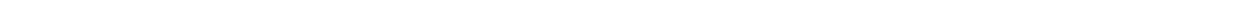


ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА
ДИПЛОМСКЕ АКАДЕМСКЕ СТУДИЈЕ ОИРС

АНАЛИЗА ПРОЦЕСА И ПЕТРИЈЕВЕ МРЕЖЕ

СКРИПТА

Аутор: Драгана Макајић-Николић



САДРЖАЈ

1. АНАЛИЗА ПРОЦЕСА.....	1
1.1. Пословни процеси	1
1.1.1. Дефиниције и карактеристике пословних процеса.....	1
1.1.2. Приручник о организационим процесима МПТ.....	3
1.1.2.1. Управљање зависностима.....	3
1.1.2.2. Декомпозиција и специјализација процеса.....	4
1.1.2.3. Складиште процеса електронског пословања	6
2. ПЕТРИЈЕВЕ МРЕЖЕ (PN)	13
2.1. Дефиниција и динамика Петријевих мрежа	13
2.2. Врсте и развој PN	14
2.2.1. Бинарне PN	15
2.2.2. Обичне (Ordinary) PN	17
2.2.3. Петријеве мреже вишег нивоа (High-level PN)	21
2.3. Обојене Петријеве мреже (CPN).....	23
2.4. Проширења PN	28
2.4.1. Хијерархијске PN	29
2.4.2. Временске PN	29
2.4.3. Стохастичке PN	30
2.5. Анализа Петријевих мрежа	32
2.5.1. Својства Петријевих мрежа	33
2.5.2. Стабло досежљивости	36
2.5.3. Инваријанта	37
2.5.4. Симулација	39
ЛИТЕРАТУРА.....	40

1. АНАЛИЗА ПРОЦЕСА

1.1. Пословни процеси

Како су основне одлике пословања у другој половини XX века убрзане промене технологије, тржишних и друштвених услова, основна потреба савремених предузећа је могућност брзе и лаке прилагодљивости на те промене. То је, последњих година, довело до развоја различитих концепата у пословању: *Just-in-time* (JIT), *Total Quality Management* (TQM), *Flexible Manufacturing Systems* (FMS), *Computer Integrated Manufacturing* (CIM), *Computer Integrated Logistic* (CIL), Реинжењеринг пословних процеса (*Business Process Reengineering* - BPR) итд. Заједничко за све ове концепте је да су, уместо на организациону структуру, оријентисани на организационе (пословне) процесе.

1.1.1. Дефиниције и карактеристике пословних процеса

Неке од најчешће цитираних дефиниција пословних процеса су:

Davenport (1990): “Пословни процес је специфичан низ пословних активности у времену и простору, који има свој почетак и крај и прецизно дефинисане улазе и излазе.”

Harrington (1991): “Пословни процес је свака активност или група активности које узимају улаз, повећавају му вредност и тиме производе излаз за унутрашњег или спољашњег корисника.”

Hammer и *Champy* (1993): “Пословни процес је колекција активности која има више улаза и ствара више излаза који имају неку вредност за корисника.”

Crowston (1998): “Пословни процес је низ активности које изводи организациони актер да би произвео и утрошио ресурсе.”

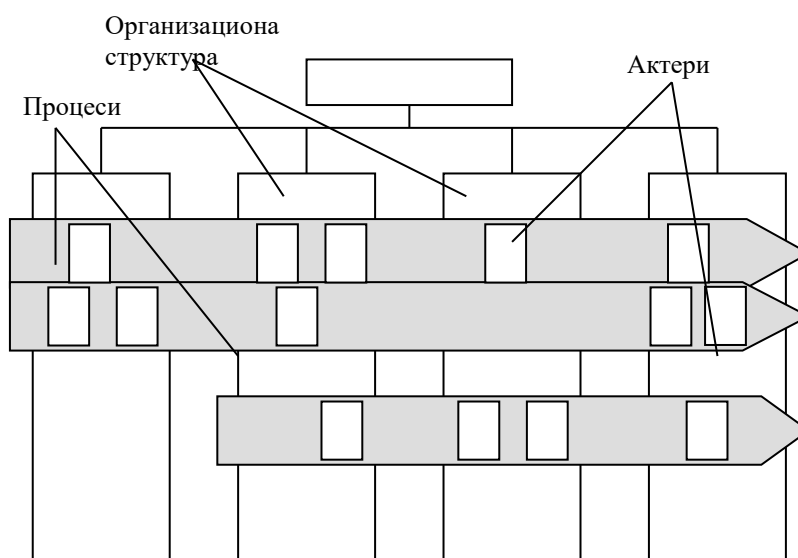
Szibrik, *Ramos* (2001): “Пословни процес је процес друштвене интеракције између агената/актера, односно привремено уређен скуп догађаја које одређују и изводе агенти.”

Из дефиниција се може приметити да се временом пословни процеси све више посматрају са аспекта учесника у њима. Ови учесници се често називају агенти или

актери, који су активни учесници процеса [8], насупротив пасивним објектима процеса. Учесници у процесу могу имати неку од следећих улога [51]:

- Власник процеса – особа која је одговорна за излазе, тј. резултате процеса (најчешће власник предузећа или вођа тима),
- Учесник процеса – особе које извршавају активности процеса и преводе улазе у излазе (најчешће чланови тима),
- Интересне групе или појединци (*stakeholders*) – групе или особе који могу утицати или су под утицајем предузећа (конкуренти, добављачи, финансијске и политичке институције итд.) [36] и
- Корисници пројекта – купци производа или корисници услуге која је резултат (излаз) процеса.

Важна карактеристика пословних процеса, која није поменута у набројаним дефиницијама, је да они пролазе кроз организационе и функционалне границе предузећа. Ако се посматра нека од традиционалних организационих структура, са функционално груписаним задацима и запосленима, пословни процеси се могу приказати као низови активности које пролазе кроз те функције:



Слика 1. Пословни процеси [50]

Пословни процес се може посматрати и као низ активности које повећавају вредност резултата процеса. Овај низ се назива ланац додатне вредности и представља језгро пословног процеса. Поред ових активности, процес чине и активности управљања, односно активности које повезују и координирају активностима језгра процеса.

У наставку ће бити приказан приступ представљању пословних процеса развијен на МИТ (*Massachusetts Institute of Technology, Cambridge, Massachusetts*).

1.1.2. Приручник о организационим процесима МИТ

Крајем 1990-тих година је представљен нови пројекат у МИТ *Center for Coordination Science* назван *Process Handbook Project* (PH) који се састоји у развијању методологије и софтверског алата за представљање организационих процеса на различитим нивоима апстракције и сакупљању, организовању и анализирању примера о томе како различите организације изводе сличне пословне процесе [35]. Сврха PH је да помогне у редизајну постојећих пословних процеса, осмишљавању нових процеса (нарочито оних који користе предности ИТ) и ширењу идеја о организационој пракси.

МИТ методологија представљања пословних процеса се заснива на два концепта: управљање зависностима и декомпозиција-специјализација процеса [35,41,42]. Ови концепти су преузети из рачунарских наука (*Computer science*) и теорије координације и прилагођени су конкретним потребама.

1.1.2.1. Управљање зависностима

МИТ методологија разликује активности производње (језгра процеса) од активности координације. Ова разлика је преузета из теорије координације и карактеристична је за анализу зависности. Основна идеја је да се, кад год постоји зависност између две активности, захтева координација. Активности координације се могу идентификовати на више начина: директним тражењем међу свим активностима, што није увек лако и очигледно, тражењем учесника у процесу који координирају или елиминисањем активности производње из листе свих активности, при чему су активности које остају или активности координације или активности које треба избацити из процеса. Често су активности које користе више информација од осталих (информационо интензивне), управо активности координације.

Постоје различите врсте зависности и различити координациони механизми за управљање њима, при чему се различитим зависностима може управљати на исти начин, као и истом зависношћу на различите начине. Ово даје могућност да се постигну

значајна побољшања у дизајну процеса увођењем бољих координационих механизма или додавањем нових активности координације за управљање зависностима којима се у постојећем процесу не управља. Међутим, зависности између активности могу да представљају и ограничење начинима на које процес може бити редизајниран.

У описаном МПТ приступу су представљене 3 основне врсте зависности, а сматра се да све остале зависности могу да се прикажу као њихова специјализација или комбинација [35,41,42]:

1. Ток (произвођач-корисник) – Једна активност производи излаз који је улаз за другу активност. Код ове зависности треба координирати 3 аспекта:
 - условљеност (потпуна или секвенцијална) – излаз не може да се користи док (потпуно или један његов део) не буде произведен. Механизми координације који се овде могу применити су: *pull* (производња по наруџбини) или *push* (производња за залихе), *Just in Time*, канбан систем, детаљно планирање итд;
 - употребљивост – произведени излаз треба да достигне неки ниво перформанси да би био употребљив за активности које следе. Могући механизми координације су коришћење стандарда или консултовање индивидуалних корисника о нивоу перформанси и
 - доступност – произведени излаз мора да буде физички доступан кориснику. Могуће је транспортовати излаз кориснику или производити на месту употребе.
2. Дељење улаза – Више активности дели исти (ограничени) улаз. Координација овакве зависности се може вршити применом неког од правила *FIFO* (*First In First Out*), *LIFO* (*Last In First Out*), *PRI* (приоритет), *SIRO* (*Stochastic In Random Out*) [62], одлуком менаџера, у односу на захтеве тржишта итд.
3. Слагање, дељење излаза – Више активности доприноси производњи истог излаза. Примери овакве зависности су када тимови раде на различитим деловима производа и морају да усклађују свој рад, или када сировине за производњу неког производа стижу са различитих извора.

1.1.2.2. Декомпозиција и специјализација процеса

У организовању пословних процеса у МПТ приступу се користе две димензије: различити *делови* (декомпозиција) пословних процеса и различити *типови* (специјализација) пословних процеса.

Декомпозиција је најкоришћенији начин за описивање организационих процеса и организационе структуре. Декомпозиција процеса је раздвајање процеса на кораке (активности) који треба да се изврше да би се процес комплетирао, односно остварио његов циљ [41]. Активности могу даље да се декомпонују у подактивности. Резултат овог разлагања је листа активности која треба да садржи: назив активности, носиоца активности (једна особа, одсек, група или информациони систем), циљ активности, улазе, излазе и алате које активност користи и контекст активности (фактори који утичу на успех активности, или су разлог њеног обављања или је подстичу). Сваки процес је истовремено и део (потпроцес) неког већег процеса.

Специјализација је нова димензија у опису процеса која анализира процесе по типу и даје одговоре на питања: како, шта, коме итд. Са аспекта специјализације, процеси се могу организовати у класе процеса које се односе на исти скуп понашања неког система. Класе процеса се могу дефинисати помоћу [67]:

- Минималног скупа понашања система који обухвата сва понашања (активности) које сваки процес из посматране класе мора да садржи. Специјализација процеса се врши додавањем активности које посматрани процес разликује од осталих процеса у класи.
- Максималног скупа понашања система који обухвата сва могућа понашања (активности) које процеси из посматране класе могу да садрже. Специјализација процеса се врши уклањањем активности, односно рестрикцијом овог скупа.

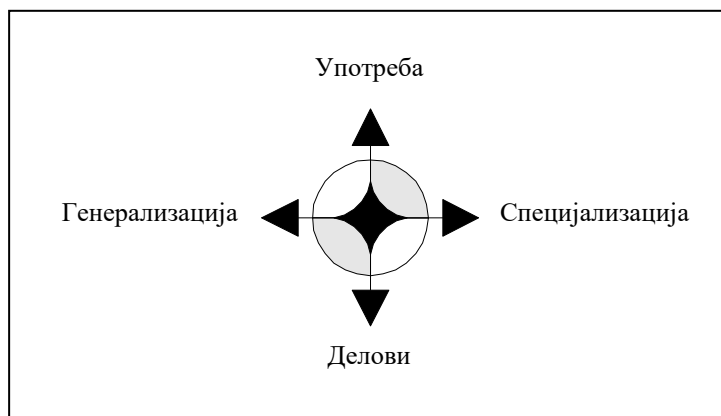
Први начин дозвољава креативност у формирању нових пословних процеса јер даје само њихов костур, док је други начин погодан када је важно спречити нежељена понашања система јер не дозвољава ни једно понашање ван дефинисаних.

Сваки процес се, такође, може посматрати и као специјализација неког општег, генералног процеса, тако да је могуће формирати хијерархију специјализација. Важно својство оваквог представљања процеса је да специјализације процеса наслеђују особине и декомпозицију својих генерализација. Зависности између активности се такође наслеђују кроз хијерархију специјализације.

Описани приступ декомпозиције и специјализације може се графички приказати компасом процеса на слици 2, на коме се вертикална компонента односи на делове, а хоризонтална на типове пословних процеса. Сваки пословни процес се може поставити у

средиште компаса, а избором неке од 4 страна се добијају пословни процеси који дају одговор на следећа питања:

- Који су различити делови овог процеса? - *Parts* (у даљем тексту делови)
- У ком већем процесу се користи овај процес? - *Uses* (употреба)
- Који су други процеси попут овог? - *Generalization* (генерализација)
- На које друге начине може да се изврши овај процес? - *Specialization* (специјализација)



Слика 2. Компас процеса [35]

Описани приступ, са концептима управљања зависностима и декомпозиције-специјализације, омогућава формирање неке врсте граматике за генерисање алтернативних процеса. Граматика се састоји из лексикона основних елемената (пословни процеси описани у бази - *MIT e-Business Process Repository*) и правила или ограничења за комбиновање елемената (механизми координације). Граматички модели могу да дају велики број варијанти за редизајнирање посматраног процеса, али треба спречити да такво формално представљање занемари неке битне особености постојећег процеса.

1.1.2.3. Складиште процеса електронског пословања

У сарадњи са *Phios* корпорацијом је развијена и база (складиште процеса електронског пословања - *MIT e-Business Process Repository*) која тренутно садржи преко 5000 процеса. База садржи и изабране примере и моделе из различитих области пословања, међу којима су:

- Процес контроле менаџмента на примеру *IBM*, којим је обухваћена контрола 13 пословних јединица компаније *IBM* у испуњењу нивоа 7 уговором утврђених

-
- перформанси: раст, профит, повраћај средстава, проток новца, задовољење купаца, квалитет и морал запослених;
- Процес продаје књига преко електронске радње (*Amazon.com*), који се састоји од: куповине књига за залихе и по наруџбини, продаје књига преко електронске радње и управљање појединачном интернет дистрибуцијом;
 - Процес централизованог праћења перформанси добављача (Моторола), који омогућава свим менаџерима набавке, без обзира у којој су филијали Мотороле запослени, да виде перформансе (квалитет, стабилност цена, време испоруке итд.) свих компанијиних добављача из области која их интересује;
 - Процес распоређивања ресурса на задатке, на примеру *General Electrics*. Фабрика *GE, Auburn*, се бави производњом по поруџбини и нуди производе у различитим облицима. У *GE* је развијен систем *ON-TRAC* који запосленима даје податке о приоритету, расположивости материјала, захтевима сваког пројекта, на основу чега запослени без ичијег надзора одлучују на ком ће пројекту и шта да раде;
 - Процес вођења интервјуа са купцима уз показивање прототипа (*AT&T*), где се потенцијални купци изјашњавају о томе шта би на производу променили и колико би га платили;
 - Процес одређивања продајне цене после коришћења производа, на примеру Бостонског филхармонијског оркестра, чији је диригент *B. Zander* одлучио да публика, после извођења концерта, одлучи да ли ће платити;
 - Процес прављења лутки по наруџбини (Барби програм *Mattel* корпорације);
 - Процес прављења рачунара за залихе (*Compac*);
 - Процес дистрибуције снимљене музике преко електронске радње (*MP3.com*) итд.

У представљању процеса у бази је у потпуности подржан концепт декомпозиције и специјализације, и сви процеси су међусобно повезани помоћу компаса процеса.

На највишем нивоу хијерархије (декомпозиције) процеса у МИТ бази је процес који би се могао назвати Пословање (*Produce as a business*). На следећем нивоу хијерархије се налазе процеси:

- Куповање (*Buy*), са подактивностима: Идентификовање сопствених потреба, Идентификовање потенцијалних извора, Избор добављача, Наручивање, Пријем, Плаћање и Управљање односима са добављачима.
- Дизајн (*Design*), са подактивностима: Идентификовање потреба или захтева, Идентификовање могућности производње, Развијање потреба и процес дизајнирања,

-
- Прављење (*Make*),
 - Продавање (*Sell*), са подактивностима: Идентификовање потенцијалних купаца, Идентификовање потреба потенцијалних купаца, Информисање потенцијалних купаца, Добијање поруџбина, Испорука производа или услуга, Примање уплате, Управљање односима са купцима, и
 - Управљање пословањем (*Manage a business*), са подактивностима: Развијање стратегије, Управљање ресурсима, Управљање учењем и променама, Управљање осталим везама са окружењем.

Приказане подактивности се могу даље декомпоновати.

Најопштији процес у бази је процес Деловање (*Act*). У наставку су дати процеси на следећем нивоу генерализације (специјализације) и њихове специјализације:

- Мењање (*Modify*)
 - Шта: Мењање информација, Мењање физичких добара,
 - Како: Поправљање, Припремање, Производња, Померање, Развијање, Редуковање,
- Чување (*Preserve*)
 - Како: Чекање, Настављање, Задржавање, Складиштење,
- Уништавање (*Destroy*)
 - Шта: Избацивање нежељених корака процеса, Елиминисање трошкова/времена у жељеним корацима процеса, Елиминисање нежељених производа, Избацивање транспорта,
 - Како: Повлачење, Елиминисање,
- Управљање (*Manage*),
 - Шта: Управљање зависностима, Управљање односима, Управљање ланцем снабдевања, Управљање ризиком, Управљање животним циклусом, Управљање процесом експлоатације,
 - Како: Ограничавање, Проверавање/Надгледање, Додељивање, Алоцирање, Организовање,
- Комбиновање (*Combine*),
 - Како: Груписање, Повезивање,
- Одвајање (*Separate*),
 - Шта: Раздвајање улаза, Издвајање из скупа, Издвајање из интегрисане целине,
 - Како: Сегментирање, Раздвајање, Извлачење,

- Одлучивање (Decide),
 - Како: Класификовање, Селектовање, Тестирање, Алоцирање,
- Креирање (Create)
 - Шта: Прављење, Дизајнирање
 - Како: Грађење, Развијање, Рачунање, Дуплирање, Представљање, Предвиђање,
- група неклассификованих (Unclassified) процеса: Наслеђивање, Обнављање итд.

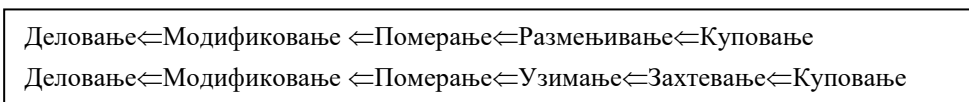
Ови процеси су специјализовани давањем одговора на питања: како, шта, коме итд., и на исти начин се могу даље специјализовати.

Пример 1: У МПТ бази је процес Куповање (*Buy*) описан као процес у коме се врши размена у којој корисник (купац) даје новац произвођачу у замену за нешто што за њега има вредност: производ, информацију или услугу.

Употреба: процес Куповање је део (потпроцес) многих процеса описаних у бази МПТ. Неки од тих процеса су: Пословање великог произвођача, Пословање брокера, Пословање дистрибутера, Дистрибуција информација, Дистрибуција инвестиционог капитала, Продаја преко аукције, Продаја преко електронске радње, Продаја туристичких услуга, Производња енергије, Производња по наруџбини, Производња за залихе, Масовна производња, Производња брзе хране, Обезбеђивање телефонских услуга, Обезбеђивање софтверских услуга, Обезбеђивање интернет услуга, Обезбеђивање транспортних услуга, Тржишна размена итд.

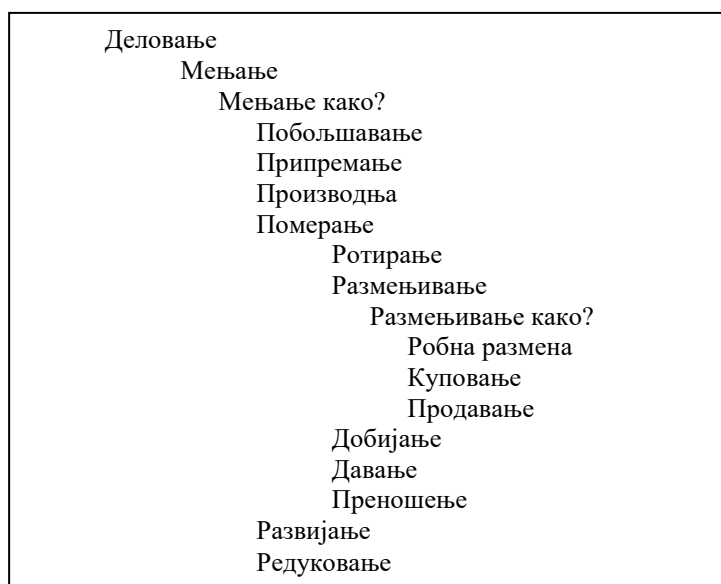
Делови, односно потпроцеси процеса Куповање су: Идентификовање сопствених потреба, Идентификовање потенцијалних извора, Избор добављача, Наручивање, Пријем, Плаћање и Управљање односима са добављачима.

Генерализација процеса Куповање се може извршити на један од два начина са слике 3:



Слика 3. Генерализације процеса Куповање

Детаљнији приказ прве генерализације је дат на слици 4:



Слика 4. Детаљна генерализација процеса Куповање

Специјализација: неке од понуђених специјализација процеса Куповање су:

- За коју сврху: Куповина стандардних производа за залихе, Куповина стандардних производа по наруџбини, Куповина посебних производа по наруџбини;
- Шта: Куповина сировина, Куповина физичких ресурса, Куповина информационих ресурса, Ангажовање људских ресурса;
- На који начин: Куповина директно на тржишту, Куповина лицем у лице, Куповина у радњи, Куповина преко интернета итд.

Као што је већ наглашено специјализације процеса наслеђују својства својих генерализација. Тај принцип је примењен и у бази МИТ, па се сваки од процеса међу набројаним специјализацијама састоји из активности: Идентификовање сопствених потреба, Идентификовање потенцијалних извора, Избор добављача, Наручивање, Пријем, Плаћање и Контакт са добављачима. За конкретни специјализовани процес, ове активности су идентичне или конкретизоване, а могу бити и подељене у подактивности. Тако је активност Контакт са добављачима у процесу Куповина сировина подељена у подактивности: Процењивање добављача, Управљање политиком према добављачима и Управљање односима са добављачима.

Неке од техника за моделирање и анализу пословних процеса су:

- Дијаграм активности. У [25,24] је приказан Дијаграм активности као све чешће коришћена, кориснику блиска верзија Дијаграма промене стања. Развијена је и методологија моделирања пословних процеса која обједињује Структурну систем анализу (ССА) и Дијаграм активности (ДА). ССА се користи као метода за декомпоновање целокупног система у хијерархију пословних функција и њихов детаљан опис. Тиме се долази до логичког модела, односно скупа општих “атомских” активности независно од конкретног посла у коме ће се обављати. Затим се, помоћу ДА, описује динамика послова, односно описује се начин повезивања “атомских” активности у специфичан појединачни посао. Тиме се добија физички модел посматраног пословног процеса;
- *IDEF (Integrated Definition for Function Modeling)* [63] је техника за моделирање тока активности и ентитета у процесу. Настала је 1981. године као део пројекта *Integrated Computer-Aided Manufacturing (ICAM)*. Постоје бројне верзије технике, али основне су *IDEF0* и *IDEF3*, а у литератури се може наћи и са ознаком *SADT*. *IDEF* дијаграми су оријентисани на активности и њихову повезаност ентитетима (најчешће подацима), који су представљени улазима и излазима активности (дијаграм има и дуалну варијанту у којој су ентитети повезани активностима). Улаз у активност може бити излаз из неке друге или контрола посматране активности. Активности се може придружити и *mechanism*, односно особа или систем који је извршавају. *IDEF* техника омогућава и хијерархијски приказ, али само тако што се свака активност може, у посебном дијаграму, описати помоћу подактивности. Главна снага технике је у

-
- једноставности приказа и у томе што постоји велики број алата који је подржавају. Недостатци *IDEF* технике су: не садржи апарат за анализу моделираног процеса, није примењива за велике моделе, не може да моделира кашњење и чекање, као ни праву динамику процеса (због чега се сматра псеудо-динамичком методом [60]) итд;
- *PAG (Process Activity Graph)* [8] је двојка (S,A) , где је S скуп чворова, а A скуп грана. Чворовима S су приказана стања у којима организација може да се нађе, а гранама A активности којима организација из једног стања прелази у неко друго. На овај начин *PAG* моделира кретање организације, преко низа активности, из почетног (улазног) у једно или више циљних (излазних) стања. Сваки низ активности који повезује почетно са неким од завршних стања се назива пут процеса. Сваки од путева може моделирати један пословни процес у организацији, али овако дефинисан граф се може користити и за моделирање различитих варијанти (специјализација) истог процеса. *PAG* се може приказати и матрично, а техника се може проширити и увођењем стохастике у модел. Тиме се добија добар алат за анализу пословних процеса. Недостатак технике је у томе што не обухвата ресурсе (актере, капацитете, податке) који су веома важни јер се у пословним процесима често јавља заузетост актера, дељење капацитета итд. Међутим, главни и веома изражени проблем *PAG*, као и сваке технике оријентисане на стања система, је што не може да се примени на велике моделе, као ни на мале моделе (процесе) са великим бројем могућих стања;
 - *AW (Action-Workflow)* [68,69]. *AW* техника се сматра техником за моделирање комуникационе структуре у пословним процесима. Основна јединица *AW* модела је *workflow* између извођача активности процеса и корисника резултата. Сваки *workflow* се састоји из четворофазне петље која моделира следећа 4 стања: предлагање, пристајање, извођење и прихватање (примање), која подразумевају различите типове комуникације између учесника. У свакој *workflow* петљи се ради о обавезивању извођача да испуни неки захтев корисника. Петље су међусобно повезане луковима, којима се моделира прелазак из једног у следећи комуникациони однос. Главни недостатак ове технике је што је оријентисана на комуникацију унутар процеса а не на то како се сам процес одвија. *AW* модел се, при томе, ограничава на комуникацију између два учесника, а у већини пословних процеса је број заинтересованих страна углавном већи од два;
 - *SD (Strategic Dependency, Actor Dependency)*. *SD* је техника моделирања оријентисана на односе између учесника пословних процеса. Претпоставка технике је да између учесника постоји однос зависности у коме један учесник (*dependor*) зависи од другог
-

учесника (*dependee*). Ова зависност је увек у вези са неким објектом зависности (*dependum*), који може бити активност, ентитет или услов. *SD* модел се састоји из чворова и више типова лукова. Чворовима се моделирају учесници пословних процеса, а гранама типови зависности који указују на различите степене слободе који могу постојати у односима зависности. Зависности, које се графички приказују различитим гранама, могу бити:

- зависност циља, један учесник процеса, у остварењу свог циља, зависи од услова које треба да му обезбеди други учесник;
- зависност задатка, један учесник зависи од активности другог учесника;
- зависност ресурса, један учесник зависи од ресурса које му обезбеђује други учесник; и
- мека зависност циља, варијанта прве зависности у којој је зависном учеснику остављена могућност да одлучи да ли ће чекати да му други учесник обезбеди потребне услове или ће деловати без њих (када је то могуће).

SD модели се могу даље проширивати увођењем различитих типова чворова или увођењем хијерархије. Тиме се повећава снага моделирања, али се не превазилазе главни недостаци технике: немогућност моделирања динамике пословних процеса и непостојање апарата за анализу модела;

- Ланци процеса вођени догађајима (*EPC - Event Driven Process Chains*) [60] је техника која је развијена, почетком 90-тих година у Немачкој, као база за алате за моделирање пословних процеса (*ARIS, Nautilus, SAP R/3*). *EPC* је граф који се састоји из 3 врсте чворова: активних, пасивних и веза. Активни чворови представљају активности, а пасивним се моделирају различита “стања” система (нпр. у алату *ARIS* постоји 5 типова пасивних чворова за моделирање различитих врста ресурса). Чворови веза моделирају логичке везе између активности: и, или ексклузивно или. *EPC* техника омогућава и хијерархијско моделирање декомпоновањем активности на подактивности, односно разлагањем активних чворова на поддијаграме. Од механизма координације који се односе на коришћење ресурса, *EPC* може да моделира само додељивање ресурса. Основни недостатак технике је немогућност моделирања динамике процеса;
- Петријеве мреже, чија примена ће бити детаљније описана у наставку.

2. ПЕТРИЈЕВЕ МРЕЖЕ (PN)

2.1. Дефиниција и динамика Петријевих мрежа

Структура Петријеве мреже (*Petri net* – PN) је оријентисан, тежински, бипартитни граф који се састоји из две групе чворова:

- Прелаз (*transitions* – t) – који представљају догађаје који треба да се одиграју или операције које треба да се изврше. Графички се приказују правоугаонцима или линијама, и
- Места (*places* – p) – која представљају узрок и/или последицу догађаја или бафер у коме су смештене информације или други ресурси потребни за извршење операције. У литератури се може наћи и ознака s (од немачког *stellen* [15]). Места се графички приказују круговима или елипсама.

Оријентисане гране у PN повезују места са прелазима и прелазе са местима, никад два прелазе или два места. Свакој грани је придружена тежина као позитиван цео број. Уколико недостаје, подразумева се да је тежина 1.

Формално се структура PN описује на следећи начин:

Дефиниција 1: Структура Петријеве мреже је четворка $N=(P, T, A, W)$, где је [45]:

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_n\} \text{ коначан скуп места,} \\ T &= \{t_1, t_2, \dots, t_q\} \text{ коначан скуп прелазе,} \\ A &\subseteq (P \times T) \cup (T \times P) \text{ коначан скуп грана,} \\ W &: A \rightarrow \{1, 2, \dots\} \text{ функција тежина придружена гранама,} \end{aligned} \tag{1}$$

и где је $P \cap T = \emptyset$ и $P \cup T \neq \emptyset$.

Нека је [45]:

$$\begin{aligned} {}^{\circ}t &- \text{ скуп свих улазних места прелазе } t, \text{ тј. } {}^{\circ}t = \{p \mid p \in P, (p, t) \in A\}, \\ t^{\circ} &- \text{ скуп свих излазних места прелазе } t, \text{ тј. } t^{\circ} = \{p \mid p \in P, (t, p) \in A\}, \end{aligned}$$

${}^{\circ}p$ – скуп свих улазних прелаза места p , тј. ${}^{\circ}p = \{t \mid t \in T, (t, p) \in A\}$,

p° – скуп свих излазних прелаза места p , тј. $p^{\circ} = \{t \mid t \in T, (p, t) \in A\}$.

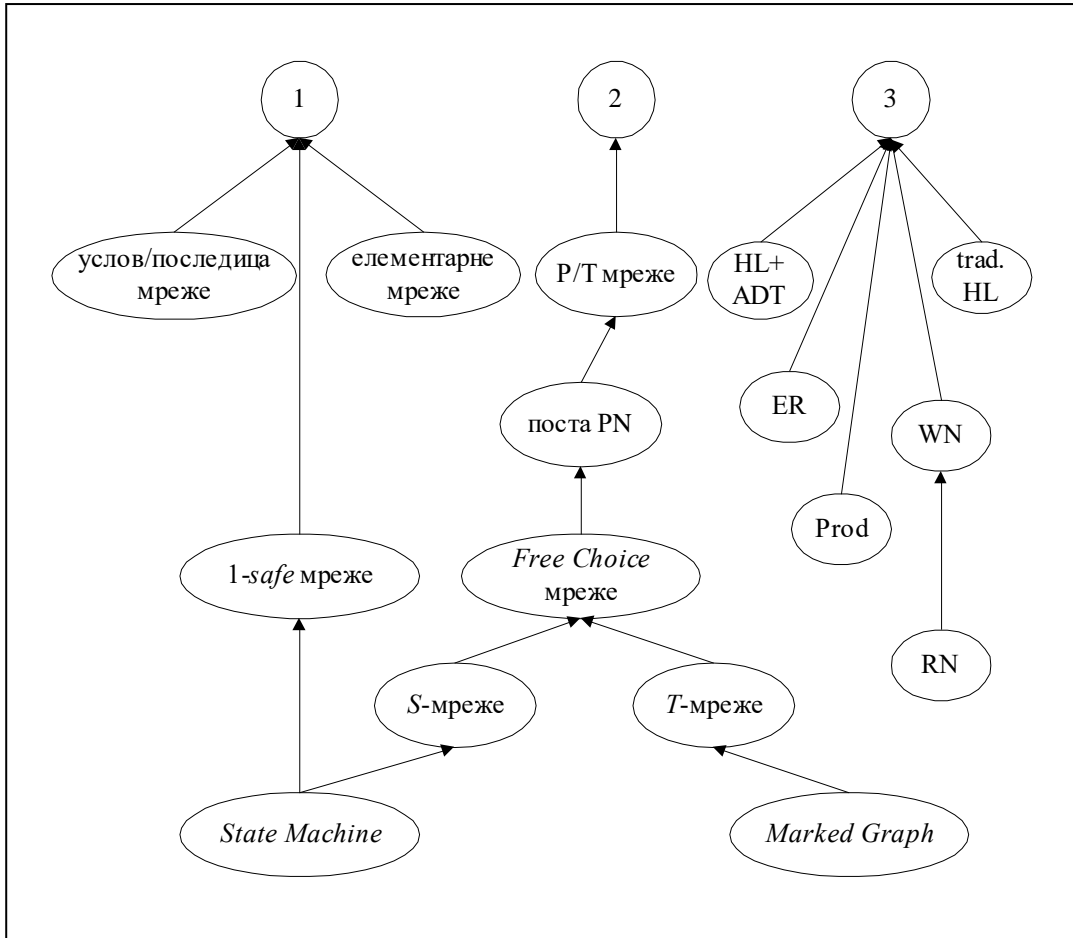
Свако место у PN може да садржи један или више жетона (токена, знакова - *token*), који омогућавају моделирање динамике система. Број жетона у месту p се назива маркирање (*marking*) и означава са $M(p)$. Расподела жетона у местима Петријеве мреже се назива маркирање PN и означава са M . Једно маркирање PN представља једно стање моделираног система. Почетно маркирање на мрежи означава се са M_0 , тако да се Петријева мрежа најопштије може приказати као $PN=(N, M_0)$.

Променом маркирања PN се моделира промена стања система. Маркирање се мења када се неки од прелаза запали (*firing*), односно догоди (*occur*). Паљењем прелаза t се из свих места $p \in {}^{\circ}t$ ослобађају жетони, а у свим местима $p \in t^{\circ}$ се појављују жетони. Тиме се добија ново маркирање, односно ново стање моделираног система. Колико ће се жетона ослободити, односно појавити зависи од врсте PN. Скуп свих маркирања која се могу достићи низом трансформација из неког маркирања M паљењем низа прелаза означава се са $R(M)$.

Да би се одређени прелаз запалио, односно догодио потребно је да буду испуњени одређени услови, чиме је тај прелаз омогућен (*enabled*). Услови за паљење прелаза, такође, зависе од врсте Петријеве мреже.

2.2. Врсте и развој PN

Петријеве мреже се разликују у томе колико жетона могу да садрже њихова места и да ли су жетони неструктурирани или имају структуру која одражава објекат који моделирају. На слици 10 је приказана класификација PN у 3 врсте [12]:



Слика 5. Класификација Петријевих мрежа

Прву врсту PN са слике 10 чине бинарне PN, другу врсту чине обичне (*Ordinary*) PN а трећу мреже вишег нивоа (*High-level PN*).

2.2.1. Бинарне PN

Уколико место у PN представља услов, онда присуство, односно одсуство (једног) жетона указује да ли је услов испуњен (*true*) или не (*false*). Ови жетони се још називају буловски жетони зато што њихов број у месту може бити 0 или 1. Буловски жетони се графички представљају (црном) тачком у месту. Мрежа која садржи само места са буловским жетонима се назива PN нивоа 1 или бинарна PN. Најпознатији тип овакве мреже је **Услов/Последица мрежа** (*Condition/Event, C/E*). То је мрежа коју је С.А. Petri представио у својој докторској тези, односно то је прва Петријева мрежа.

Структура C/E мреже је чиста, односно без петљи (*self-loop*). Петље су парови (p, t) такви да $p \in {}^o t$ и $p \in t^o$, тј. да је p улазно и излазно место прелаза t . Формално се C/E мрежа

може представити на следећи начин [12]:

Дефиниција 2: C/E мрежа је четворка (P, T, A, C) , где је:

$$\begin{aligned}
 P &= \{p_1, p_2, \dots, p_n\} \text{ коначан скуп места,} \\
 T &= \{t_1, t_2, \dots, t_q\} \text{ коначан скуп прелаза,} \\
 A &\subseteq (P \times T) \cup (T \times P) \text{ коначан скуп грана,} \\
 C &= \{M_1, M_2, \dots\} \text{ скуп свих маркирања која нису } \textit{deadlock}^*.
 \end{aligned}
 \tag{2}$$

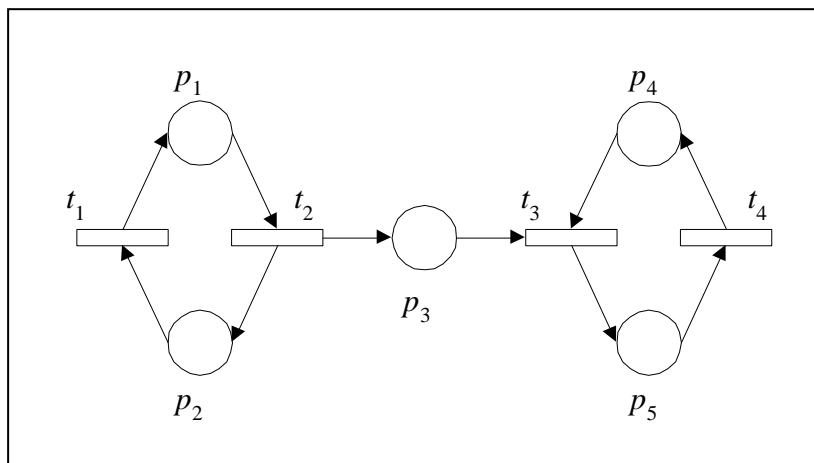
Правило паљења прелаза се у C/E мрежама може описати на следећи начин: сваки прелаз t C/E мреже може да се упали у неком маркирању M ако:

$$\forall t \in T \exists M \in C (M(o_t) = 1 \wedge M(t^o) = 0)
 \tag{3}$$

односно, ако у свим улазним местима прелаза t постоји по један жетон (тј. услов је *true*) и ако су сва излазна места прелаза t празна.

Када се прелаз t запали, из свих његових улазних места се ослобађа по један жетон, а у свим његовим излазним местима се производи по један жетон. Важно својство C/E мрежа је да свако маркирање може да се достигне из било ког другог маркирања у коначном броју корака унапред или уназад.

Пример 2: На слици 11 је приказана структура PN која представља део процеса набавке. Задужени из функције производње (у даљем тексту производња) формира налоге за набавку одређене сировине и шаље их задуженом из функције набавке (у даљем тексту набавка). Када прими захтев, набавка извршава набавку тражене сировине. Производња може у једном тренутку да формира само један захтев за набавку, а набавка да истовремено извршава само један захтев. Производња и набавка су повезане једним бафером у који стаје највише један захтев за набавку. То значи да производња може да пошаље нови захтев за набавку тек када је претходни преузет.



Слика 6. C/E мрежа процеса набавке

* *Deadlock* маркирање је маркирање из кога се не може прећи ни у једно маркирање.

Прелаз t_1 са слике 11 представља формирање, а t_2 слање налога за набавку. Местом p_1 је моделиран услов да налог буде послат: када се формира налог за набавку (упали прелаз t_1) у месту p_1 се појави један жетон, што омогућава паљење прелаза t_2 , тј. слање налога у бафер. Када се прелаз t_2 запали, ослобађа се жетон из места p_1 , а појављује се у месту p_2 . Жетон у месту p_2 моделира стање у коме је прослеђен претходни налог за набавку, што представља услов за поновно паљење прелаза t_1 , односно формирање новог налога за набавку. Паљењем прелаза t_2 истовремено се појављује жетон у месту p_3 , које моделира бафер у који се смештају налози за набавку. Жетон у овом месту омогућава паљење прелаза t_3 , односно, преузимање налога из бафера. Паљење овог прелаза је условљено и жетоном у месту p_4 који указује да је претходни налог извршен и да се може преузети нови.

Паљењем прелаза t_3 , тј. преузимањем налога за набавку из бафера, појављује се жетон у месту p_5 које моделира преузете налоге. Истовремено се ослобађа жетон из места p_4 . Одсуство жетона у месту p_4 значи да се може кренути са извршавањем тренутног налога. То је услов за паљење прелаза t_4 , који моделира извршење тренутног налога, односно набављање сировине. Када је сировина набављена, тј. када се прелаз t_4 запали, ослобађа се жетон из места p_5 , а појављује у месту p_4 , чиме је задовољен услов за поновно паљење прелаза t_3 , односно преузимање новог налога за набавку, ако га има у баферу (тј. ако је жетон у месту p_3).

C/E мрежа (P, T, A, C) са слике 11 се може формално описати на следећи начин:

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

$$A = \{(t_1, p_1), (p_1, t_2), (t_2, p_2), (p_2, t_1), (t_2, p_3), (p_3, t_3), (t_3, p_5), (p_5, t_4), (t_4, p_4), (p_4, t_3)\}$$

$$C = \{[1,0,0,1,0], [0,1,0,1,0], [0,1,1,1,0], [1,0,1,1,0], [0,1,0,0,1], [1,0,0,0,1], [0,1,1,0,1]\}$$

Скуп C је скуп свих могућих маркирања. Тако нпр. прво маркирање $[1,0,0,1,0]$ значи да се у местима p_1 и p_4 налази по један жетон, а да су остала 3 места празна. Ово маркирање моделира стање у коме је формиран нови налог за набавку и извршен претходни налог. Док последње маркирање $[0,1,1,0,1]$ моделира стање у коме је производња спремна за формирање новог налога за набавку, тренутни налог је у баферу а претходни налог је преузела набавка и спремна је за извршење.

У групи бинарних мрежа се налазе и:

- Елементарне мреже (*Elementary Net*, EN). То су C/E мреже са задатим почетним маркирањем;
- 1-Safe мреже;
- *State machine*.

Ове мреже имају велики апарат за анализу моделираних процеса, али је њихова употреба ограничена на моделирање малих и релативно једноставних система.

2.2.2. Обичне (*Ordinary*) PN

Ако је местом Петријеве мреже моделиран бафер, онда број жетона у месту указује на број, тј. количину информација или другог ресурса у баферу. PN чија места могу да садрже више од једног жетона се назива мрежа 2. нивоа, обична (*Ordinary*) или “црно-бела” PN. Друго важно својство ове мреже је да су жетони у њој неструктурирани и да

су сви графички представљени (црним) тачкама. Ове мреже се формално дефинишу на следећи начин:

Дефиниција 3: Обична (“црно-бела”) PN је петорка $PN=(P, T, A, W, M_0)$, где је [45]:

$$\begin{aligned}
 P &= \{p_1, p_2, \dots, p_n\} \text{ коначан скуп места,} \\
 T &= \{t_1, t_2, \dots, t_q\} \text{ коначан скуп прелаза,} \\
 A &\subseteq (P \times T) \cup (T \times P) \text{ коначан скуп грана,} \\
 W &: A \rightarrow \{1, 2, \dots\} \text{ функција тежина придружена гранама,} \\
 M_0 &: P \rightarrow \{0, 1, 2, \dots\} \text{ почетно маркирање,}
 \end{aligned} \tag{4}$$

и где је $P \cap T = \emptyset$ и $P \cup T \neq \emptyset$.

Да би се симулирало динамичко понашање система, маркирање у црно-белој PN се мења по следећим правилима паљења (*firing*) прелаза [39]:

1. Прелаз t је омогућен (*enabled*) ако

$$\forall p \in {}^{\circ}t, M(p) \geq W(p, t) \tag{5}$$

тј. када улазно место садржи најмање онолико жетона колика је тежина гране која води од места p до прелаза t , где је $M(p)$ маркирање, односно број жетона у месту p за маркирање M , а $W(p, t)$ тежина гране од (p, t) .

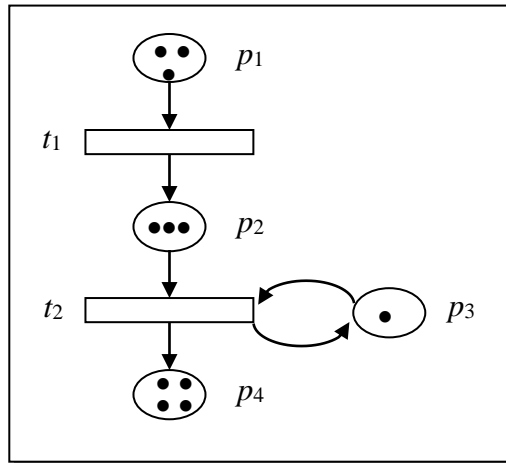
2. Паљење прелаза t уклања $W(p, t)$ жетона из сваког места $p \in {}^{\circ}t$, и додаје $W(t, p)$ жетона у свако место $p \in t^{\circ}$.

Формално, паљење прелаза је трансформација неког маркирања M у маркирање M' , дефинисана са:

$$M'(p) = \begin{cases} M(p) - W(p, t) & \text{ако } p \in {}^{\circ}t \\ M(p) + W(p, t) & \text{ако } p \in t^{\circ} \\ M(p) & \text{иначе} \end{cases} \tag{6}$$

Пример 3: У примеру 2 функције производње и набавке могу истовремено да формирају, односно извршавају само по један захтев за набавку и повезане су бафером у који стаје највише један захтев. У реалним системима то обично није случај. Функција производње може да формира више налога, шаље их чим их формира, а функција набавке их преузима и извршава у складу са својим могућностима: један по један, ако у набавци ради један запослени, или више истовремено, ако ради више запослених.

Процес из примера 2 се помоћу црно-беле PN може приказати на следећи начин:

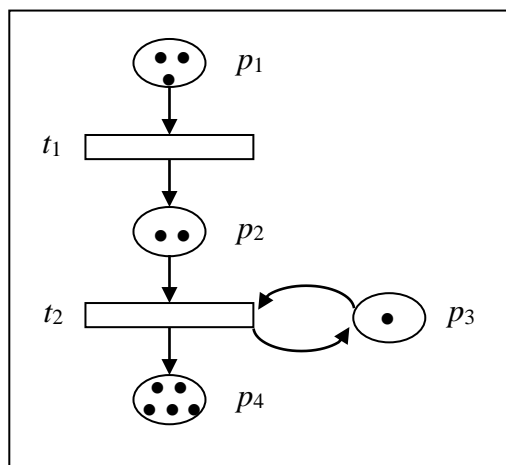


Слика 7. Црно-бела PN процеса набавке

Место p_1 моделира припремљене налоге за набавку, прелаз t_1 њихово слање задуженима у функцији набавке, а место p_2 бафер са прослеђеним налозима. Прелаз t_2 представља извршење налога за набавку, а место p_4 извршене налоге. У почетном маркирању постоје 3 припремљена, 3 прослеђена и 4 извршена налога за набавку. Место p_3 моделира запослене који извршавају налоге за набавку. Како се у месту налази само 1 жетон значи да је само 1 запослени задужен за извршавање ове активности. Овај жетон обезбеђује да прелаз t_2 не може да се запали више од једанпут истовремено, што моделира стање да тај запослени може да извршава само један налог у једном тренутку. Да је број жетона у месту p_3 већи, нпр. 3, то би значило да су 3 запослена ангажована на извршењу налога за набавку. Тада би прелаз t_2 могао да се запали 3 пута истовремено, што значи да у једном тренутку могу да се извршавају 3 налога.

Жетони у овој мрежи су неструктурирани, али могу представљати различите објекте. Који ће објекат бити моделиран зависи од места у коме се жетон налази. Тако жетони у местима p_1 и p_2 представљају необрађене налоге за набавку, а у месту p_4 обрађене налоге за набавку. Жетони у месту p_3 не представљају налоге, већ запослене (у овом случају једног) задужене за обрађивање пристиглих налога за набавку.

У почетном маркирању $[p_1, p_2, p_3, p_4] = [3, 3, 1, 4]$ са слике 3 су омогућена оба прелазна, односно, могуће је проследити неки од 3 формирана или извршити неки од 3 прослеђена налога. Паљењем нпр. прелазна t_2 се уклања 1 жетон из места p_2 и 1 жетон из места p_3 . Истовремено се производи по 1 жетон у местима p_4 и p_3 . Овим се моделира узимање једног прослеђеног налога за набавку и његово извршавање. При том извршавању је заузет запослени који обавља извршење налога, а након извршења се он ослобађа. Као резултат овог паљења добија се маркирање $[3, 2, 1, 5]$ на слици 13.



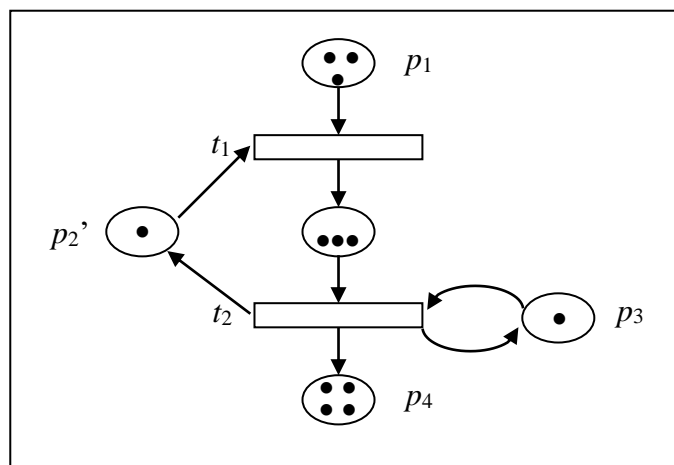
Слика 8. Црно-бела PN процеса набавке после паљења прелазна t_2

Бафери у које се смештају одређени физички ресурси и информације могу бити ограниченог капацитета. Како се ови бафери моделирају местима, онда и места морају бити ограниченог капацитета. Мрежа са таквим местима се назива мрежа са ограниченим капацитетом. Код мрежа са ограниченим капацитетом постоји додатни услов за паљење прелаза t , а то је да број жетона у сваком излазном месту $p \in {}^o t$ не прелази његов капацитет $Q(p)$ после паљења прелаза t . Мрежа са ограниченим капацитетом се може увек трансформисати у мрежу са неограниченим капацитетом, помоћу следећа два корака (претпоставља се да је PN чиста) [39]:

корак 1. сваком месту p са ограниченим капацитетом, додати фиктивно место p' са почетним маркирањем $M'_0(p') = Q(p) - M_0(p)$,

корак 2. између сваког улазног прелаза t места p и фиктивног места p' нацртати нову грану (p', t) такву да је $W(p', t) = W(t, p)$; између сваког излазног прелаза t места p и фиктивног места p' нацртати нову грану (t, p') такву да је $W(t, p') = W(p, t)$. На овај начин збир жетона у месту p и његовом фиктивном месту p' остаје $Q(p)$, пре и после паљења прелаза.

Пример 4: Нека је место p_2 са слике 12 капацитета 4. То значи да у бафер где се стављају прослеђени налози могу да стану највише 4 налога за набавку. Овај ограничени капацитет места p_2 се може индиректно моделирати увођењем новог места p_2' у претходна 2 корака, чиме се добија мрежа на слици 14:



Слика 9. Моделирање ограниченог капацитета места

Са слике се може видети да када је нпр. место p_2 пуно (тј. када су у њему 4 жетона), прелаз t_1 не може да се упали, јер је фиктивно место p_2' празно. Маркирања која се могу достићи у PN на слици 12 и у PN на слици 14, из истог почетног маркирања су иста, односно ове две мреже су еквивалентне [39].

Мрежа дефинисана са (4) се још назива и *Place/Transition (P/T)* мрежа, а осим ње, у групу обичних PN спадају и:

- *Free Choice (FC)* мреже. FC је мрежа у којој је свака грана места p или јединствена излазећа или јединствена улазећа грана неког прелаз t [12,39];
- *S-мреже*. S-мрежа је подкласа FC мреже у којој сваки прелаз t има највише једно улазно и једно излазно место [12];
- *T-мреже*. T-мрежа је подкласа FC мреже у којој место p има највише један улазни и један излазни прелаз [12];
- *Marked graph (MG)*. MG је чиста проста PN тежине свих грана су једнаке 1) таква да свако место p има тачно један улазни и тачно један излазни прелаз [39,45].

Бинарне и црно-беле мреже се још називају и Петријеве мреже ниског нивоа (*low-level PN*). Црно-беле PN имају значајно већу моћ у моделирању од бинарних PN, а при томе задржавају могућност за детаљну анализу моделираног процеса. За ове мреже се може рећи да су, по могућности примене, између бинарних и Петријевих мрежа вишег нивоа (*high-level PN*), које ће бити објашњене у следећем поглављу, а које имају изузетну моћ моделирања али далеко сиромашнији апарат за анализу у односу на *low-level PN*.

2.2.3. Петријеве мреже вишег нивоа (*High-level PN*)

У раду са *low-level* Петријевим мрежама основни проблем су велике димензије модела које се добијају ако се моделира реалан систем. Да би се овај проблем превазишао уведене су *high-level* Петријеве мреже (HLPN) у којима се, за означавање елемената мреже, користе структурирани жетони и алгебарски изрази. За разлику од PN, које су аутори дефинисали на различите начине, у случају HLPN се уводе интернационални стандарди за синтаксу, семантику и графичку презентацију HLPN. Стандардом се бави један од комитета ISO [15,16], чији су чланови аутори из 31 земље. Овај стандард још увек није обухватио или разрешио све елементе, као што су својства понашања и структурна својства мрежа, чак ни ознаке места (p од енглеског *place* и италијанског *posti* или s од немачког *stellen*).

HLPN поред структуре дефинисане са (1) садржи: скуп типова (боја) жетона, функције које се придружују местима и прелазима, описе грана и почетно маркирање [15]. У [12] су HLPN подељене у следеће групе:

-
- HLPN са апстрактним типовима података (HL+ADT). Ове мреже су алгебарске HLPN где су жетони и правила преласка из једног маркирања у друго дефинисани помоћу алгебарских израза;
 - *Environment Relationship* (ER) мреже, у којима жетони представљају функције које пресликавају скуп идентификатора (*ID*) тј. променљивих у скуп вредности (*V*);
 - *Product* (Prod) мреже, које садрже две специјалне врсте грана: инхибитор гране и бришуће (*erase*) гране. Ове мреже се састоје из описа (предговора, *preamble*) са дефиницијама скупова и функција, неозначене мреже и ознака жетона, прелаза и домена за места, односно врсте жетона који се могу наћи у месту;
 - *Regular* мреже (RN) су обојене мреже (CPN, детаљније у наставку текста) у којима се свака вредност жетона појављује само једном, односно места у RN могу представљати само скупове;
 - *Well-Formed (Coloured)* мреже (WN) су проширења RN која имају исту снагу моделовања као CPN, али су описи скупова типова (боја) дати у много експлицитнијем облику;
 - Објектне PN, у којима су и жетони – Петријеве мреже. У оваквим HLPN постоје две врсте мрежа: мреже објекти, тј. жетони и системске мреже, тј. мреже којима се моделира процес [14];
 - Традиционалне HLPN, које чине *Predicate/Transition* мреже (PrTN) и *Coloured* PN (CPN). Ове две врсте мрежа су веома сличне. PrTN је прва HLPN [21] која је прво теоретски уведена, а њена (широка) примена је објављивана у каснијим радовима. CPN мреже имају две различите презентације: описну (*expression*, која је блиска PrTN) и функционалну (која превазилази неке проблеме PrTN у анализи моделираног процеса). CPN, са својим проширењима, су најчешће коришћене Петријеве мреже у моделирању и анализи процеса.

Иако је део HLPN, CPN је у овом раду детаљније објашњена у посебном поглављу, зато што су управо ове мреже коришћене у моделирању и анализи пословних процеса.

2.3. Обојене Петријеве мреже (CPN)

Прву верзију CPN (названу CP⁸¹-мрежа) дефинисао је 1981. године *Kurt Jensen* у раду “*Coloured Petri Nets and the Invariant Method*”, а прва свеобухватна дефиниција CPN је објављена 1986. године у раду “*Coloured Petri Nets*” истог аутора [21]. Основна одлика CPN је да је сваком месту придружен тип податка, који одређује врсту податка који место може да садржи. Тип податка се на мрежи уписује одмах поред места, фонтом *italic* или у боксу поред места. Сваки жетон у месту носи вредност податка, која припада типу тог места. Жетони се на мрежи такође уписују поред места, а не у месту у коме се налазе. Ова вредност податка се назива и боја жетона, а тип податка – скуп боја.

Како место p у CPN може да садржи два или више жетона са истом вредношћу, садржај места се не може представити као скуп, већ као фамилија (*multi-set*) [6] жетона.

Дефиниција 4 [21]: Фамилија m , над непразним скупом S , је функција $m: S \rightarrow N$ која се представља као формална сума

$$\sum_{s \in S} m(s) \cdot s. \quad (7)$$

Са S_{MS} означава се скуп свих фамилија над S , где је MS скраћеница за фамилију (*multi-set*). Коефицијенти фамилије су ненегативне целобројне вредности $\{m(s) | s \in S\}$. $s \in m$ акко $m(s) \neq 0$.

Другим речима, $m(s)$ представља број појављивања елемента $s \in S$ у фамилији m . Ако се у некој фамилији над скупом S сваки елемент из S појављује само по једном ($1 \cdot s, s \in S$), онда се фамилија може представити и самим скупом S . Фамилије поседују велики број алгебарских својстава и над њима се могу извршити бројне стандардне операције (сабирање, одузимање, множење скаларом итд.) [21].

Маркирање у CPN сада може да се представи као функција која сваком месту p придружује фамилију жетона одговарајућег типа. Овај тип податка обухвата све жељене атрибуте моделираног објекта, а може, ако је потребно да буде и неструктуриран. Места у CPN могу бити и ограниченог капацитета, што се моделира индиректно као и у случају црно-белих PN.

Важно својство CPN је да је догађање прелаза t условљено формирањем повезујућег елемента (*binding element*). Повезујући елемент се може објаснити на следећи начин: Гране у CPN имају свој опис (*arc expression*). Опис гране одређује тачан број и врсту жетона која се паљењем прелаза t ослобађа из излазног места $p \in {}^o t$. Променљиве у опису гране морају бити истог типа као и тип жетона садржаних у излазном месту. Како се у излазном месту може налазити фамилија жетона, паљењем прелаза t се ослобађа конкретан жетон, односно врши се евалуација, тј. конкретизација описа гране. Ова конкретизација, односно избор одговарајућег жетона који ће се ослободити из свих излазних места $p \in {}^o t$ се назива формирање повезујућег елемента.

Појмови који се односе на прелазе и гране су:

- тип променљиве v , који се означава са $Type(v)$;
- тип описа $expr$, који се означава са $Type(expr)$;
- скуп променљивих у опису гране $expr$, који се означава са $Var(expr)$;
- повезивање (*binding*) скупа променљивих V , који свакој променљивој $v \in V$ придружује елемент $b(v) \in Type(v)$. Овим се формира повезујући елемент за који посматрани прелаз t може да се догоди;
- вредност добијена евалуацијом описа гране ($expr$) у повезивању b , која се означава са $expr \langle b \rangle$. За $Var(expr)$ се захтева да буде потскуп од b , а евалуација се врши замењивањем сваке променљиве $v \in V(expr)$ вредношћу $b(v) \in Type(v)$ која је одређена повезивањем. Евалуација се врши за све улазне гране посматраног прелаза t .

Опис гране без променљивих се назива затворени (*closed*) опис. Он може да се евалуира у свим повезивањима, а све евалуације дају исту вредност, која у том случају може заменити опис гране.

Прелазу у CPN могу садржати израз, тј. функцију чији је резултат буловска променљива. Ова функција се назива чување (*guard*). Увођењем ове функције чувања се обезбеђује да одређени повезујући елемент буде омогућен само ако је испуњен услов паљења прелаза, односно ако је вредност функције чувања *true*. Буловски тип променљиве ($\{true, false\}$) означава се са **B**.

Сада је могуће формално дефинисати CPN.

Дефиниција 5 [21]: Обојена PN је деветорка $CPN = (\Sigma, P, T, A, N, C, G, E, M_0)$, где је

- (i) Σ коначан скуп типова (боја),
- (ii) $P = \{p_1, p_2, \dots, p_n\}$ коначан скуп места,
- (iii) $T = \{t_1, t_2, \dots, t_q\}$ коначан скуп прелаза,
- (iv) $A \subseteq (P \times T) \cup (T \times P)$ коначан скуп грана такав да $P \cap T = P \cap A = T \cap A = \emptyset$,
- (v) $N: A \rightarrow (P \times T) \cup (T \times P)$ функција чворова,
- (vi) $C: P \rightarrow \Sigma$ функција типова (боја), (8)
- (vii) G функција чувања, која пресликава T у израз такав да:
 $\forall t \in T: [\text{Type}(G(t)) = \mathbf{B} \wedge \text{Type}(\text{Var}(G(t))) \subseteq \Sigma]$,
- (viii) E опис гране, који пресликава A у израз такав да:
 $\forall a \in A: [\text{Type}(E(a)) = C(p)_{MS} \wedge \text{Type}(\text{Var}(E(a))) \subseteq \Sigma]$, где је p место у $N(a)$
- (ix) M_0 почетна функција, која пресликава P у затворени израз такав да:
 $\forall p \in P: [\text{Type}(M_0(p)) = C(p)_{MS}]$.

Израз (i) дефинише типове, тј. боје садржане у мрежи. Изразима (ii -v) је описана структура мреже, дефинисана са (1). Изразом (vi) је описано да се сваком месту придружује тип жетона који могу да се налазе у том месту. Са (vii) је дефинисана буловска функција чувања којом се задају додатни услови паљења прелаза. Изразом (viii) је описано да у опису грана морају да буду садржане променљиве истог типа који је додељен улазном, односно излазном месту гране. (ix) представља почетно маркирање на мрежи.

Петријева мрежа којом се моделира конкретан процес се углавном не описује на овај формалан начин, већ се директно приказује графом и описима на графу. Када се CPN приказује графички, она се састоји из 3 дела:

- Структура мреже, односно оријентисан, бипартитни граф дефинисан са (1), тј. (ii -v) из дефиниције 5;
- Опис мреже или чвор описа мреже (*description node*), који се графички приказује као правоугаоник у коме се налази опис променљивих, константи и функција који се користе у мрежи;
- Ознаке на мрежи: типови места, описи грана, услови паљења прелаза и почетно маркирање на мрежи.

Пример 5: Процес, који је описан у примерима 2 и 3, се помоћу CPN може моделирати детаљније него што је то урађено мрежама на сликама 11 и 12. Обојеном PN је могуће обухватити више врста сировина, као и потребну количину за сваку од сировина. Да би се то моделирало, потребно је навести ове типове података у опису мреже (слика 15). Овај чвор описа мреже је, као и ознаке на мрежи, написан у програмском језику CPN ML, који је базиран на функционалном програмском језику *Standard Modelling Language* (SML). У примерима се користи овај језик зато што је он интегрисан у програмски пакет *Design/CPN* који је примењен у четвртном делу овог рада. О језику SML и програмском пакету *Design/CPN* ће детаљније бити писано касније.

```

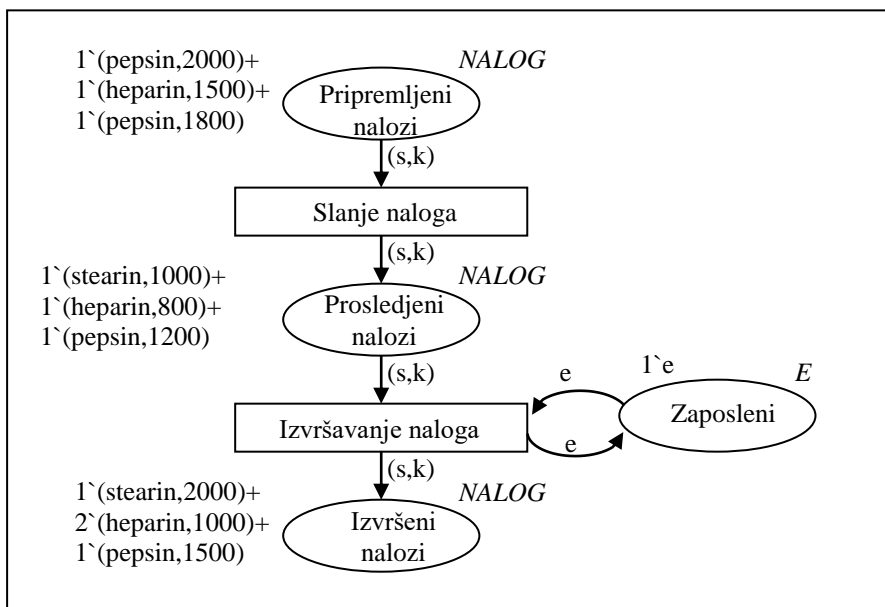
color SIROVINA = with stearin|pepsin|heparin;
color KOLICINA = int;
color NALOG = product SIROVINA * KOLICINA;
color E = with e;
var s: SIROVINA;
var k: KOLICINA;

```

Слика 10. Чвор описа мреже CPN процеса набавке

Тип (боја) података *SIROVINA* из чвора описа мреже се односи на врсту сировине која се наручује, и може имати вредности *stearin*, *pepsin* или *heparin*. То значи да се у моделу могу јавити само ове 3 врсте сировина. Тип *KOLICINA* је *int* (*integer*), тј. може имати само целобројну вредност, а односи се на количину сировине коју треба набавити. Боја *NALOG* моделира налог за набавку и обухвата податке о врсти (називу) сировине и потребној количини.

Променљиве *s* и *k* су типова *SIROVINA* и *KOLICINA*, а сваки жетон који моделира налог за набавку је облика (s,k) . Жетони типа *E* нису структурирани, већ могу да имају само вредност *e*. Овим жетонима се моделирају запослени који извршавају радне налоге. У детаљнијем моделу и ови жетони могу бити структурирани и описани тако да сваки жетон моделира по једног запосленог са његовим битним атрибутима. На основу датог описа процеса и типова података се може формирати CPN на слици 16.



Слика 11. CPN процеса набавке

За разлику од црно-белих PN, у обојеним PN се називи места и прелаза не уписују поред већ у местима и прелазима. Поред места се уписује тип жетона (података) који у том месту могу да се налазе.

Са слике 16 се види да се у местима “Припремљени налози”, “Прослеђени налози” и “Извршени налози” могу наћи само жетони типа *NALOG* односно облика (s,k) , а у месту “Запослени”

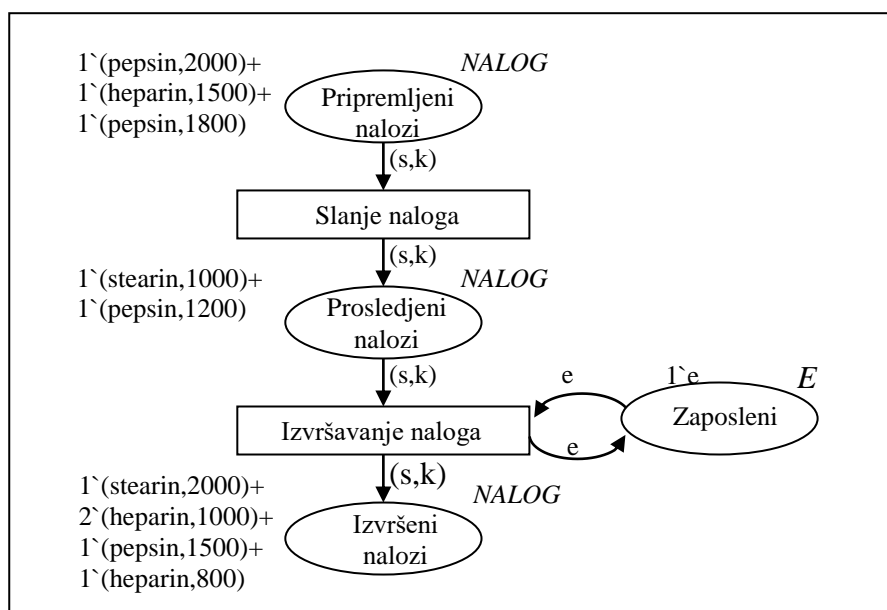
неструктурирани жетони типа *E*. Ово одговара изразу (vi) дефиниције 5. Поред места се уписује и маркирање, односно број и врста жетона у месту, као фамилија жетона.

Почетно маркирање (израз (ix) дефиниције 5) на мрежи на слици 16 је исто као на слици 12, [3,2,1,4], али се у CPN на слици 16 зна и структура ових жетона. Познато је да постоје: 3 припремљена налога за набавку: један за 2000 кг пепсина, други за 1500 кг хепарина и трећи за 1800 кг пепсина; 2 прослеђена налога: један за 1000 кг стеарина, други за 800 кг хепарина и трећи за 1200 кг пепсина; и 4 извршена налога: један за 2000 кг стеарина, два за по 1000 кг хепарина и један за 1500 кг пепсина. Ознака “1`e” поред места “Запослени” указује на то да је, као и у PN на слици 12, за извршење налога задужен само један запослени.

Ознаке на улазним и излазним гранама прелаза “Слање налога” и “Извршавање налога” указују на то да се, паљењем (догађањем) ових прелаза, из њихових улазних места ослобађа по један жетон облика (s,k) , односно типа *NALOG*, а у њиховим излазним местима производи по један жетон таквог типа. Овде треба нагласити да је опис тих грана, у складу са изразом (viii) дефиниције 5, условљен типом података који је тим местима придружен. Ознака “e” на излазној и улазној грани места “Запослени”, на исти начин указује да се троши један неструктурирани жетон (ангажује један запослени) када се догоди прелаз “Извршавање налога”, и поново ослобађа након догађања (паљења) тог прелаза.

Као и у црно-белој PN са слике 12 и овде су оба прелаза омогућена. Као што је раније описано, паљењем (догађање) прелаза је условљено формирањем повезујућег елемента (*binding element*). У случају прелаза “Слање налога” је могуће формирати 3 повезујућа елемента: $s = \text{pepsin}, k = 2000$; $s = \text{heparin}, k = 1500$ и $s = \text{pepsin}, k = 1800$. Прелаз “Слање налога” се може догодити за сваки од ових повезујућих елемената. За догађање прелаза “Извршавање налога” се могу формирати 2 повезујућа елемента: $s = \text{stearin}, k = 1000, e = e$, $s = \text{heparin}, k = 800, e = e$ и $s = \text{pepsin}, k = 1200, e = e$.

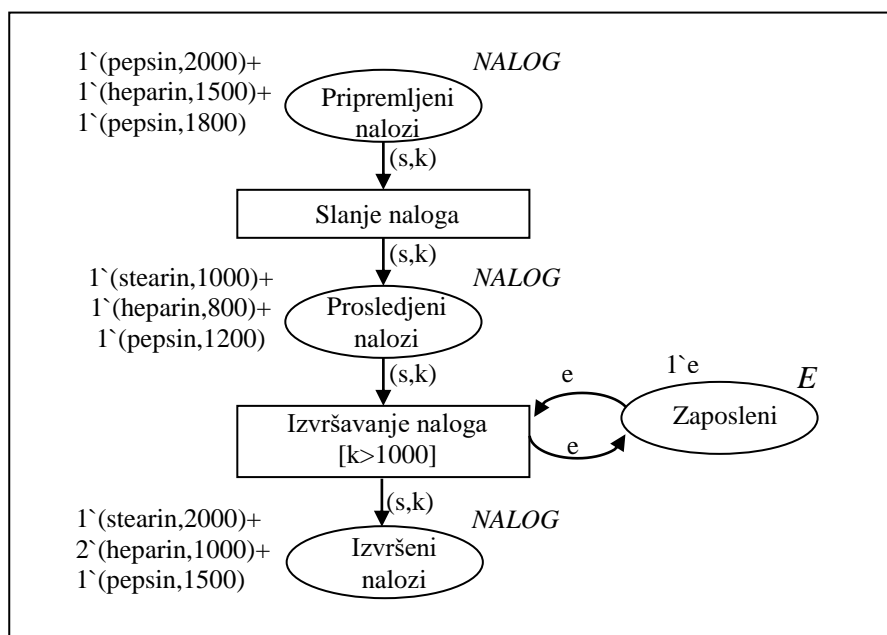
Ако се, као и у примеру 3, догоди прелаз “Извршавање налога” за нпр. повезујући елемент: $s = \text{heparin}, k = 800$, добиће се CPN на слици 17:



Слика 12. CPN процеса набавке после паљења прелаза “Извршавање налога”

Добија се исто маркирање као у црно-белој PN на слици 13, али је у овом случају информација прецизнија: познато је који је налог за набавку извршен.

У дефиницији 5 се налази и израз (vii) који се односи на функцију чувања. Ова функција се може увести у CPN са слике 16, ако се у процесу набавке дода услов да налог за набавку може да се изврши само ако је потребна количина сировине већа од 1000 кг. У том случају би прелазу “Извршавање налога” морала да се придружи функција чувања облика: $k > 1000$. Овим би мрежа са слике 16 постала:



Слика 13. CPN процеса набавке са функцијом чувања

Функција чувања се графички приказује у угластим заградама у самом прелазу на који се односи. Увођењем ове функције прелаз “Извршавање налога” више није омогућен за повезујући елемент: $s = \text{heparin}$, $k = 800$, $e = e$.

2.4. Проширења PN

Проширења PN су последица потреба моделирања конкретних зависности или ситуација у процесима. Тако су нпр. за потребе моделирања празних машина уведене PN са инхибитор гранама [45]. Инхибитор гране моделирају услов да њихов улазни прелаз може да се запали само када је излазно место празно. Због неодређености многих информација о моделираним процесима, уведене су фази PN [4]. Ово проширење PN увођењем елемената фази логике се може реализовати на више начина (фази маркирања која месту придружују фази број као број жетона или функцију припадности неком месту; фази правила паљења прелаз итд.) Да би се моделирале методе одлучивања у процесу одлучивања уведене су објектне PN [14]. У овим мрежама се жетонима моделирају методе одлучивања, где је сваки жетон посебна (објекат) PN која моделира методу. Објекат мреже су жетони веће, системске мреже у којој су места доносиоци одлуке или конкретни проблеми којима се придружују методе решавања проблема.

Листа могућих проширења PN је велика. Овде су детаљније описане хијерархијске, временске и стохастичке PN, зато што су то проширења која су коришћена у моделирању конкретног процеса у четвртом делу рада.

2.4.1. Хијерархијске PN

Сврха увођења хијерархијских мрежа је да се велики модел разложи на већи број мањих који се могу анализирати независно један од другог. Овакво хијерархијско моделирање омогућава приказивање декомпозиције процеса на различитим нивоима детаљности. Начин увођења хијерархијских мрежа је такав да се хијерархијска CPN увек може превести у одговарајућу нехијерархијску, и обрнуто. Хијерархијска и њој еквивалентна нехијерархијска мрежа имају иста својства и исто понашање.

2.4.2. Временске PN

Временске PN су мреже у којима је местима и/или прелазима придружено одређено трајање. Ако је трајање придружено месту p , оно се односи на време које жетон, од момента пристизања у место p паљењем неког прелаза t , треба да проведе у том месту [45]. У литератури је далеко више разматрано трајање које се придружује прелазима. Разлог за то је што места углавном представљају услове или стања, а прелази представљају операције или акције које имају одређено трајање, односно кашњење. Нека је паљењу прелаза t придружено трајање τ , и нека паљење t почиње у тренутку T_0 . Онда се паљење прелаза t састоји у [45]:

- уклањању жетона из свих места $p \in {}^o t$ у тренутку T_0 , и
- додавању жетона у сва места $p \in t^o$ у тренутку $T_0 + \tau$.

Број и врста жетона који се уклањају, односно додају зависи од описа грана која повезују прелаз са местима. Претпоставља се да између тренутака T_0 и $T_0 + \tau$ жетони остају у прелазу t [45].

Ако у PN постоји бар један прелаз са трајањем паљења или бар једно место у коме се жетон задржава одређено време, PN је временска. Уобичајено је, ако у PN постоје оба типа прелаза, да се прелази са трајањем графички приказују правоугаоником, а тренутни прелази линијом.

У обојеним Петријевим мрежама постоје и временски, односно невременски типови (боје) података. У случају временског типа, жетон поред своје стандардне структуре

садржи и временску вредност. Ова вредност представља најранији временски тренутак у коме се жетон може искористити за паљење неког прелаза, тј. тренутак када жетон почиње да постоји. Да би се ово омогућило, уводи се глобални сат који моделира време.

Све претходно доводи до тога да се одређени прелаз може запалити (догодити) када је омогућен и спреман (*ready*). Прелаз је омогућен, као и у случају невременске CPN, када се у његовим улазним местима налази бар онолико жетона колико је задато описом одговарајуће гране. Прелаз је спреман када је временска вредност жетона у улазним местима већа или једнака тренутном времену глобалног сата [21].

Понашање временске CPN се сада може описати на следећи начин: у одређеном моменту глобалног сата пале се сви прелази који су омогућени и спремни, односно, формирају се сви повезујући елементи који се могу формирати; затим се глобални сат помера на први временски тренутак у коме је следећи прелаз омогућен и спреман.

У CPN се трајање графички приказује ознаком @+ иза које стоји константа или израз који одређује кашњење. Ова ознака се додаје опису гране која излази из временског прелаза. Временским жетонима се такође додаје ознака @+ иза које стоји број у угластим заградама, који представља временски тренутак од кога жетон постоји [21]. У чвору описа мреже се мора назначити који је тип података временски. То се ради тако што се, иза описа типа података, односно испред знака “;” дописује *timed*.

2.4.3. Стохастичке PN

Трајање придружено паљењу прелаза може бити детерминистичко и стохастичко. Уколико у PN постоји бар један прелаз чије је трајање паљења стохастичко, мрежа се назива стохастичком и обележава са SPN. У SPN је трајање паљења прелаза случајна променљива која подлеже некој расподели, а време потребно за паљење датог прелаза је различито за свако ново паљење прелаза. SPN имају велику примену у моделирању променљивог трајања операција, што је веома често у системима у којима операције обавља човек или када на трајање операција утичу неки спољни случајни догађаји. Ако у мрежи постоје и тренутни и прелази са стохастичким трајањем, мрежа се назива генерализована SPN и означава са GSPN [12]. Уобичајено је, али не и обавезно, да се тренутни прелази и прелази са стохастичким трајањем графички приказују различито (тренутни линијом, а прелаз са трајањем правоугаоником исл.).

Пример 6: Петријева мрежа са слике 12 је проширена увођењем трајања активности слања и извршавања налога за набавку и увођењем још једног запосленог за обављање активности извршења налога. Опис CPN која моделира овај процес је дат на слици 19.

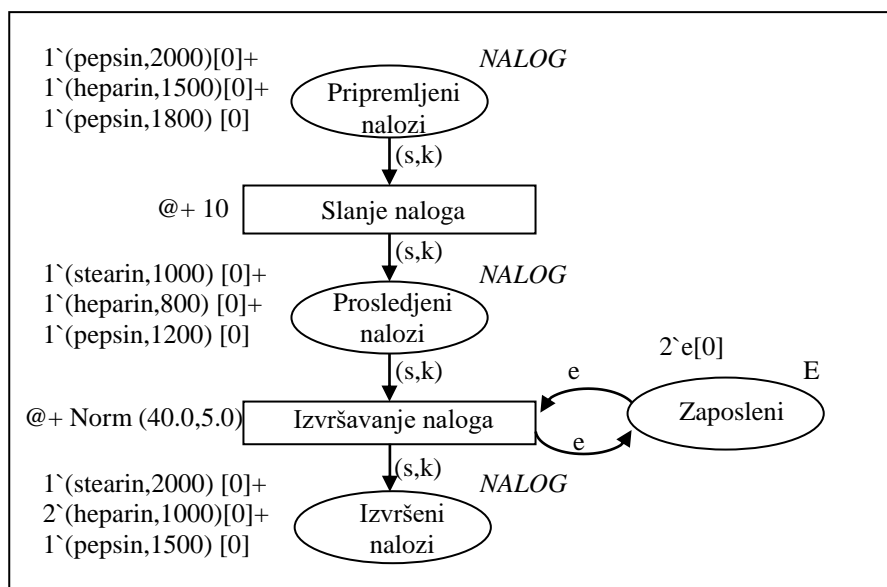
```

color SIROVINA = with stearin|pepsin|heparin;
color KOLICINA = int;
color NALOG = product SIROVINA * KOLICINA timed;
color E = with e timed;
var s: SIROVINA;
var k: KOLICINA;
fun Norm(x,y)=round (normal (x,y));

```

Слика 14. Чвор описа мреже GSPN процеса набавке

Функција $Norm(x,y)$ је уведена зато што трајање извршавања налога за набавку подлеже нормалној расподели. Мрежа којом се моделира процес слања и извршења налога за набавку приказан је на слици 20.



Слика 15. GSPN процеса набавке

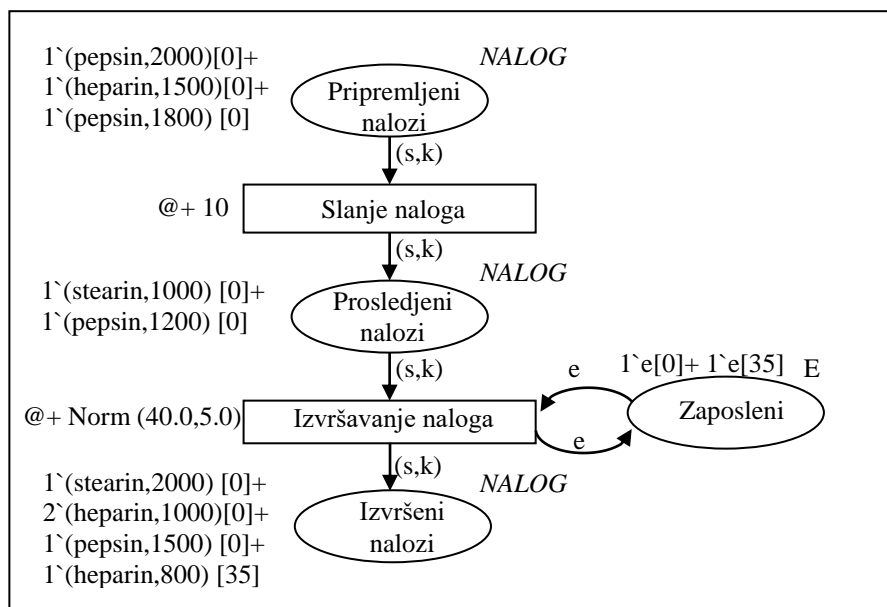
Ознаке $[0]$ поред сваког жетона у свим местима представљају њихову временску вредност. Пошто је у питању почетно маркирање временска вредност свих жетона, као и глобалног сата је 0. Активност “Слање налога” траје 10 временских јединица $[v_j]$, што значи да се, када се тај прелаз запали, одмах уклања један жетон из места “Слање налога”, а тек након 10 v_j се појављује жетон у месту “Извршавање налога”. Пошто нема никаквих ограничења, прелаз “Слање налога” може одмах након првог паљења да се запали за неки нови повезујући елемент.

Трајање активности “Извршавање налога” подлеже нормалној расподели са средњом вредношћу 40 и одступањем од 5 v_j . Трајање ове активности је због тога различито за свако ново паљење прелаза. Једним догађањем прелаза “Извршавање налога” се уклања по један жетон из места “Прослеђени налози” и “Запослени”, а тек након времена $Norm(40, 5)$ се производи по један жетон одговарајућег типа у местима “Извршени налози” и “Запослени”. Одмах након првог паљења је могуће да се овај прелаз још једном догоди. Тиме се ослобађа још по један жетон у местима “Извршени налози” и “Запослени”. Након овог паљења у месту “Прослеђени налози” остаје још један жетон, односно налог који би могао да се извршава, али је место “Запослени” празно. Ово празно место моделира случај када је сваки од двоје запослених ангажован на извршењу по једног налога.

Прелаз “Извршени налози” ће бити омогућен тек када прође прво од трајања $Norm(40, 5)$ придружених повезујућим елементима за које је прелаз био запаљен. Тиме ће се произвести један

жетон у месту “Запослени”. Овим је моделирана чињеница да ће следећи налог за набавку моћи да се извршава тек када један (први) од запослених заврши налог који тренутно извршава, односно када буде слободан. Ако се, као и у примеру 5, догоди прелаз “Извршавање налога” за исти повезујући елемент: $s = \text{heparin}$, $k = 800$, и ако трајање паљења буде нпр. 35 вј. добиће се GSPN на слици 21.

Из маркирања са слике 21 је, док је глобални сат на 0 вј., могуће 3 пута запалити прелаз “Слање налога” и још једанпут прелаз “Извршавање налога”. Жетони $1^{\text{'heparin,800}}$ и $1^{\text{'e}}$ ће постојати у својим местима тек када глобални сат пређе на 35. вј.



Слика 16. GSPN процеса набавке после паљења прелаза “Извршавање налога”

Иако је SPN временска мрежа, постоји велики број радова који је посматрају као посебну класу PN. У [38] и [39] је показано како је k -ограничена PN, са експоненцијалном расподелом паљења прелаза, изоморфна са коначним Марковљевим ланцем. У [29] је дата веза између GSPN и Марковљевих ланаца и дат алгоритам за реализацију Марковљевих процеса, заснован на PN.

2.5. Анализа Петријевих мрежа

Петријеве мреже, нарочито PN нижег нивоа, представљају моћан алат за анализу многих својстава система. Код PN вишег нивоа се многе од метода за анализу не могу уопште или могу применити веома ограничено. У овом поглављу ће прво бити описана нека од својстава PN, а затим две групе метода којима се та својства могу анализирати: стабло десежљивости и инваријанта.

Поред ове две групе метода постоје и методе редукције којима се решава проблем величине модела. За разлику од хијерархијских мрежа које чувају целу мрежу, редукционе методе редукују поједине делове PN модела. Редукција PN је процедура која трансформише PN у њихове редуковане мреже или подмреже које чувају жељена својства оригиналне мреже. Пошто су у овом раду коришћене хијерархијске мреже, о методама редукције неће бити детаљније писано. Један од недостатака ових метода је да се односе на PN које нису временске тј. PN чије се паљење прелаза извршава тренутно [26,39,45].

2.5.1. Својства Петријевих мрежа

Својства PN се могу поделити у две групе: својства понашања и структурна својства

Својства понашања (*behavioural properties*) су својства која зависе од структуре PN и од почетног маркирања. Нека од тих својстава су:

- **Досежљивост** (*Reachability*) Проблем досежљивости се састоји у испитивању да ли се може достићи неко, жељено или нежељено, стање. Када се неки процес моделира помоћу PN ово испитивање се своди на анализу да ли се жељено или нежељено маркирање може достићи у PN. Развијени су бројни општи алгоритми и показано је да је проблем досезања решив, мада он захтева најмање експоненцијално време да би се доказао у општем случају (литература наведена у [39]).
- **Ограниченост** (*Boundedness*) Како места у PN могу представљати бафере са различитим ресурсима (производи, подаци итд.), проверавањем ограничености мреже утврђује се да ли ће доћи до претоваривања бафера. Ограниченост PN је уведена кроз следеће дефиниције [45]:

Дефиниција 6 [45]: Место p Петријеве мреже $PN=(N, M_0)$ је k -ограничено ако број жетона у овом месту никад не прелази k , тј.

$$M(p) \leq k, \forall M \in R(M_0) \quad (9)$$

Место p Петријеве мреже је ограничено ако је k -ограничено за неко целобројно k .

Дефиниција 7 [45]: Петријева мрежа $PN=(N, M_0)$ је k -ограничена ако број жетона у сваком од њених места никад не прелази k , тј.

$$M(p) \leq k, \forall p \in P, \forall M \in R(M_0) \quad (10)$$

Петријева мрежа је ограничена ако је k -ограничена за неко целобројно k .

Дефиниција 8 [45]: Петријева мрежа $PN=(N, M_0)$ је сигурна (*safe*) ако је 1-ограничена, тј.

$$M(p) \leq 1, \forall p \in P, \forall M \in R(M_0) \quad (11)$$

- **Liveness** Ако систем није добро дизајниран може се десити да достигне стање у коме је блокиран. Да би се испитала могућност оваквог стања у PN теорији се користи концепт *liveness*.

Дефиниција 9 [45]: Прелаз t Петријеве мреже $PN=(N, M_0)$ је жив (*live*) ако може бити упаљен из сваког маркирања $M \in R(M_0)$, тј.

$$\forall M \in R(M_0), \exists M' \in R(M) \text{ такво да је } t \text{ допустиво за } M'. \quad (12)$$

Дефиниција 10 [45]: $PN=(N, M_0)$ је жива ако су јој сви прелази живи.

Дефиниција 11 [45]: Маркирање $M \in R(M_0)$ је *deadlock* ако ни један прелаз у мрежи $PN=(N, M_0)$ није допустив из тог маркирања.

Дефиниција 12 [45]: $PN=(N, M_0)$ је *deadlock free* ако ни једно од маркирања $M \in R(M_0)$ није *deadlock*.

Дакле, PN може бити жива или не, а PN која није жива може бити *deadlock free* или не.

- **Повратност и изворно стање** (*Reversibility* и *Home State*) У функционисању система потребно је, када дође до грешке у раду или из неког другог разлога, вратити систем у почетно стање или у неко друго изабрано стање, из кога он даље наставља са радом. Могућност да се системом управља на овај начин се у PN испитује помоћу својства повратности и изворног стања.

Дефиниција 13 [45]: PN је повратна ако је могуће вратити се у почетно маркирање из неког маркирања достижног из почетног маркирања, тј.

$$M_0 \in R(M) \quad \forall M \in R(M_0). \quad (13)$$

Дефиниција 14 [45]: Маркирање M_a у PN са почетним маркирањем M_0 , је изворно стање ако је достижно из било ког маркирања достижног из M_0 , тј.

$$M_a \in R(M) \quad \forall M \in R(M_0). \quad (14)$$

- **Покривеност** (*Coverability*) За маркирање M у PN= (N, M_0) се каже да га је могуће покрити (*coverable*) ако постоји маркирање $M' \in R(M_0)$ такво да је:

$$M'(p) \geq M(p) \quad (15)$$

за свако место p у мрежи [39].

- **Постојаност** (*Persistence*) За мрежу PN= (N, M_0) се каже да је постојана ако, за свака два прелаза, паљење једног неће онемогућити паљење другог прелаза [39].
- **Равноправност** (*Fairness*) У [39] су представљена два основна концепта равноправности: ограничена и безусловна (глобална) равноправност. За два прелаза t_1 и t_2 у мрежи PN= (N, M_0) се каже да су ограничено равноправна (или *B-fair*) ако је максималан број пута, који један од њих може бити упаљен док други није упаљен, ограничен. За мрежу PN= (N, M_0) се каже да је *B-fair* мрежа ако је сваки пар прелаза у мрежи у *B-fair* релацији. За секвенцу паљења σ се каже да је безусловно (глобално) равноправна ако се сваки прелаз у мрежи појављује неограничено често у σ . За мрежу PN= (N, M_0) се каже да је безусловно равноправна мрежа ако је свака секвенца паљења σ из $M \in R(M_0)$ безусловно равноправна.
- **Узајамно искључивање** (*Mutual exclusion*) Посматрају се две компоненте система које имају своја критична стања. Те две компоненте се морају синхронизовати на такав начин да могу да се, ако је потребно, нађу у својим критичним стањима, али никад случајно истовремено [56]. Проблем узајамног искључивања у смислу PN је дизајнирати PN модел који ће испунити ова два захтева.
- **Properly terminating** За PN= (N, M_0) се каже да је *properly terminating* ако се њено извршавање завршава тако да не остане ни један жетон у мрежи [26].

Структурна својства (*structural properties*) зависе само од структуре Петријеве мреже. Она су независна од почетног маркирања, односно важе за свако почетно маркирање. У реалном систему то су својства која систем поседује без обзира на стање у коме се налази. Нека од њих су:

- **Структурна liveness** За PN се каже да је структурно жива ако постоји почетно маркирање M_0 такво да је PN= (N, M_0) жива [45]. Следи да PN која је жива је

структурно жива, али обрнуто не важи. Структурну *liveness* је немогуће директно доказати, већ је могуће само извести потребне услове за структурну *liveness* из неких других структурних својстава.

- **Структурна ограниченост** За PN се каже да је структурно ограничена ако је ограничена за свако коначно почетно маркирање M_0 . [39]
- **Conservativeness** За PN се каже да је строго *conservative* ако је укупан број жетона у мрежи константан $\forall M \in R(M_0)$. Ово својство треба испитати ако је потребно да укупна количина ресурса у систему који се посматра буде константна.
- **Поновљивост (Repetitivity)** За PN се каже да је (парцијално) *repetitive* ако постоји почетно маркирање M_0 и секвенца паљења σ у којој се сваки (неки) прелаз појављује неограничен број пута. [39,45]

2.5.2. Стабло досежљивости

Метода стабла досежљивости (покривања) подразумева пребројавање свих достижних маркирања или свих “покривајућих” маркирања. Она може да се примени на све класе PN, али је, због брзог раста димензије и сложености стабла, ограничена на мреже са малим бројем чворова.

Када је дата $PN=(N,M_0)$, из почетног маркирања M_0 се може добити онолико нових маркирања колико је омогућених прелаза. Из новодобијених маркирања се, паљењем омогућених прелаза добијају нова маркирања, итд. Да би се представила сва маркирања која се могу достићи, користи се **стабло досежљивости (Reachability tree)**. Стабло почиње са почетним маркирањем (корен стабла) и грана се луковима у нова маркирања, за свако омогућено паљење прелаза. [3]. Свако маркирање на мрежи се представља чвором стабла досежљивости. Грана стабла досежљивости која повезује 2 чвора представља прелаз чијим се догађањем прелази из улазног чвора стабла (маркирања PN) у излазни чвор стабла (маркирање PN).

У стаблу досезања се, на различитим и на истим нивоима, могу појавити иста маркирања. За приказивање маркирања PN може се користити и **граф покривања (Coverability graph)**, који се добија из стабла покривања. Чворови графа покривања су чворови стабла покривања који су представљају прво појављивање неког маркирања, а гране графа покривања повезују “суседна” маркирања, односно, ознака гране је прелаз,

почетак гране маркирање у коме се прелаз догађа, а крај гране маркирање које се догађањем прелаза добија.

Називи стабло досежљивости (*Reachability tree*) и граф покривања (*Coverability graph*) се углавном везују за Обичне (црно-беле) PN. Код CPN се у литератури [21] може наћи термин **потпуни граф догађања, О-граф** (*Full Occurrence Graph-O-graph*). О-граф је граф који садржи чвор за свако достижно маркирање у CPN и грану за свако догађање повезујућег елемента. Улазни чвор гране је маркирање у коме се догађа прелаз за дати повезујући елемент, а улазни чвор је маркирање које се тим догађањем добија.

Када све променљиве у описима грана и функцијама чувања CPN припадају типовима (бојама) са коначним бројем могућих вредности, лако се доказује да је О-граф ограничен. Међутим, то најчешће није случај, тако да је проблем О-графа, као и стабла досежљивости, велики број стања посматраног процеса, тј. маркирања посматране CPN. Због тога се уводе две врсте графова којима се покушава да се овај проблем превазиђе: ОЕ-граф и ОС-граф. **Граф догађања са класама еквиваленције, ОЕ-граф** (*Occurrence Graph with Equivalence Classes-OE-graph*) је граф у коме сваки чвор представља класу еквивалентних маркирања. Ове класе еквивалентних маркирања зависе од процеса који се моделира и мора да их дефинише сам корисник. Гране ОЕ-графа представљају повезујуће елементе. Улазни чвор гране је класа која садржи маркирање из ког се може формирати посматрани повезујући елемент. Излазни чвор гране је класа која садржи маркирање које се добија догађањем прелаза за посматрани повезујући елемент. Граф који се на овај начин добија је много мањи од О-графа. **Граф догађања са симетријом, ОС-граф** (*Occurrence Graph with Symetries-OS-graph*) је граф који се може формирати само за CPN у којима постоје симетрична маркирања.

2.5.3. *Инваријанта*

У PN се могу одредити 2 величине: p -инваријанте (*place invariants*) и t -инваријанта (*transition invariants*). Помоћу p -инваријанти се могу анализирати многа својства, нпр. досежљивост, ограниченост, *liveness*, *fairness* итд. t -инваријанте су дуал p -инваријанти, које одређују секвенце догађања прелаза које имају исто почетно и завршно маркирање и помажу у уочавању циклуса у мрежи и могућности враћања у почетно маркирање.

Обе врсте инваријанти се рачунају помоћу матрице инциденције.

Дефиниција 15 [45]: Матрица инциденције $U = [u_{ij}]$, $i = \overline{1, n}$, $j = \overline{1, q}$ чисте (без петљи) $PN=(N, M_0)$ је дефинисана на следећи начин:

$$u_{ij} = \begin{cases} W(t_j, p_i) & \text{ако } t_j \in^0 p_i \\ -W(t_j, p_i) & \text{ако } t_j \in p_i^0 \\ 0 & \text{иначе} \end{cases} \quad (16)$$

где је n број места, q број прелаза у PN , а $W(t_j, p_i)$ тежина гране (t_j, p_i) .

Ова дефиниција матрице инциденције је спој дефиниције матрице инциденције бипартитног графа и дефиниције тежинске матрице (матрице растојања) [7].

Дефиниција 16 [45]: n -вектор Z , ненегативних целих бројева је p -инваријанта ако

$$Z \cdot U = 0 \quad (17)$$

где је U матрица инциденције PN , а n број места у PN .

Дефиниција 17 [45]: q -вектор W , ненегативних целих бројева је t -инваријанта ако

$$U \cdot W^T = 0 \quad (18)$$

где је U матрица инциденције PN , а q број прелаза у PN .

Матрица инциденције се може користити и за формирање једначине стања посматраног процеса. Ова једначина омогућава да се аналитички одреди тражено маркирање у PN . Пре дефинисања једначине стања је потребно увести неке појмове. Нека је дато почетно маркирање M_0 , допуштена секвенца догађања прелаза у PN σ , и **firing count** вектор $V_\sigma = [v_1, \dots, v_q]$, где је v_j број појављивања (догађања) прелаза t_j у секвенци σ . Вектор V_σ се у литератури назива и карактеристични вектор секвенце паљења σ [22].

Дефиниција 17 [45]: Нека је M маркирање које се добија из почетног маркирања M_0 стартовањем секвенце σ . Једначина стања има следећи облик:

$$M^T = M_0^T + U \cdot V_\sigma^T \quad (19)$$

$$\text{или } U \cdot V_\sigma^T = M^T - M_0^T = \Delta M \quad (20)$$

Потребно је нагласити да маркирање, које се одреди помоћу једначине стања (19), не може да се достигне ако секвенца догађања σ није допуштена.

2.5.4. Симулација

Већина примена CPN је намењена испитивању логичке исправности система и процеса, односно њихових динамичких својстава и функционалности. Међутим, CPN се могу користити и за испитивање перформанси као што су нпр. време потребно за извршење циклуса, максимално време извршења неке активности, просечно време чекања на испуњење неког услова, просечан број објеката који чекају да се над њима изврши нека активност итд. За овакву анализу користи се симулација временских CPN. Како трајања активности углавном нису константна већ подлежу некој расподели, најбоље је симулирати генерализоване стохастичке CPN (GSPN).

На основу резултата симулације мере се вредности статистичких променљивих које представљају перформансе које је потребно испитати. Ове статистичке променљиве могу бити невременске (*untimed*) и временске (*timed*) [27,28]. Максимална, минимална, прва, последња, просечна вредност, број појављивања, сума, просечна вредност, варијанса су вредности које се могу мерити помоћу оба типа променљивих. Тренутак првог или последњег појављивања, време извршавања активности и други временски интервали се могу мерити само временским статистичким променљивим.

ЛИТЕРАТУРА

- [1] Bons, R.W.H., Lee, R.M., Wagenaar, R.W., Wrigley, C.D., "Modelling Inter-organizational Trade Procedures Using Documentary Petri Nets", *Proceedings of the Hawaii International Conference on System Sciences*, 1995
- [2] BPR online Learning Center, <http://www.prosci.com>
- [3] Brink, R. S., "A Petri Net Design, Simulation and Verification Tool", A Thesis Submitted for the Degree of Master of Science in Computer Engineering, Rochester Institute of Technology, New York, 1996, <http://www.csh.rit.edu/~rick/thesis>
- [4] Cardoso, J., B. Pradin-Chézalviel, "Logic and Fuzzy Petri Nets", *Manufacturing and PN, XVIII International Conference on Application and Theory of PN*, Toulouse, 1997
- [5] Crowston, K., Osborn, C., "A Coordination Theory Approach to Process Description and Redesign", <http://ccs.mit.edu/ccsmain.html>
- [6] Цветковић, Д., "Теорија графова и њене примене", Научна књига, Београд, 1990
- [7] Цветковић, Д., М. Чангаловић, Ђ. Дугошија, В. Ковачевић-Вујчић, С. Симић, Ј. Вулета, "Комбинаторна оптимизација: Математичка теорија и алгоритми", ДОПИС, Београд, 1996
- [8] Datta, A., "Automating the Discovery of AS-IS Business Process Models: Probabilistic and Algorithmic Approaches", *Information Systems Research*, Vol 29, No 3, pp 275-301, 1998
- [9] Dazzi, L., Fassino, C., Saracco, R., Quaglini, S., Stefanelli, M., "A Patient Workflow Management System Built on Guidelines", *American Medical Informatics Association Annual Fall Symposium (SCAMC)*, 1997
- [10] "Design/CPN Reference Manual", Version 2.0, <http://www.daimi.au.dk/DesignCPN>, 1996,
- [11] Дулановић, Ж., "Менаџмент – Процес организовања", ФОН, Београд, 1996
- [12] General Classification of PN на основу рада Bernardinello, L., and F. DeCindio, "A Survey of Basic Net Models and Modular Net Classes", 1992, <http://www.aut.utt.ro/~mappy/petri>
- [13] Gordon, S., Billington, J.: "Applying Coloured Petri Nets and Design/CPN to an Air-to-Air Missile Simulator" *CPN'98 First Workshop and Tutorial on Practical Use of Coloured Petri Nets and Design/CPN*, Aarhus, Denmark, 1998
- [14] Heitsch, S., Köhler, M., Martens, M., Moldt, D., "High-level Petri Nets for a Model of Organizational Decision Making" *21st International Conference on Application and Theory of Petri Nets*, Aarhus, Denmark, June 26-30, 2000
- [15] *High-level Petri Nets – Concepts, Definitions and Grafical Notation*, Committee Draft ISO/IEC 15909, Version 3.4, 1997, <http://www.petrinets.org>
- [16] *High-level Petri Nets Standard*, Letter Ballot Summary of CD 15909, 1998, <http://www.petrinets.org>
- [17] <http://www.daimi.aau.dk/Cpnets>
- [18] <http://www.exspect.com>
- [19] Janssen, W. Jonkers, H., Verhoosel, J., "What makes business process special? An evaluation framework for modelling languages used in business process redesign", K. Siam, Y. Wand, J. Parsons (eds), *Proc. 2nd CA I SE/IFIP 8.1. International Workshop on Evaluation on Modelling in Systems Analysis and Design*, Barcelona, 1997.
- [20] Jensen, K., Christensen, S., Kristensen, L.M., "Desing/CPN Occurrence Graph Manual", <http://www.daimi.au.dk/DesignCPN>, 1996,
- [21] Jensen K.: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Volume 1,2,3 Springer-Verlag, 1997
- [22] Laftit, S., J.M. Proth, X. Xie, "Optimization of Invariant Criteria for Event Graphs", *IEEE Transactions on Automatic Control*, Vol. 37, No 5, pp 547-555, 1992
- [23] "Концепт и систем текућег планирања у ICN Југославија", Галеника, Београд, Интерни документ, 1998.
- [24] Лазаревић, Б., Нешковић, С., "Методологија моделирања пословних процеса", *YUINFO`98*, Копаоник, 1998
- [25] Лазаревић, Б., Седлар, М., "Реинжењеринг и моделирање пословних процеса", *Саветовање економист Црне Горе*, Херцег Нови, 1998

-
- [26] Lee, K.H., and J. Favrel, "Hierarchical Reduction Method for Analysis and Decomposition of Petri Nets", *IEEE Transactions on Systems, Man, And Cybernetics*, Vol. SMC-15, No 2, pp 272-281, 1985
- [27] Lindstrom, B., Wells, L., "Design/CPN, Performance Tool Manual", <http://www.daimi.au.dk/DesignCPN>, 1999
- [28] Lindstrom, B., Wells, L., "Simulation Based Performance Analysis in Design/CPN", CPN'98, *First Workshop and Tutorial on Practical Use of Coloured Petri Nets and Design/CPN*, 1998
- [29] Lyon, D., "On the Use of Stochastic petri Nets for the Performance of Real-time Nth Order Stochastic Composition", <http://www.docjava.com/pub/document/cmj>
- [30] Макајић-Николић, Д., "Примена технике мрежног планирања у процесу оперативног планирања и реализације оперативног плана", *SYM-OP-IS'98*, pp 251-254, 1998
- [31] Макајић-Николић, Д., Вујошевић, М., "Један приступ процени ризика у оперативном планирању", *SymOrg'00*, pp 523-528, 2000
- [32] Макајић-Николић, Д., Вујошевић, М., "Анализа стабла неисправности за процес оперативног планирања применом расплнутих бројева", Семинарски рад, ФОН, Београд, 2000
- [33] Макајић-Николић, Д., Вујошевић, М., Васић, Б., "Примена обојених Петријевих мрежа у моделирању система одржавања аутобуса", *SYM-OP-IS'01*, pp 449-452, 2001
- [34] Malhotra, Y., "Business Process Redesign: An Overview", *IEEE Engineering Management Review*, Vol26, No3, 1998
- [35] Malone, T.W., Crowston, K., Lee, J., Pentland, B. et al., "Tools for Inventing Organizations: Toward a Handbook of Organizational Processes", *Management Science*, Vol 45, No 3, pp 425-443, 1999
- [36] Милићевић, В., "Менаџмент – Стратегијски менаџмент", ФОН, Београд, 1996
- [37] Mogulothu, V, Ammar, H.H., Lateef, K., Nikzadeh, T., Miao, Z. "Performability Analysis of the Commanding Component of NASA's Earth Observing System" <http://www.csee.wvu.edu/~ammar/vin-aug-14.pdf>
- [38] Molloy, M.K., "Performance Analysis Using Stochastic Petri Nets", *IEEE Transactions on Computers*, Vol. C-31, No 9, pp 913-917, 1982
- [39] Murata, T. "Petri Nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, Vol. 77, No 4, pp 541-580, 1989
- [40] Панић, Б., "Примена Петријевих мрежа у производњи", Дипломски рад, ФОН, Београд, 2001
- [41] Pentland, B.T., "Process Gramars: A Generative Approach to Process Redesign", http://ccs.mit.edu/wp_toc.html
- [42] Pentland, B.T., Osborn, C.S., Wyner, G., Luconi, F., "Useful Descriptions of Organizational Process: Collecting Data for the Process Handbook", http://ccs.mit.edu/wp_toc.html
- [43] Poh. H.L., "Business Process Reengineering: Definition and Models Revisited", <http://www.iscs.nus.sg/~isasia/paper/complete.html>
- [44] Prasad, B., "Hybrid Re-engineering Strategies for Process Improvement", *Business Process Management Journal*, Vol 5, No2, pp 178-198, 1999
- [45] Proth, J.M., and X. Xie, "Petri Nets: A Tool for Design and Management of Manufacturing Systems", John Wiley & Sons, 1996
- [46] Ramos, F., Frausto, J., Camargo, F., "An Expressive Coloured Petri Nets Methodology Applied to a Business to Business Environment", *MOCA'01, Workshop on Modelling of Objects, Components, and Agents*, August 28, 2001
- [47] Schomig, A.K., Rau, H., "A Petri Net Approach for the Performance Analysis of Business Process", *Performance and Dependability Modelling with Stochastic Petri Nets, Seminar @ IBFI, Schloss Dagstuhl, Germany*, 1995.
- [48] Shishkov, B. "Analysis Of Suitability, Appropriateness And Adequacy Of Use Cases Combined With Petri Nets For Business Systems Modeling", *International Conference on Enterprise Information System*, Portugal, 2001
- [49] Simon, K.A., "From Structure to Process: A Vision of a Process Based Organization", *ENTER 95*, Innsbruck, Austria, 1995, <http://www.informatik.gu.se/~kai/pub/structure>
- [50] Simon, K.A., "Towards a Theoretical Framework for Business Process Reengineering", MSc-thesis in Information Systems, in the report-series "Studies in the use of information technology", <http://www.informatik.gu.se/~kai/pub/thesis> 1994
- [51] Spinelli, M., Seidman, A., Popovic, R., Zimmerman, J., Diehl, F.D., "Council Conduct First Management Seminar: Process Management and Reengineering", *Urban Administrator*, Vol 2, No1, 1995
-

-
- [52] Страк, М., Вујошевић, М., Макајић-Николић Д., “Закључивања о стању аутобуса помоћу фази Петријевих мрежа”, *SYM-OP-IS'01*, pp 461-464, 2001
- [53] Straub, P.A., Hurtado, C.L., “Business Process Behavior is (Almost) Free-choice”, *CESA'96, Session on Petri Nets for Multi-Agent and Cooperative Systems*, Lille, France, July 9-12, 1996
- [54] Straub, P.A., Hurtado, C.L., “The simple control property of business process models”, *XV International Conference of the Chilean Computer Science Society*, Arica, Chile, 1995
- [55] Szirbik, N., Wagner, G., “Steps Towards Formal Verification of Agent-based E-Business Applications”, *MOCA '01, Workshop on Modelling of Objects, Components, and Agents*, August 28, 2001
- [56] Törn, A., Материјал за курс о PN на Computer Science Department of Abo Academy University and TUCS Turku, Finland, 1998, <http://dalton.abo.fi/~atorn/petri>
- [57] Tsalgatiodu, A., Louridas, P., Fesakis, G., Schizas, T., “Multilevel Petri Nets for Modeling and Simulating Organizational Dynamic Behaviour”, *Simulation&Gaming*, Vol.27, No 4, pp 484-506, 1996
- [58] van der Aalst, W.M.P. van Hee, K.M., “Business Process Redesign: A Petri-net-based approach”, *Computer in Industry*, Vol 29, No 1-2, pp 15-26, 1996.
- [59] Villapol, M.E., Billington, J., “Modelling and Initial Analysis of the Resource Reservation Protocol using Coloured Petri Nets” *21st International Conference on Application and Theory of Petri Nets*, Aarhus, Denmark, June 26-30, 2000
- [60] von Uthman, C., “Improving Petri Nets Fitness for Use in Business Process Modelling”, Draft paper, <http://www.wi.uni-muenster.de/is/mitarbeiter/ischut/pn>, 1999.
- [61] Vujošević, M., Makajić-Nikolić, D., Vasić, B., “An Application of Coulored Petri Nets to Modeling of Bus Maintenance System”, *EURO 2001, The European Operational Research Conference 9-11 July 2001*, Rotterdam, The Netherlands, www.euro2001.org
- [62] Vujošević, M., “Операциона истраживања: изабрана поглавља”, Факултет организационих наука, Београд, 1999.
- [63] Vuksic, V.B., Giaglis, G.M., Hlupic, V., “IDEF Diagrams and Petri Nets for Business Process Modelling: Suitability, Efficacy, and Complementary Use”, *IECIS 2000, International Conference on Enterprise Information Systems*, Stafford, pp 242-247, 2000
- [64] Weber H., Sunbul A., Padberg J., “Evolutionary Development of Business Process Centered Architectures Using Component Technologies”, *Fifth International Conference on Integrated Design and Process Technology, IDPT 2000*
- [65] Wikarski D., “An Introduction to Modular Process Nets”, Technical Report, <http://www.icsi.berkeley.edu/techreports>, 1996.
- [66] William, J.K., Teng, J.T.C., “Business Process Change. A Study of Methodologies, Technique and Tools”, *Management Information System Quaterly*, V.21, N.1, 1997.
- [67] Wyner, G.M., Lee, J., “Defining Specialization for Process Models” *Center for Coordination Science @ MIT, 2001 Technical Reports and Working Papers*, CCS No. 216, Sloan No. 4159, 2001
- [68] Yu, E.S.K., “Models for Supporting the Redesign of Organizational Work”, *Proceedings Conf. On Organizational Computing Systems (COOCS'95)*, Milpitas, California, USA, 1995
- [69] Yu, E.S.K., Liu, L., “Modelling Trust in the i* Strategic Actors Framework”, http://micmac.mitel.com/conferences/2001/proceedings/Eric_Yu_MICON_Proceedings-ER.pdf, 2001