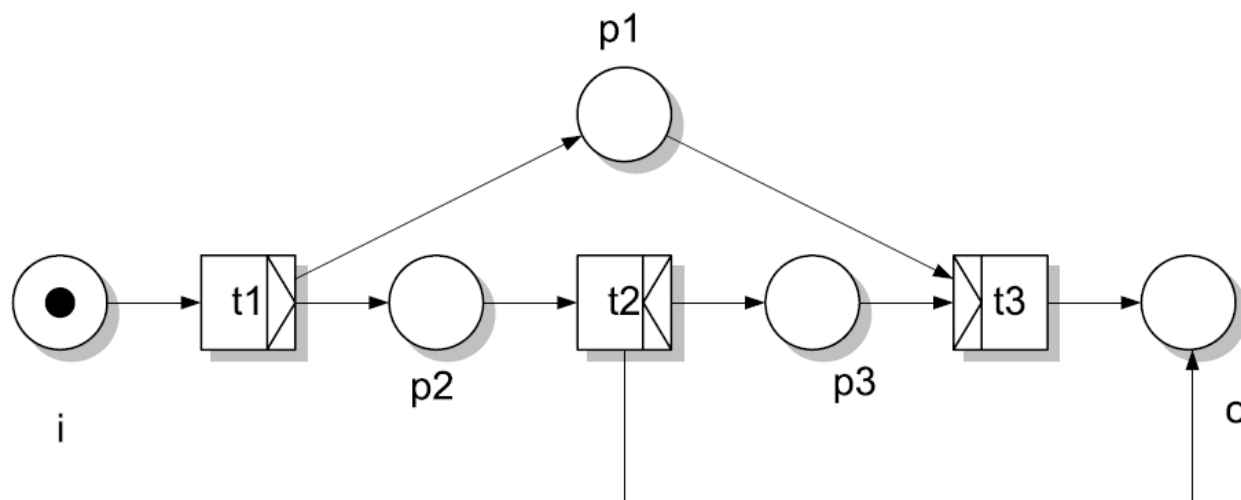


Fig. 6.9. Workflow net with deadlock/remaining tokens

- Ako *t1* postavi token u *p1*, ali ne i u *p2*, tada *and* spajanje *t3* nikada ne može biti omogućeno – deadlock
- Ako *t1* postavi token u *p2*, tada se *t2* može “okinuti”, i postaviti token u izlazno mesto *o*, a u tom momentu postoji token i u *p3*

Ispravnost – definicija

- Na osnovu prethodnih primera, može se definisati kriterijum ispravnosti
- Definicija stanja *workflow* mreže: Neka je $PN=(P,N,F)$ *workflow* mreža, $i \in P$ njeno početno mesto, $o \in P$ njeno krajnje mesto, i M, M' oznake stanja
 - $[i]$ je stanje mreže u kome postoji tačno jedan token u mestu i i nijedno drugo mesto ne sadrži nijedan token
 - $[o]$ je stanje mreže u kome postoji tačno jedan token u mestu o i nijedno drugo mesto ne sadrži nijedan token
 - $M \geq M'$ ako i samo ako je $M(p) \geq M'(p), \forall p \in P$
 - $M > M'$ ako i samo ako je $M \geq M' \wedge \exists p \in P: M(p) > M'(p)$

Ispravnost – definicija

- Definicija ispravnosti: *Workflow* sistem (PN, i) sa *workflow* mrežom $PN=(P,N,F)$, je ispravan ako i samo ako
 - Za svako stanje M dostupno iz stanja $[i]$ postoji sekvenca “okidanja” koja vodi od M do $[o]$

$$\forall M([i] \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} [o])$$
 - Stanje $[o]$ je jedino stanje dostižno iz $[i]$ kada je bar jedan token u mestu o

$$\forall M([i] \xrightarrow{*} M \wedge M \geq [o]) \Rightarrow (M = [o])$$
 - Ne postoje “mrtve” tranzicije u stanju mreže $[i]$

$$(\forall t \in T) \exists M, M': [i] \xrightarrow{*} M \xrightarrow{t} M'$$

Analiza dostupnosti

- Može se koristiti da se utvrdi da li je posmatrana *wroflow* mreža ispravna
 - Stanja i prelazi stanja se eksplicitno predstavljaju
 - Graf dostupnosti se koristi da prikaže različita stanja u kojima se graf može naći
 - Sastoji se od čvorova i označenih grana
 - Čvorovi predstavljaju stanja, a grane tranzicije stanja

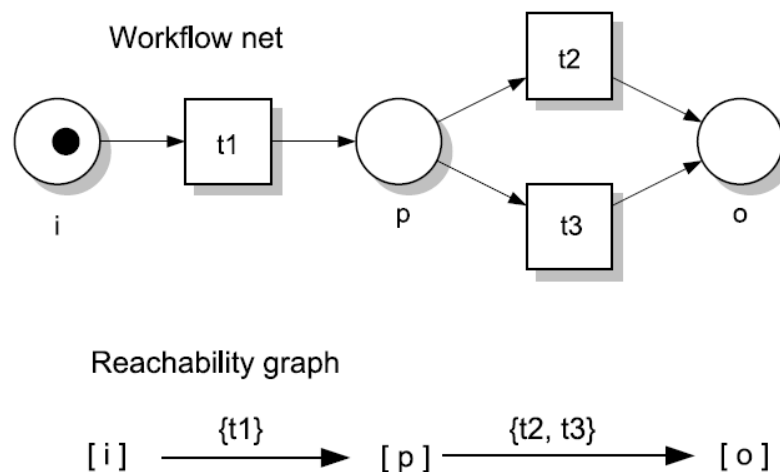


Fig. 6.10. Workflow net and corresponding reachability graph

Teorema korektnosti

- Analiza dostupnosti na osnovu crtanja grafa je pogodna za relativno male modele
- Veliki modeli su nepogodni jer se javlja “eksplozija stanja”
- Postoje računarski potpomognuti sistemi koji omogućavaju da se utvrdi korektnost
 - Automatsko generisanje grafova dostupnosti – zbog “eksplozije stanja” vreme izvršavanja eksponencijalno raste sa složenošću modela – nepogodno za implementaciju
 - Korišćenje nekog od alata za Petri mreže kako bi se na osnovu njih proverila korektnost – bazira se na teoremi kojom se uspostavlja veza između korektne *Workflow* mreže i “životnosti” i “ograničenosti” Petri mreže

Teorema korektnosti (2)

- Osnovna ideja je da se od *workflow* mreže izvede Petri mreža dodavanjem tranzicije t^* koja spaja kranje mesto o sa početnim mestom i

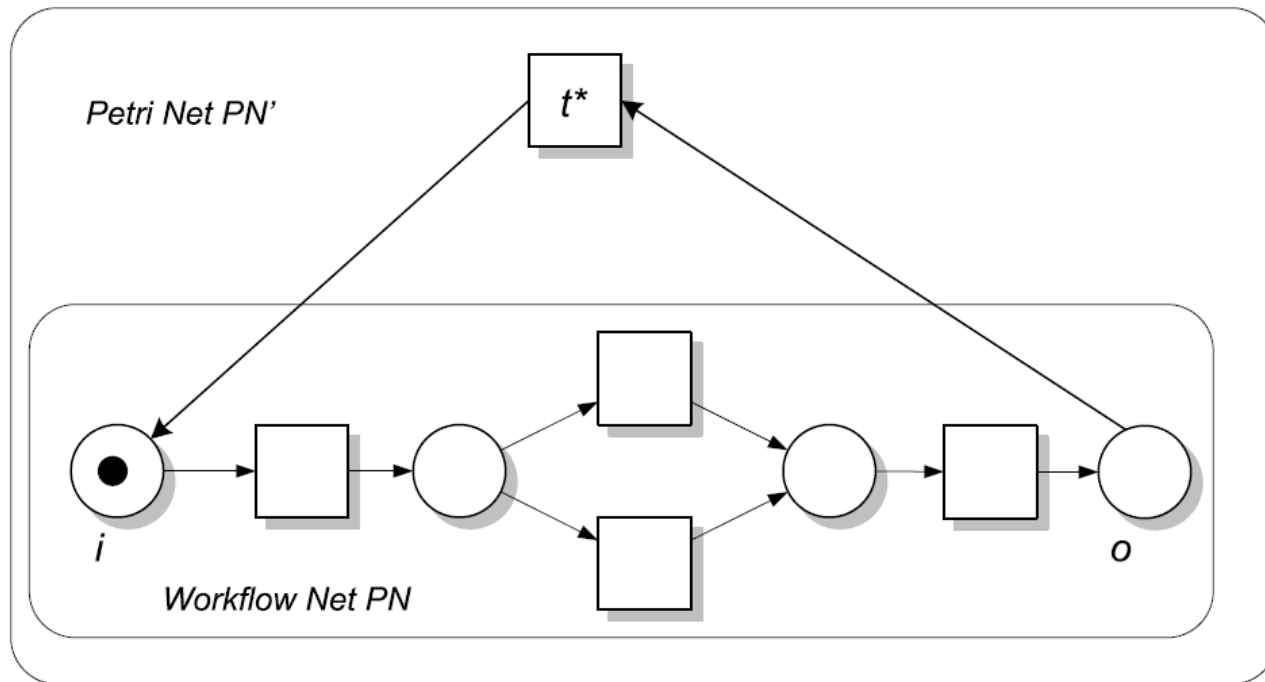


Fig. 6.13. Workflow net PN and Petri net PN' , illustrating soundness theorem

Teorema korektnosti (4)

- Teorema: Neka je $PN = (P, T, F)$ *workflow* mreža i $t^* \notin T$. PN je korektna ako i samo ako je (PN', i) takvo da je Petri mreža $PN' = (P', T', F')$ takva da je $P' = P$, $T' = T \cup t^*$ i $F' = F \cup \{(o, t^*), (t^*, i)\}$ i ako je ona “životna” i “ograničena”
 - Kako je PN' živa, za svaku tranziciju postoji sekvenca “opaljivanja” koja počinje u početnom stanju $[i]$, a koja dovodi do njenog aktiviranja. Ovo posebno važi za dodatnu tranziciju t^* , kojoj je jedino ulazno mesto o a ono je dostupno iz inicijalnog stanja.
 - Kada se desi tranzicija t^* token se smešta u i . Ponovo se može utvrditi sekvenca opaljivanja koja dovodi do o . Mreža je i ograničena jer bi se inače tokeni nagomilavali u nekom mestu Petri mreže.

Teorema korektnosti (5)

- Dokaz da ako je PN korektna onda je i PN' ograničena se izvodi kontradikcijom
 - Pretpostavimo da PN' nije ograničena
 - Onda mogu da postoje stanja mreže M i M' takva da postoje proizvoljne tranzicije koje menjaju stanje $i \rightarrow M \rightarrow M'$ i da je $M' > M$, što omogućava agregaciju tokena u nekom mestu Petri mreže
 - Kako je PN korektna *workflow* mreža onda postoji sigurno sekvenca “opaljivanja” koja stanje M prevodi u stanje $[o]$
 - Ako se ista sekvenca primeni na stanje M' ona će mrežu dovesti u stanje $M'' > [o]$ – što faktički znači da postoji token u o , ali istovremeno i u nekom drugom mestu u mreži – što je u suprotnosti sa pretpostavkom korektnosti mreže PN.
- Za velike mreže algoritmi i dalje pokazuju eksponencijalnu složenost, ali postoji podskup mreža za koje je rešenje polinomijalno – mreže slobodnog izbora

Mreže slobodnog izbora

- Petri mreža $PN=(P,T,F)$ je mreža slobodnog izbora ako i samo ako za $t_1, t_2 \in T$ važi ili $\bullet t_1 = \bullet t_2$ ili $\bullet t_1 \cap \bullet t_2 = \emptyset$

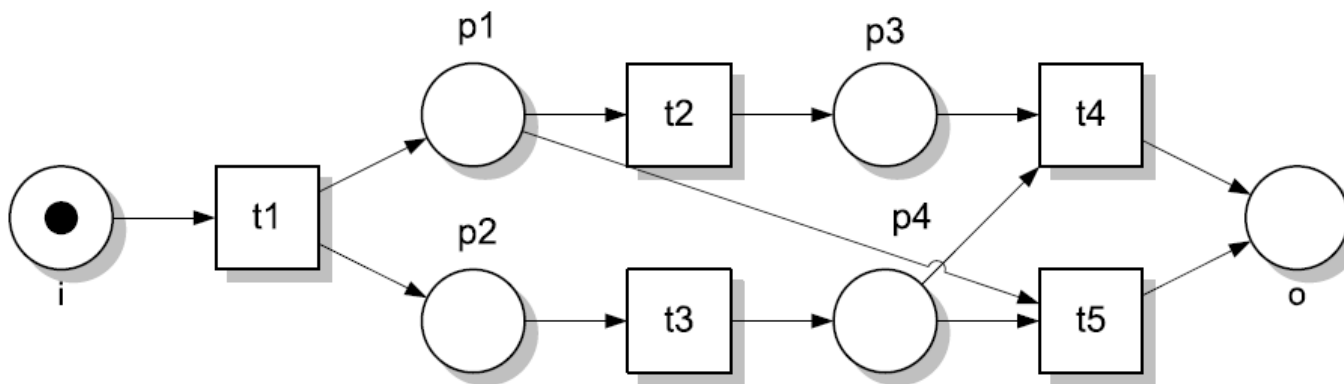


Fig. 6.14. Non-free-choice workflow net

Mreže slobodnog izbora (2)

- Mreže koje ne zadovoljavaju ovaj uslov nisu poželjne kao model poslovnog procesa jer ponašanje sistema zavisi od redosleda izvršavanja u konkurentskim granama
 - Ako se t_2 desi pre t_3 onda se t_5 ne može nikada desiti jer token u p_1 više nije na raspolaganju
 - Ako se t_3 desi pre t_2 onda postoji mogućnost da se desi i t_5 – što znači da ne postoji sloboda izbora za izvršavanje t_4 i t_5 već oni direktno zavise od redosleda kojim su se izvršili t_2 i t_3

Relaksirani kriterijum korektnosti

- Za neke poluformalne metode modelovanja, kod kojih je naglasak na prihvatljivoj reprezentaciji procesa, kriterijum korektnosti može biti pretvrd
- Kriterijum relaksirane korektnosti uzima u obzir ove potrebe, ali da pri tome ipak postoji formalni način proveriti modela

Primer: prijem robe

- Jednostavan procesni lanac pokretan događajima
- Dobro opisuje semantiku procesa, ali ima potencijal za neželjeno izvršavanje
 - Ako xor uđe u levu granu, a roba je OK, nikada neće biti ispunjen uslov za and join, i proces je blokiran
 - Ako se desi da je xor izabrao desnu granu, a roba je loša, proces će se obaviti, ali se neće evidentirati žalba

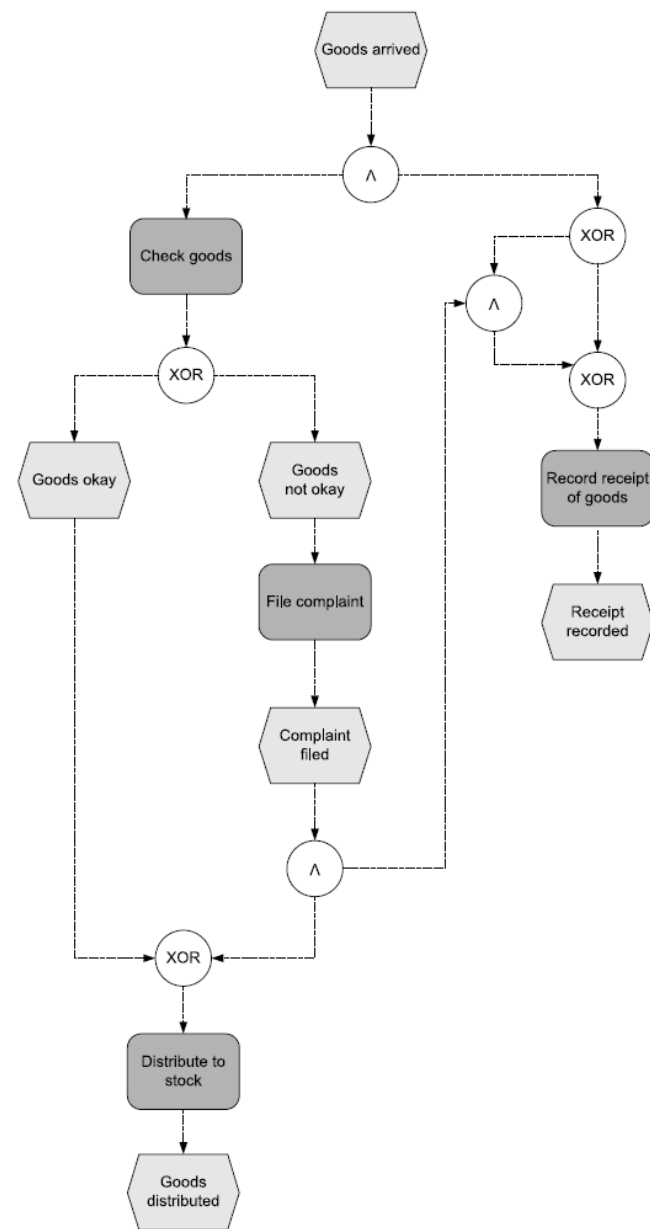


Fig. 6.16. Typical example of an event-driven process chain

Osnovna ideja

- Model procesa je prihvatljiv ako dozvoljava da se izvrše instance procesa koje dovode do željenog ishoda
- Formalno se ova procesna mreža može prevesti u Petri mrežu u tri faze
 - Generisanje modula Petri mreže
 - Kombinovanje modula u Petri mrežu
 - Dodavanje početnog mesta (izvora) i ishodišta mreže – Petri mreža se transformiše u *workflow* mrežu

Prevođenje procesnih lanaca u Petri mrežu

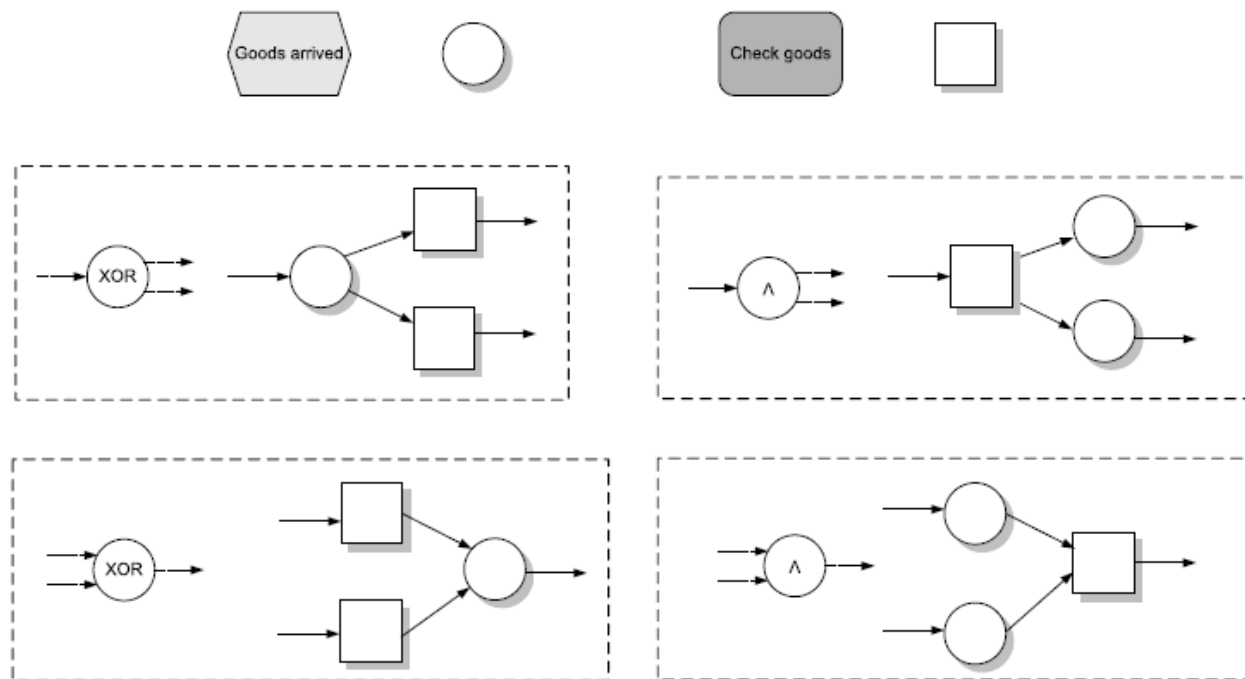


Fig. 6.17. Translation of event-driven process chain into a Petri net

Prevođenje procesnih lanaca u Petri mrežu (2)

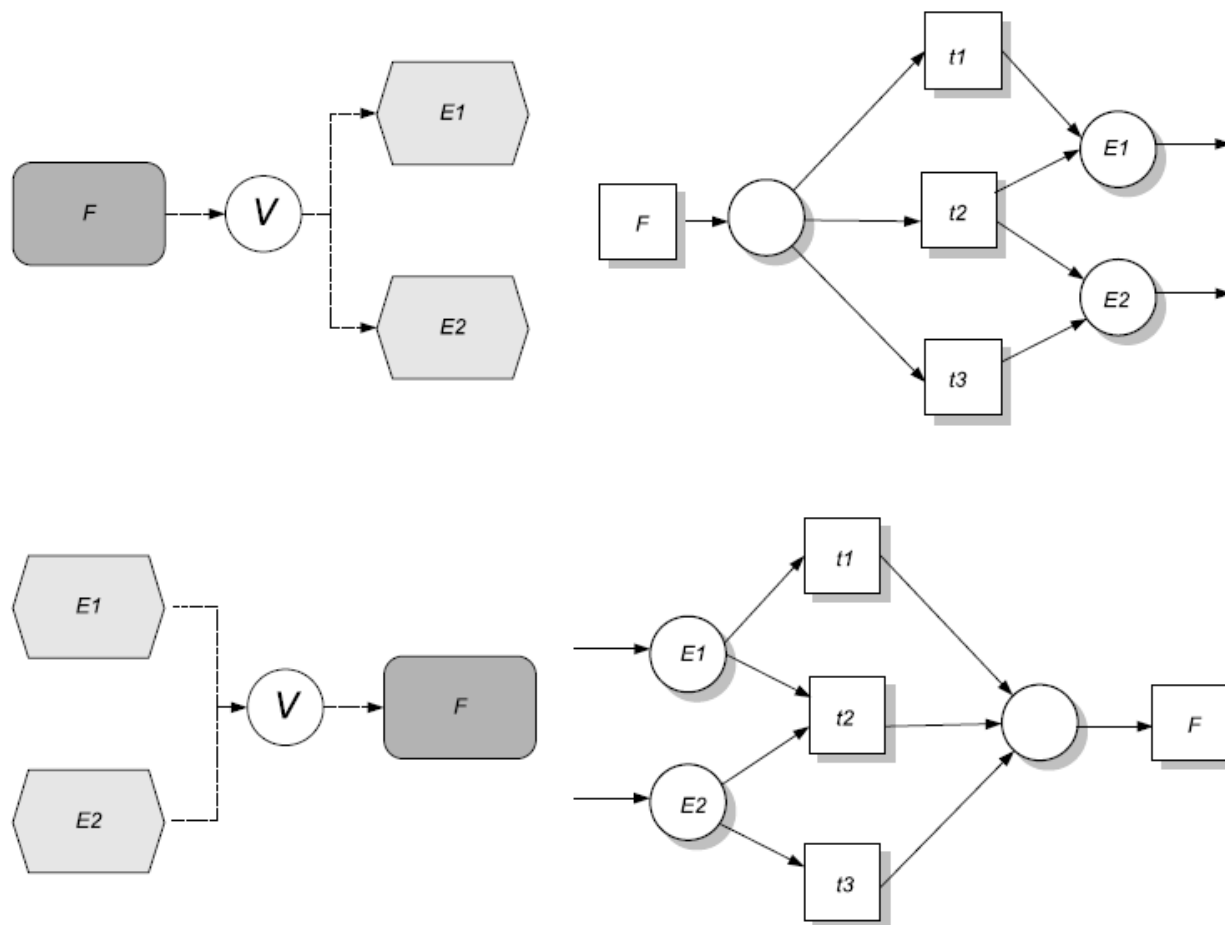


Fig. 6.18. Mapping of event-driven process chain *or connector* to workflow net

Prevođenje procesnih lanaca u Petri mrežu (3)

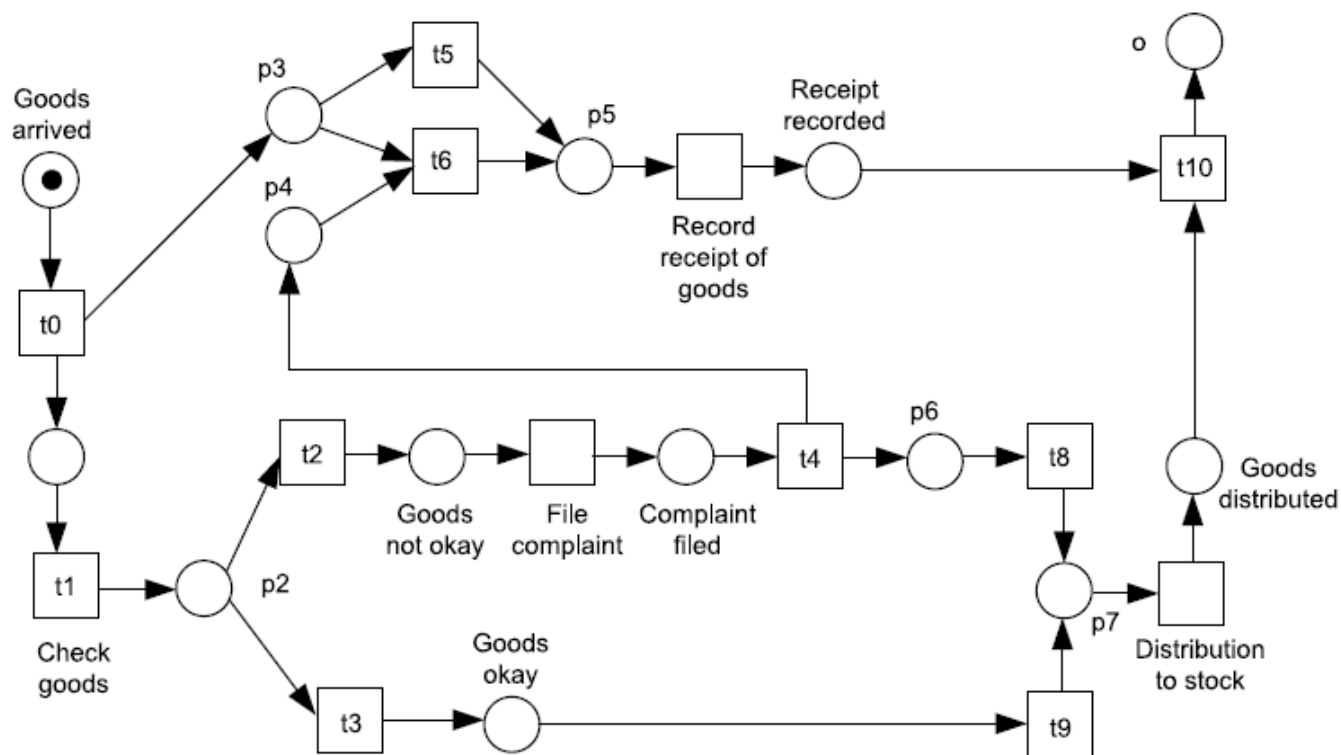


Fig. 6.19. Relaxed sound workflow net

Prevođenje procesnih lanaca u Petri mrežu (4)

- Rezultujuća mreža nije mreža slobodnog izbora (tranzicije t_5 i t_6 dele isto ulazno mesto, ali t_6 ima i dodatno)
 - Zbog toga se može desiti neželjeno ponašanje
 - Neke instance procesa mogu završiti tako da naruše korektnost mreže
 - Druge instance mogu se završiti očekivanim ishodom

Relaksirani kriterijum korektnosti

- Za svaku tranziciju postoji sekvenca “okidanja” koja je sadrži, a koja konačno dovodi do željenog ishoda procesa
- Uvodi se pojam *korektne sekvence okidanja*
 - **Def:** Neka je $S=(PN, i)$ *workflow* sistem. Neka su σ i σ' sekvence okidanja, a M i M' stanja mreže. σ je *korektna sekvenca okidanja* ako vodi do stanja mreže iz koga je moguće dostići konačno stanje $[o]$ tj.
 $[i] \xrightarrow{\sigma} M$ i $\exists \sigma'$ takvo da važi $M \xrightarrow{\sigma'} [o]$

Relaksirani kriterijum korektnosti

- Definicija: *workflow* sistem $S=(PN, i)$ je korektan u skladu sa relaksiranim kriterijumom ako i samo ako je svaka tranzicija PN element neke od *korektnih sekvenci okidanja*

$$\forall t \in T \exists M, M' : ([i] \xrightarrow{*} M \xrightarrow{t} M' \xrightarrow{*} [o])$$

Slabi kriterijum korektnosti

■ Razvijen u kontekstu koreografija

- Pretpostavimo da se svaka orkestracija u okviru koreografije reprezentuje kao web servis
- Svaki web servis ispoljava određeno konverzaciono ponašanje
- Pretpostavka da će sve funkcionalnosti koje web servis nudi biti i korišćene je isuviše rigidna
- Moguće je da koreografija procesa ispoljava željeno ponašanje, a da se ne koriste sve mogućnosti web servisa
- Ovakvo ponašanje je slaba korektnost

Slabi kriterijum korektnosti

- **Def:** *workflow* sistem (PN, i) zadovoljava slab kriterijum korektnosti ako i samo ako važi:
 - Za svako stanje mreže M koje je dostižno iz početnog stanja $[i]$, postoji sekvenca okidanja koja od M vodi do konačnog stanja $[o]$
 - Stanje mreže $[o]$ je jedino stanje dostižno iz stanja $[i]$, takvo da je barem jedan token u mestu o

Spajanje modula po interfejsima: agencija i putnik

Sintaksno kompatibilni moduli
Spajanjem nastaje mreža u
kojoj su zajednička
komunikaciona mesta postala
interna, a preostala formiraju
interfejs ka drugim sistemima

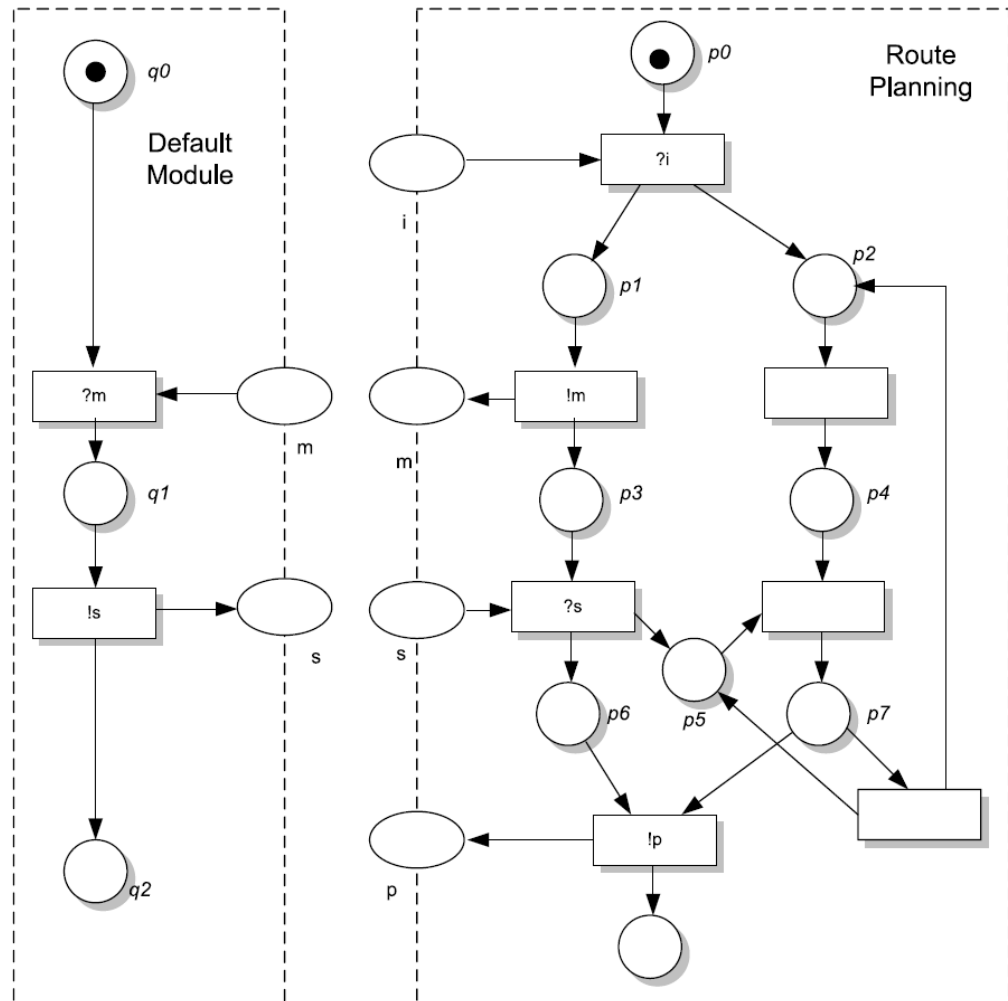


Fig. 6.20. Workflow module with syntactically compatible default module

Spajanje modula po interfejsima: agencija i putnik

Ako se moduli mogu tako komponovati da ne preostaju neupareni interfejsi onda se može uspostaviti distribuirani proces

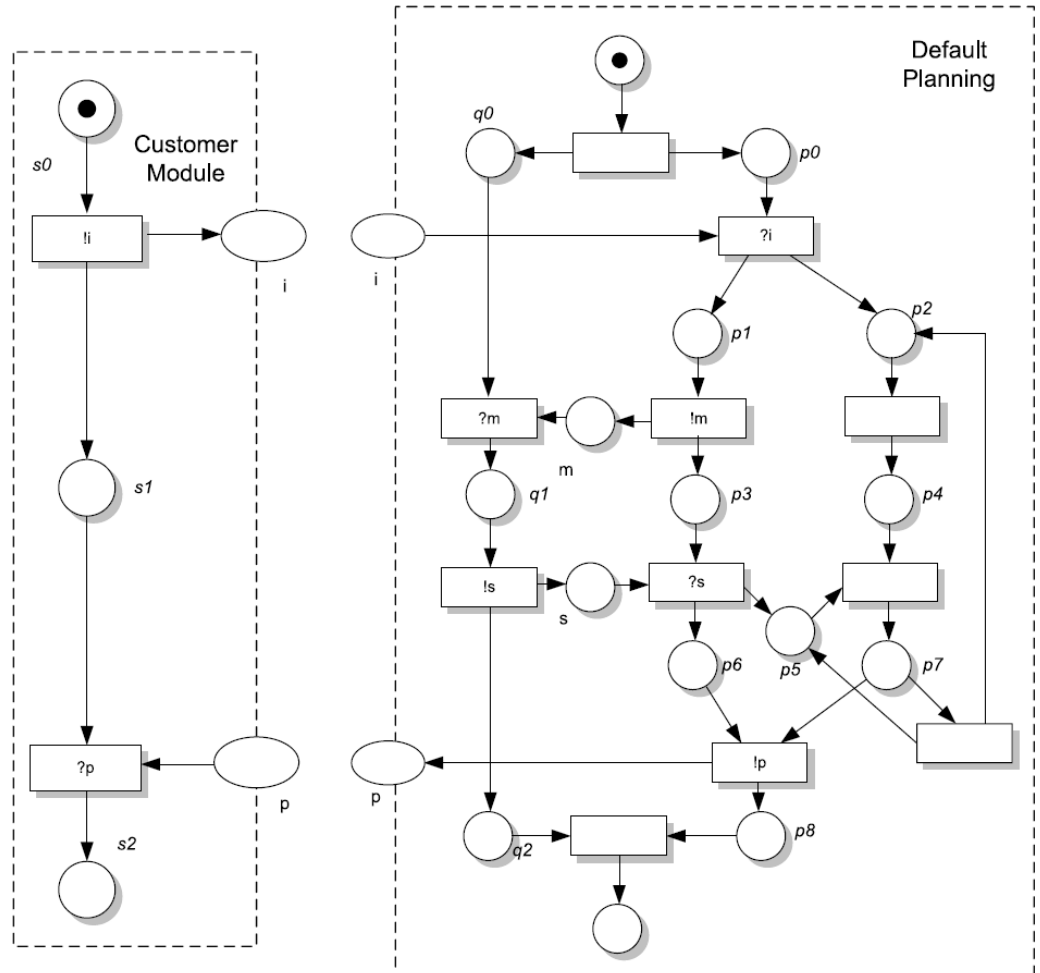


Fig. 6.21. Workflow module default planning and customer workflow module

Spajanje modula po interfejsima: agencija i putnik

Model nastao spajanjem tri
modula

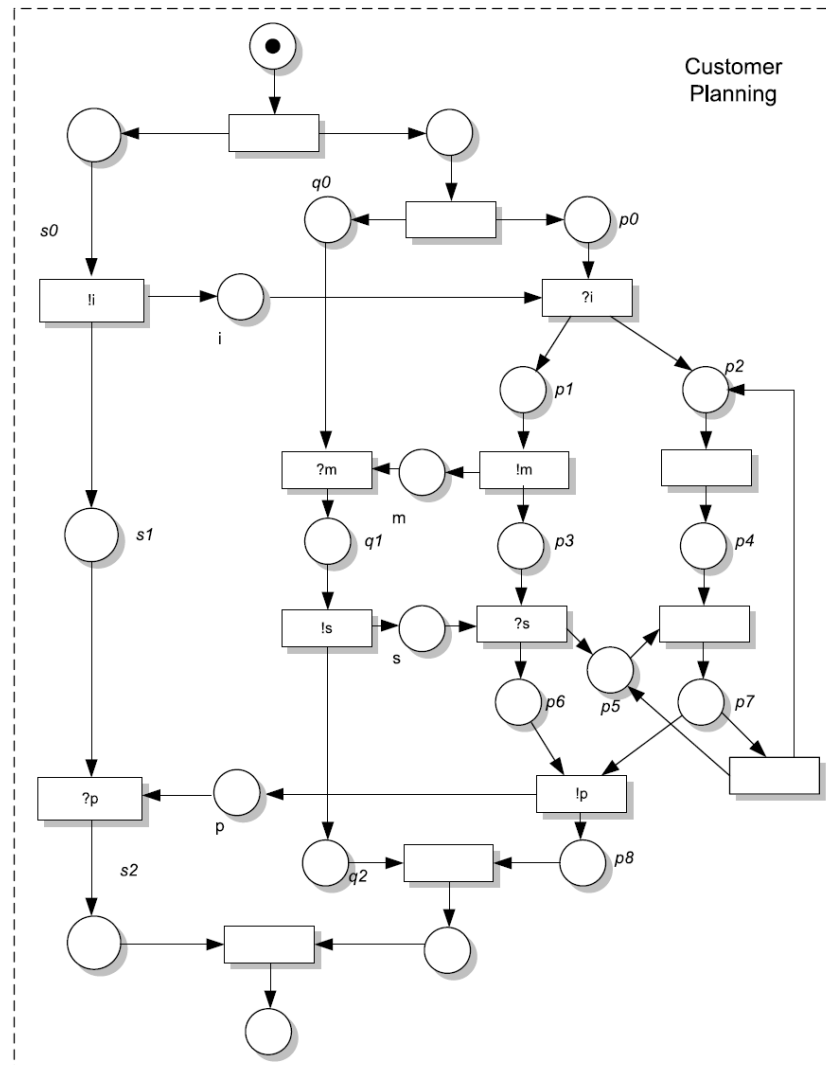


Fig. 6.22. Distributed business process composed of three workflow modules

Primer:

Ukoliko se apstrahuju komunikaciona mesta svaki modul je korektan

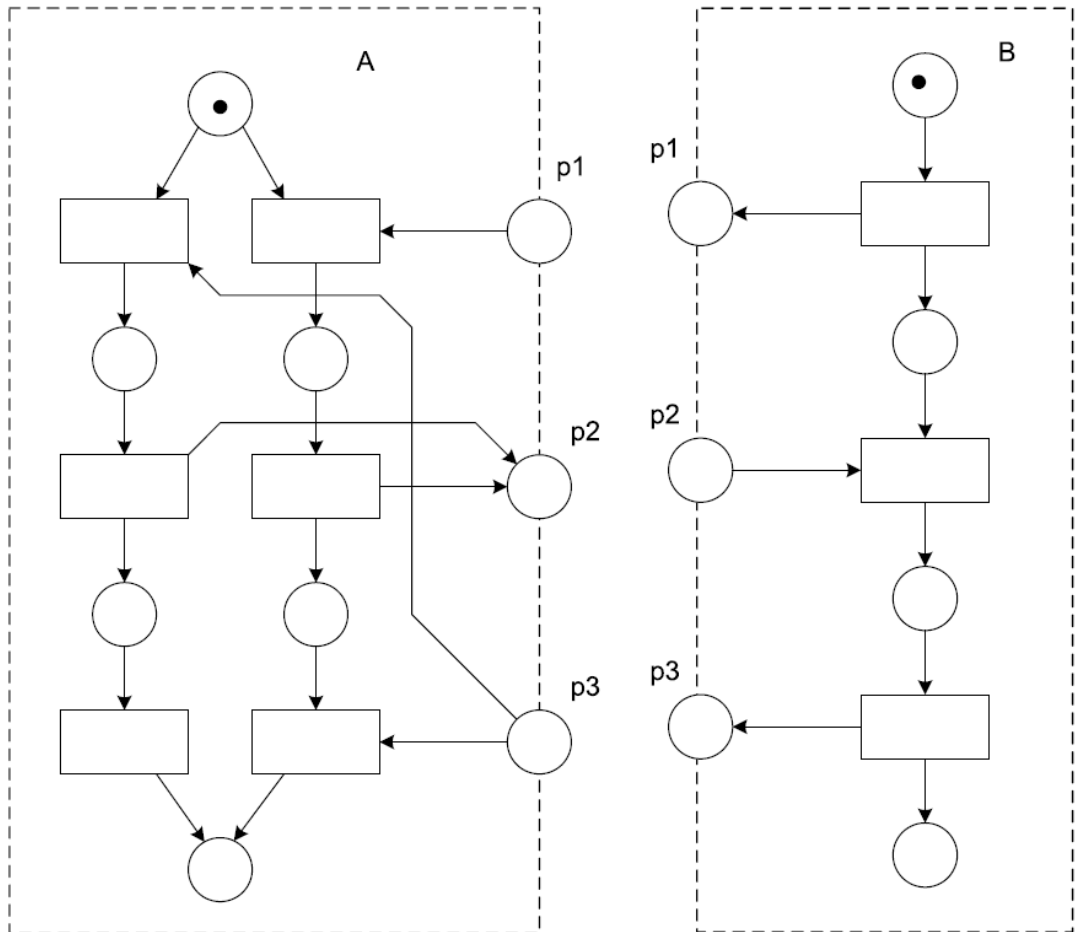


Fig. 6.23. Example of workflow modules with merged communication places

Primer:

Iako mreža nastala kombinovanjem zadovoljava slabi kriterijum korektnosti, nije i korektna, jer zbog načina kompozicije određene grane su potpuno nemoguće

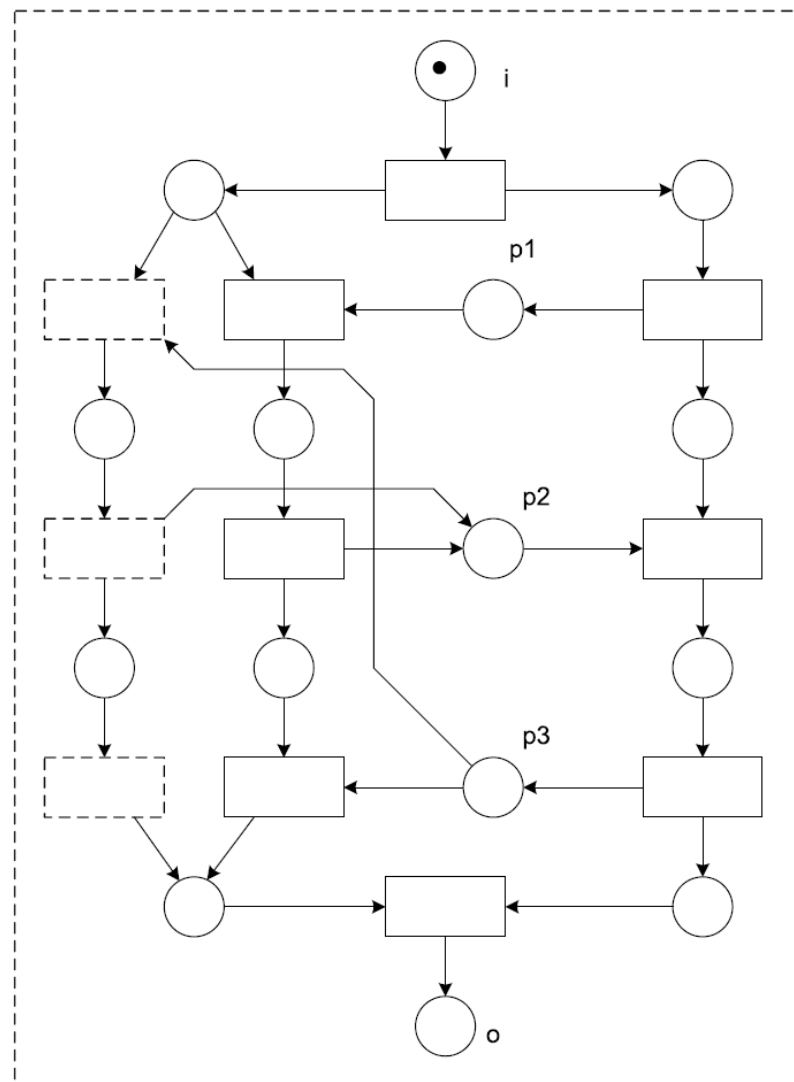


Fig. 6.24. Example of weak sound workflow net