

SOME POSSIBILITIES OF MODELING COLORED PETRI NETS

GOHARIK PETROSYAN

National Academy of Sciences of the Republic of Armenia,
International Scientific-Educational Center,
PhD in Physical and Mathematical Sciences, Associate Professor
petrosyan_gohar@list.ru, petrosyangoharik72@gmail.com

ARMEN GABOUTCHIAN

Moscow State University of Medicine and Dentistry named after A.I. Evdokimov,
PhD in Medical Sciences
armengaboutchian@mail.ru

VLADIMIR KNYAZ

FSUE State Scientific Research Institute of Aviation Systems (GosNIIAS),
PhD in Technical Sciences,
knyaz@gosniias.ru

<https://doi.org/10.52853/25792903-2021.2-gpag>

Abstract

Petri nets are a mathematical apparatus for modelling dynamic discrete systems. Their feature is the ability to display parallelism, asynchrony and hierarchy. First was described by Karl Petri in 1962 [1,2,8]. The Petri net is a bipartite oriented graph consisting of two types of vertices - positions and transitions connected by arcs between each other; vertices of the same type cannot be directly connected. Positions can be placed by tags (markers) that can move around the network. [2] Petri Nets (PN) used for modelling real systems is sometimes referred to as Condition/Events nets. Places identify the conditions of the parts of the system (working, idling, queuing, and failing), and transitions describe the passage from one state to another (end of a task, failure, repair). An event occurs (a transition fire) when all the conditions are satisfied (input places are marked) and give concession to the event. The occurrence of the event entirely or partially modifies the status of the conditions (marking). The number of tokens in a place can be used to identify the number of resources lying in the condition denoted by that place [1,2,8].

Coloured Petri nets (CPN) is a graphical oriented language for design, specification, simulation and verification of systems [3-6,9,15]. It is in particular well-suited for systems that consist of several processes which communicate and synchronize. Typical examples of application areas are communication protocols, distributed systems, automated production systems, workflow analysis and VLSI chips.

In the Classical Petri Net, tokens do not differ; we can say that they are colourless. Unlike standard Petri nets in Colored Petri Net of a position can contain tokens of arbitrary complexity, such as lists, etc., that enables modelling to be more reliable.

The article is devoted to the study of the possibilities of modelling Colored Petri nets. The article discusses the interrelation of languages of the Colored Petri nets and traditional formal languages. The Venn diagram, which the author has modified, shows the relationship between the languages of the Colored Petri nets and some traditional languages. The language class of the Colored Petri nets includes a whole class of Context-free languages and some other classes. The paper shows modelling the task synchronization Patil using Colored Petri net, which can't be modeled using well-known operations P and V or by classical Petri network, since the operations P and V and classical Petri networks have limited mathematical properties which do not allow to model the mechanisms in which the process should be synchronized with the optimal allocation of resources.

Keywords and phrases: Petri Nets, Colored Petri Nets, Traditional Languages, Transition, Position.

ՊԵՏՐԻԻ ԳՈՒՆԱՎՈՐ ՑԱՆՑԵՐԻ ՄՈԴԵԼԱՎՈՐՄԱՆ ՈՐՈՇ ՀՆԱՐԱՎՈՐՈՒԹՅՈՒՆՆԵՐԸ

ԳՈՀԱՐԻԿ ՊԵՏՐՈՍՅԱՆ

ՀՀ Գիտությունների ազգային ակադեմիայի
Գիտակրթական միջազգային կենտրոն,
Ֆիզիկամաթեմատիկական գիտությունների թեկնածու, դոցենտ
petrosyan_gohar@list.ru, petrosyangoharik72@gmail.com

ԱՐՄԵՆ ԳԱՐՈՒՉՅԱՆ

Ա. Ի. Եվդակիմովի անվան Մոսկվայի պետական թշկա-ստոմատոլոգիական
համալսարան,
թշկական գիտությունների թեկնածու
armengaboutchian@mail.ru

ՎԼԱԴԻՄԻՐ ԿՆՅԱՉ

Ավիացիոն համակարգի պետական հետազոտական ինստիտուտ (ԱՀՊՀԻ),
տեխնիկական գիտությունների թեկնածու,
knyaz@gosniias.ru

Համառոտագիր

Պետրիի ցանցը մաթեմատիկական սարք է՝ դիսկրետ դինամիկ համակարգերի մոդելավորման համար: Նրա առանձնահատկություններից են զուգահեռությունը, անհամաժամանակությունը և հիերարխիան ներկայացնելու ունակությունը: Առաջին անգամ նկարագրվել է Կարլ Պետրիի կողմից 1962 թվականին **[1,2,8]**: Պետրիի ցանցը երկկողմանի կողմնորոշված գրաֆ է, որը բաղկացած է երկու տեսակի գագաթներից՝ դիրքեր և անցումներ, որոնք միմյանց հետ կապված են աղեղներով: Նույն տեսակի գագաթները չեն կարող ուղղակիորեն միացվել: Դիրքերը կարող են պարունակել նշիչներ (մարկերներ), որոնք կարող են տեղաշարժվել ցանցի ներսում **[2]**: Պետրիի ցանցերին (PN), որոնք օգտագործվում են իրական համակարգերի մոդելավորման համար, երբեմն անվանում են պայմանների/իրադարձությունների ցանցեր: Դիրքերը նկարագրում են համակարգի վիճակները, պայմանները (աշխատանքային, անգործուն, հերթագրում և ձախողում), իսկ անցումներով նկարագրվում են համակարգում տեղի ունեցող գործողությունները՝ անցում մի պայմանից մյուսը (առաջադրանքի ավարտ, ձախողում, նորոգում և այլն): Իրադարձությունը տեղի է ունենում (անցումը կատարվում է), երբ բավարարվում են բոլոր պայմանները (առկա են համապատասխան քանակով նշիչներ մուտքային դիրքերում) և թույլատրելի է անցման կատարումը: Իրադարձության տեղի ունենալը ամբողջությամբ կամ մասամբ փոփոխում է համակարգի վիճակների, պայմանների կարգավիճակը: Դիրքերում առկա նշիչների քանակով է պայմանավորված համակարգի վիճակին համապատասխան ռեսուրսների ծավալի որոշումը **[1,2,8]**:

Պետրիի գունավոր ցանցը (CPN) գրաֆիկական կողմնորոշված լեզու է՝ համակարգերի նախագծման, ճշգրտման, սիմուլյացիայի և ստուգման համար **[3-6,9,15]**: Այն մասնավորապես լավ է համապատասխանում այն համակարգերին, որոնք բաղկացած են մի շարք գործընթացներից, որոնք տվյալներով փոխանակվում և սինխրոնացվում են: Կիրառման բնագավառների տիպիկ օրինակներն են հաղորդակցման արձանագրությունները, բաշխված համակարգերը, ավտոմատացված արտադրական համակարգերը, աշխատանքային հոսքի վերլուծությունը և VLSI չիպերը:

Դասական Պետրիի ցանցերի նշիչները միմյանցից չեն տարբերվում, կարելի է ասել, որ դրանք անգույն են: Ի տարբերություն Պետրիի ստանդարտ ցանցի՝ Պետրիի գունավոր ցանցի դիրքը կարող է պարունակել կամայական բարդության նշիչներ, օրինակ՝ ցուցակներ և այլն, ինչը հնարավորություն է տալիս մոդելավորումն ավելի հուսալի դարձնել:

Հոդվածը նվիրված է Պետրիի գունավոր ցանցերի մոդելավորման հնարավորությունների ուսումնասիրությանը: Հոդվածում քննարկվում են Պետրիի գունավոր ցանցերի լեզուների և ավանդական ձևական լեզուների փոխադարձ կապը: Հեղինակների կողմից փոփոխված Վեննի դիագրամը ցույց է տալիս Պետրիի գունավոր ցանցերի լեզուների և որոշ ավանդական լեզուների միջև փոխադարձ կապը, որտեղ Պետրիի գունավոր ցանցերի լեզուների դասը ներառում է համատեքստից անկախ լեզուների ամբողջ դասը և որոշ այլ դասեր: Հոդվածում ցույց է տրված, որ Պատիլի համաժամանակության խնդիրը մոդելավորվում է Պետրիի գունավոր ցանցի միջոցով, որը հնարավոր չէր մոդելավորել P և V հայտնի գործողությունների կամ Պետրիի դասական ցանցի միջոցով, քանի որ P և V գործողությունները և Պետրիի դասական ցանցերն օժտված չեն համապատասխան մաթեմատիկական հատկություններով, այսինքն՝ չեն կարող մոդելավորել այն մեխանիզմները, որոնցում գործընթացը պետք է համապատասխանեցվի ռեսուրսների օպտիմալ բաշխման հետ:

Բանալի բառեր և բառակապակցություններ. Պետրիի ցանցեր, Պետրիի գունավոր ցանցեր, ավանդական լեզուներ, անցում, դիրք:

НЕКОТОРЫЕ ВОЗМОЖНОСТИ МОДЕЛИРОВАНИЯ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ

ГОАРИК ПЕТРОСЯН

Национальная академия наук Республики Армения
Международный научно-образовательный центр
кандидат физико-математических наук, доцент
petrosyan_gohar@list.ru, petrosyangoharik72@gmail.com

АРМЕН ГАБУЧЯН

Московский государственный медико-стоматологический
университет имени А.И. Евдокимова,
кандидат медицинских наук
armengaboutchian@mail.ru

ВЛАДИМИР КНЯЗЬ

Государственный научно-исследовательский институт
авиационных систем (ГосНИИАС),
кандидат технических наук
knyaz@gosniias.ru

Аннотация

Сети Петри - это математический аппарат для моделирования динамических дискретных систем. Их особенность - возможность отображать параллелизм, асинхронность и иерархию. Впервые описана Карлом Петри в 1962 г. [1,2,8]. Сеть Петри - двудольный ориентированный граф, состоящий из двух типов вершин - позиций и переходов, соединенных дугами между собой; вершины одного типа не могут быть напрямую связаны. В позиции могут быть размещены метки (маркеры), которые могут перемещаться по сети [2]. Сети Петри, используемые для моделирования реальных систем, иногда называют сетями условий / событий. Позиции определяют состояния частей системы (работа, простой, очередь и отказ), а переходы описывают переход от одного состояния к другому (завершение задачи, отказ, ремонт ...). Событие происходит (срабатывание перехода), когда выполняются все условия (входные позиции содержат соответствующее количество маркеров) и выдается разрешение возникновению события. Возникновение события полностью или частично изменяет статус условий (маркировка). Количество маркеров в позициях может использоваться для определения количества ресурсов, находящихся в состоянии, обозначенном этой позицией [1,2,8].

Раскрашенные сети Петри (CPN) – это графически ориентированный язык для проектирования, спецификации, моделирования и проверки систем [3-6,9,15]. Он особенно хорошо подходит для систем, которые состоят из ряда процессов, которые обмениваются данными и синхронизируются. Типичными примерами областей применения являются протоколы связи, распределенные системы, автоматизированные производственные системы, анализ рабочего процесса и микросхемы VLSI.

В классической сети Петри маркеры не отличаются друг от друга, можно сказать, что они бесцветные. В отличие от стандартных в раскрашенных сетях Петри позиции могут содержать маркеры произвольной сложности, например, списки и т. д., что позволяет моделировать более надежно.

Статья посвящена исследованию возможностей моделирования раскрашенных сетей Петри. В статье обсуждается взаимосвязь языков раскрашенных сетей Петри и традиционных формальных языков. Диаграмма Венна, модифицированная автором, показывает взаимосвязь между языками раскрашенных сетей Петри и некоторыми традиционными языками, где класс языков раскрашенных сетей Петри включает в себя целый класс контекстно-свободных языков и некоторые другие классы. В статье

показано моделирование задачи синхронизации Пати с использованием раскрашенных сетей Петри, которую невозможно смоделировать с помощью хорошо известных операций Р и V или классической сети Петри, поскольку операции Р и V и классические сети Петри имеют ограниченные математические свойства, которые не позволяют моделировать механизмы, в которых процесс должен быть синхронизирован с оптимальным распределением ресурсов.

Ключевые слова и словосочетания: сети Петри, раскрашенные сети Петри, традиционные языки, переход, позиция.

Introduction (*Heading 1*)

The formal definition of a Classical Petri Nets:

Petri Net $M = (C, \mu)$ pair, where $C = (P, T, I, O)$ is the network structure, and μ is the network condition. In structure C of a P -positions, T -transitions are finite sets. $I: T \rightarrow P^\infty$, $O: T \rightarrow P^\infty$ is the input and output functions, respectively, where P^∞ are all possible collections (repetitive elements) of P . $\mu: P \rightarrow N_0$ is the function of condition, where $N_0 = \{0, 1, \dots\}$ is the set of integers.

We determine (in a known manner) the allowed transitions of Petri Nets and the transitions from one state to another, as well as the set of reachable states [1,2].

Definition. The mathematical definition of Colored Petri Net: CPN is a nine-tuple $CPN = (\Sigma, P, T, A, N, C, G, E, I) : [7, 9, 15]$

- (i) Σ is a finite set of non-empty types, called colour sets,
- (ii) P is a finite set of places,
- (iii) T is a finite set of transitions,
- (iv) A is a finite set of arcs such that:
 - $P \cap T = P \cap A = T \cap A = \emptyset$
- (v) N is a node function. It is defined from A into $P \times T \cup T \times P$
- (vi) C is a colour function. It is defined from P into Σ .
- (vii) G is a guard function. It is defined from T into expressions such that:
 - $\forall t \in T : [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \Sigma]$.
- (viii) E is an arc expression function. It is defined from A , by expressions such that:
 - $\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$
 where $p(a)$ is the place of $N(a)$.
- (ix) I is an initialization function. It is defined from P into closed expressions such as:
 - $\forall p \in P : [Type(I(p)) = C(p)_{MS}]$.

Colored Petri net consists of the following components:

- The ellipses and the circles are called places. They describe the states of the system (buffers).
- The rectangles are called transitions. They describe the actions (processes).
- The arrows are called arcs. The arc expressions describe how the state of the CPN changes when the transitions occur.
- Each place contains a set of markers called tokens, each of these tokens carries a data value, which belongs to a given type.

In the CPN Tool, this is represented as initial marking next to the associated place. The distribution of tokens, called marking, in the places of a CPN determines the state of a system being modelled. The dynamic behaviour of a CPN is described in terms of the firing of transitions. The firing of a transition takes the system from one state to another. A transition is enabled if the associated arc expressions of all incoming arcs can be evaluated to a multi-set, compatible with the current tokens in their respective input places, and its guard is satisfied. An enabled transition may fire by removing tokens from input places specified by the arc expression of all the incoming arcs and depositing tokens in output places specified by the arc expressions of outgoing arcs [9].

The modelling of Context-Free languages with Colored Petri Nets. The diagram of the Interrelation of Colored Petri Nets and Traditional Languages

One of the most studied simple classes of formal languages is the class of Regular languages. It is known that any Regular language is the language of Petri Nets. [1,10] It's possible to construct Petri Net, which generates $\{a^n b^n / n > 1\}$ a Context-free language, which is not Regular. [1] Not all the languages of Petri Nets are Context-free; build a network that generates $\{a^n b^n c^n / n > 0\}$ a Context-sensitive language, which is not Context-free language. [1] Unlike Regular languages, which are the languages of Petri Nets, there are Context-free languages, which are not languages of Petri Nets. Such examples of Context-Free languages are noted in the following: $\{\omega \omega^R / \omega \in \Sigma^*\}$, $L^* = L \cup LL \cup LLL \dots$ (in particular, $\{a^n b^n / n > 1\}$). This fact illustrates the limitation of Petri Net as a tool that generates the languages. [1]

In Petri Nets, it is not possible to remember a long sequence of arbitrary characters. In Petri Nets, the sequence of limited length can be remembered (this is also possible in finite automata). [1]

However, Petri Nets do not have the "capacity of pushdown memory" necessary for the generation of Context-free languages. The interrelation of the languages of Petri Nets with other classes of languages investigated Venn is shown in Figure 1 in the form of a diagram. [1]

Figure 2 shows a Colored Petri Net, which generates the L^* language that is, Colored Petri Net is a more powerful tool than the classical Petri Net. To understand

types of data used in a figure, it is necessary to give a declaration. Figure 2 introduces two positions of count type and marked as first and second. In the figure, two transitions are marked with the symbol **a** that generates symbol **a**, and a transition marked with the symbol **b**, which generates the symbol **b**.

In the figure, the position of the count of type remembers the number of transitions are fired and regulates. Hence, the number of appearances a symbol **b** was equal to the number of appearances a symbol **a**.

In fact, when the marked with a transition is fired, it generates the symbol **a**; if the marked with **b** transition is fired, it generates the symbol **b**. To the transitions are attached logical expressions (guards): $ct > 0$, $ct > (n - 1)$, if the logical expression is true, then the transition is allowed, and if false, then the transition is not allowed. If the first position of count of type value of token is equal to one, the marked with a first transition is fired. The value of **n** must be fixed in advance.

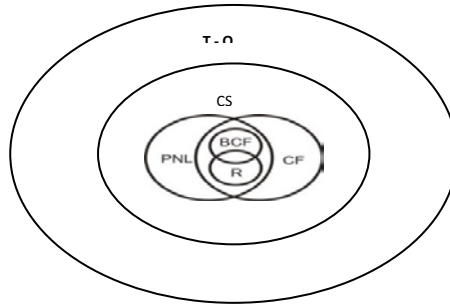


Figure 1. Interrelation of Petri Nets and Traditional Languages (T-0-the General type of languages, CS-Context- sensitive languages, PNL-Petri nets languages, CF-Context- free languages, BCF-bonded Context- free languages, R-Regular languages).

Let $n = 2$, then is fired marked with **a** the first transition, and $ct = 2$, then is fired marked with **a** the second transition, are generated by **aa** symbols, in this case second position of count of type value of token is equal to two: count= 2, and twice is fired marked with **b** the transition, are generated by **bb**, when the value of the first counter is equal to one, cycle will be repeated, etc.

Many properties of Colored Petri nets, as logical expressions, types of markers, the expression of the arcs, etc., which are used to control the transition firing [3].

In Figure 2 Colored Petri Net is constructed for the given language, which supposes the following interrelation of languages of Colored Petri Net with some of traditional languages classes (see Figure 3).

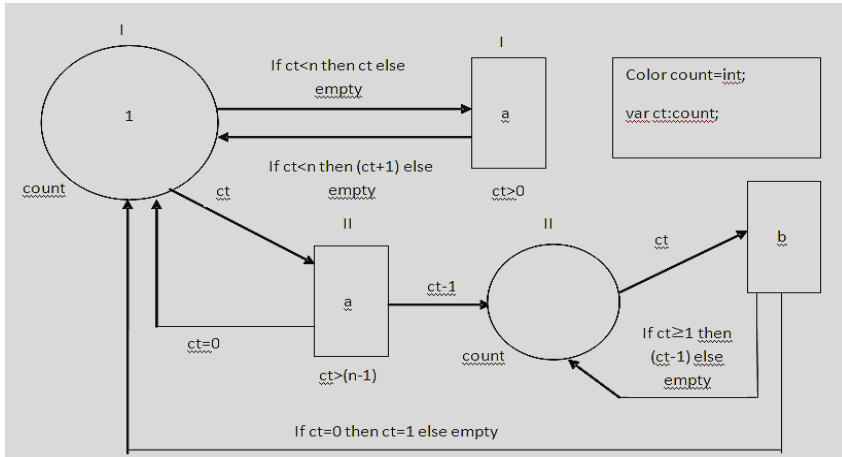


Figure 2. Modelling $L^* = L \cup LL \cup LLL \dots$ language by Colored Petri Net.

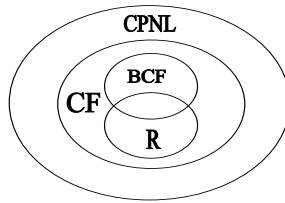


Figure 3. Interrelation of Colored Petri Nets and Traditional Languages. CPNL (Language of Colored Petri Net).

Simulation of the synchronization problem using Colored Petri nets

In 1971 Patil proved that P- and V- actions have insufficient capacity for resolving synchronization issues. His proposed solution to a model problem is called smoking a cigarette. The problem consists of four processes that model the agent's states and actions and three smokers. Each smoker continuously makes a cigarette and smokes it. To make and smoke a cigarette, we need three components: tobacco, paper and matches. One of the smokers always has paper, the second smoker has tobacco, the third has matches. Thus, the agent has an unlimited number of ingredients. The agent puts the two components on the table, the smoker who has the third ingredient, can make the cigarette and then he signals the agent. In this case, the agent offers the other two one components necessary for the smoker, and the action is repeated [10].

The actions of the smokers without coordination are as follows.

Let X - the smoker with tobacco, Y - the smoker with paper, Z - the smoker with matches, A-agent (see Table 1). The smokers' problem is, to define some additional semaphores and processes, if necessary, and introduce necessary P and V statements in

these processes to attain the necessary cooperation among themselves required to ensure continued smoking of cigarettes without reaching a deadlock.

However, there is a restriction that the process that supplies the ingredients cannot be changed and that no conditional statements may be used. The first restriction is placed because the smokers are seeking cooperation among themselves and therefore should not change the supplier, and the second restriction is justified because P and V primitives were introduced to avoid having to coordinate processes by repeatedly testing a variable until it changes its value, and because the operation of making and smoking a cigarette has no conditional actions. It will be seen that the cigarette smokers' problem has no solution.

Patil showed that there is no sequence of P- and V- actions and can't solve the problem correctly. [1,2] If we try to model the problem with the Classical Petri Net, we will get a non-active net. Also, since all tokens in Classical Petri Nets have the same type, ingredients will not differ.

The author modelled the problem with Colored Petri Net (see Figure 4), which gives the possibility of the number of modelling for solving similar problems due to several properties, such as the existence of the attached data types, functions, transitions. [3-10,14-15]

Position A presents the agent status, which has three types of tokens: (p, m) - (p-paper, m-match), (t, m) - (t-tobacco, m-match), (p, t) - (p-paper, t- tobacco), with the help of an organized net of a cycle, it is possible to constantly offer 2 different ingredients to the smokers.

In position A and also in several of positions in the net, E type is attached, that is a combination of 3 types, which means the combination can be any of these three types, and each of which is a Cartesian product: either with N and Q, U and Q, or N and U. (k,l) the pair of variables is a Cartesian product of E-type, that is either (p,m), or (t,m) or (p,t). C', D', F', C, D, F positions have E type, where C, D, F transport the relevant ingredients to the relevant smokers. C', D', F' positions give ingredients to the agent again, so the cycle is repeated. On the top of the position, we put the first mark, which means the net is initialized. If on the top of the position there is the "-" sign, it means that the position is empty or it contains no token. The type is marked in italic letters below the position. The **ct** position is connected to **T** transition. It allows making **T** transition three times per cycle. **S**, **P**, **M** positions respectively have **U**, **N**, **Q** types. They present the state of the three smokers who respectively own tobacco, paper and matches. **T4** transition would occur only if all three smokers finished smoking.

Now we will describe the net performance. **T** will occur the first transition, as there are the corresponding marks in **A** and **ct** positions. After **T** firing transition, we check if (k,l) = (p,m), token is for **C** position, and the other two (**D**, **F**) positions do not get any marks. If (k,l) = (p,m), then **T1** transition can be occurred, which describes the process of the first smoker, and (p,m) ingredients give **C'** position. **T**

transition may occur in this condition, which will move (t, m), or (p, t) tokens and **T2**, or **T3** transitions will occur.

Suppose **T1**, **T2**, **T3** transitions occur, or the agent has proposed the three types of ingredients, and the three smokers have finished smoking. In that case, the **T4** transition will occur, the agent will offer three ingredients again, and the cycle will repeat.

If we try to present this problem with Classical Petri Net, then we need to use three transitions instead of **T** transition. That also means the minimization problem of the net is also solved, which implies cost reduction due to the reduction of arcs in positions and transitions.

Declaration

Color
INT = integer;
 Color **U** = t;
 Color **N** = P;
 Color **Q** = m;
 Color
E = {Product N*Q
 OR Product U*Q
 OR Product N*U};
Var(K,I):E;
n:INT;

Processes A_x	Processes A_y	Processes A_z
pick up the paper pick up the match roll the cigarette light the cigarette smoke the cigarette return to A_x	pick up the tobacco pick up the match roll the cigarette light the cigarette smoke the cigarette return to A_y	pick up the tobacco pick up the paper roll the cigarette light the cigarette smoke the cigarette return to A_z

Table 1. The actions of the smokers.

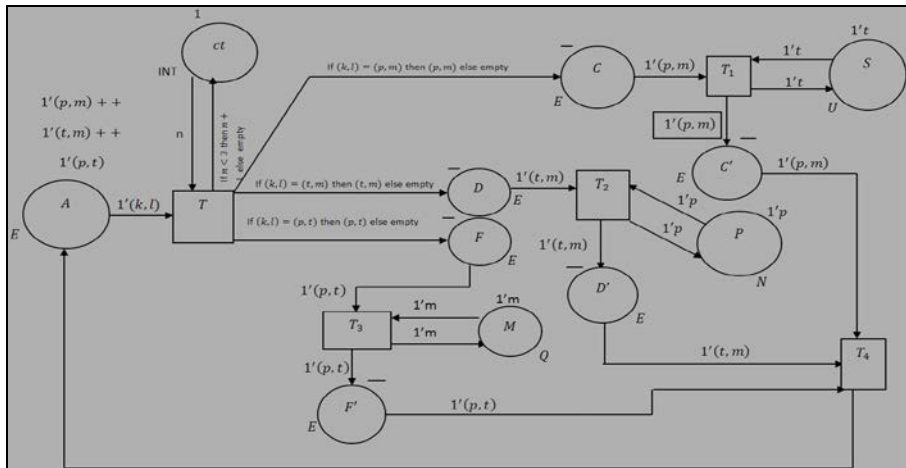


Figure 4. The modelling of Cigarette smoker's problems with Colored Petri Nets.

Conclusion

We conclude that Colored Petri Nets is ideal for simulating real systems, the study of its elements, the system's verification, the system's validation, detection and prevention in emergencies, and the system's individual components optimal variants for replacement. Furthermore, before creating a similar system and implementing the above process, Colored Petri Nets is a convenient system of imitation in process optimization and cost minimization.

REFERENCES

- [1] Peterson J., Petri Net Theory and the Modelling of Systems. Prentice-Hall. ISBN 0-13-661983-5 (1981).
- [2] Tadao M., "Petri nets: Properties, Analysis and Applications." Proc. of the IEEE, 77(4), 1989.
- [3] Jensen K. and Rozenberg G., Eds., High-Level Petri Nets. Theory and Application, Springer-Verlag, Berlin, 1991, pp. 44-122.
- [4] Jensen K., Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer - Verlag, Berlin, 1992.
- [5] Jensen K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer, 1996. Vol. 1–3.
- [6] Jensen K., "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use", Basic Concepts. Monographs in Theoretical Computer Science, Springer – Verlag, Berlin, Germany, V.1,2, 3, 1997.
- [7] Ullman J., "Elements of ML Programming," Prentice-Hall, Upper Saddle River, 1998.
- [8] Reising W., Rozenberg G. (eds), Lecture Notes on Petri Nets. Parts I and II // Lecture Notes in Computer Sciences. V. 1491 – 1492. Springer – Verlag, 1998.
- [9] Jensen K. and Kristensen L., Coloured Petri Nets - Modeling and Validation of Concurrent Systems. Springer-Verlag Berlin, 2009.
- [10] Alfred A., Jeffrey U., Theory of Parsing, Translation, & Compiling Prentice-Hall, January 1, 1973, v. 1,2.
- [11] Knut D., The Art of Programming, v. 1-3, Moscow, Mir 1976.
- [12] Orlov S., Technology of Software Development, a textbook for Universities, Petersburg, 2002.
- [13] Gordeev A., Molchanov A., System Software, textbook, St. Petersburg, 2002
- [14] Westergaard M. and. Kristensen L. The Access/CPN Framework: A Tool for Interacting with the CPN Tools Simulator. Proc. of 30th International Conference on Applications and Theory of Petri Nets (Petri Nets 2009). Lecture Notes in Computer Science 5606, pp. 313-322, Springer-Verlag Berlin, 2009.
- [15] Jensen K., Kristensen L., and Wells L. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. International Journal on Software Tools for Technology Transfer (STTT)9(3-4), pp. 213-254, 2007.

Հոդվածը ներկայացվել է տպագրության 03.03.2021,
ընդունվել է տպագրության 15.04.2021