Comparative Analysis of Some Modeling Capabilities of Classical Petri Nets and Colored Petri Nets

Petrosyan Goharik
ISEC NAS RA, ASPU
Yerevan, Armenia
e-mail: petrosyangoharik72@gmail.com
petrosyan_gohar@list.ru

ABSTRACT
In the paper the simulation of a consumer process is shown using Colored Petri Nets, which was not modeled using Classical Petri Nets, since they have limited properties [1]. The author constructed a graph of Colored Petri Net that generates context-free (CF) language $L^* = L \cup LL \cup LLL...$ (in particular, $\{L = a^n b^n / n \geq 1\}$). This language may not be modeled by Classical Petri Nets [1]. Thus, the class of languages of Colored Petri Nets is supposed to include an entire class of context-free languages. This fact illustrates the restriction of Petri Nets as a tool that generates languages [1, 2]. The modified Venn Diagram is constructed. The examples given show the limited nature of the Classical Petri Nets compared to the Colored Petri Nets. The questions discussed mean that the Colored Petri Nets are an extension of the Classical Petri Nets, as they have a number of properties that can be used to model and analyze more complex systems that could not be modeled using Classical Petri Nets.

Keywords
Petri Nets, Colored Petri Nets, Traditional Languages, Transition, Position.

1. INTRODUCTION
In modern society, reliable transmission and protection of information are of current importance. One of the most important application areas is the theory of Petri Nets. Therefore, Petri Net computer system gives opportunities to study properties, use them to solve practical problems, mainly those issues that are related to information processing models, in parallel with sources, and are considered to be an important issue [1,2]. In such cases, we need to use a Petri Net. The following problems may serve as examples for problems that often occur in discrete systems in need of design and research:

• The system uses the functions, for which it was intended.
• Whether it operates effectively.
• There might occur some errors and emergency situations.
• Whether there are potential barriers and
• If it is possible to simplify the system or replace individual components with more perfect components, without prejudice to its general functions.

• If it's possible to design more complicated and more functional systems that will meet the requirements and so on.

Petri nets are tools for the study of systems. Petri Net theory allows a system to be modeled by a Petri Net, a mathematical representation of the system. Analysis of the Petri Net can then, hopefully, reveal important information about the structure and dynamic behavior of the modeled system. This information can then be used to evaluate the modeled system and suggest improvements or changes. Thus, the development of a theory of Petri Nets is based on the application of Petri Nets in the modeling and design of systems.

The birth of Petri Nets was Petri's dissertation, but most of the work in the United States is also based on the final report of the Information System Theory Project, which translated Petri's dissertation into English as well as extended the work considerably. The paper "Events and Conditions" by Holt and Commoner is also an important part of the early works. Petri presented a short paper to the 1962 IFIP Congress, which was printed in the proceedings. This paper is based on the ideas of his dissertation.

Petri Net consists of three types of components: *places* (circles), *transitions* (rectangles) and *arcs* (arrows):

- Places represent possible states of the system;
- Transitions are events or actions, which cause a change of state;
- Every arc simply connects a place with a transition or a transition with a place.

A Change of State is denoted by a movement of token(s) (black dots) from place(s) to place(s); and is caused by the firing of a transition. The firing represents an occurrence of the event or an action taken. The firing is subject to the input conditions, denoted by token availability. A transition is fire able or enabled when there are sufficient tokens in its input places. After firing, tokens will be transferred from the input places (old state) to the output places, denoting the new state. Event is an action, which occurs in the system. The appearance of the events governs the state of the system. The state of the system can be described by multiple conditions.

The condition is a predicate or a logical description of the states of the system. The condition can admit true or false value. As the events are actions, they can take place. In order for an event to occur, appropriate existence of conditions are necessary. They call these conditions the preconditions of the event. The appearance of the event can lead to the break of preconditions and bring to another fulfilment of the condition-post conditions.

Definition. 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 P -positions, T -transitions are finite sets. $I: T \to P^\infty, O: T \to P^\infty$ are the input and output functions, respectively, where P^∞ are all possible collections (repetitive elements) of P. $\mu: P \to N_0$ is the function of condition, where $N_0 = \{0,1,...\}$ is the set of integers.

Petri Nets are appropriate to Stack Automata. Therefore, sometimes the admissibility of transition is compared with the presence of bullets in a Stack Automata. Therefore, the

expression of the transition commission is mentioned in the literature as “A firing of an enabled transition”. Transition committing is presented by the existence of token positions according to that condition. Transition committing misses the allowed tokens, which present the committing of the preconditions and form new tokens, which present the committing of the post conditions.
The Petri Net is showen in Figure 1, which simulates the model of automat-seller.

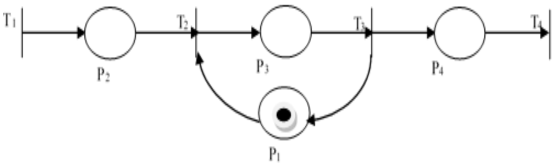

Figure 1.The Petri Net which simulates the model of automat-seller

Colored Petri Net (CPN) is a Classical Petri Net modern expansion, which was created by K. Jensen. Colored Petri Net (CP-nets or CPNs) is a modeling language developed for systems, in which communication, synchronization and resource sharing play an important role. CP-nets combine the strengths of Classical Petri Nets with the strengths of a high-level programming language (Jensen, 1992; Ullman, 1998).
The vast majority of IT systems today can be characterized as concurrent and distributed in that their operation inherently relies on communication, synchronization, and resource sharing between concurrently executing software components and applications. This development was accelerated first through the pervasive presence of the Internet as a communication infrastructure and more recently by cloud- and Web-based services, mobile applications, and multicore computing architectures [4-10]. The mathematical definition of the Colored Petri Nets can be seen in [3-7].
Colored Petri Net is a graphical oriented language for design, specification, simulation and verification of systems (Jensen, 1992; Jensen, 1996; Ullman, 1998). It is, in particular, well-suited for systems that consist of a number of processes, which communicate and synchronize. Typical examples of application areas are communication protocols, distributed systems, automated production systems, work flow analysis and VLSI chips (very large scale integration, from 10^6 to 10^7 transistors).

2. COMPARATIVE ANALYSIS OF FORMAL LANGUAGES MODELING USING CLASSICAL PETRI NETS AND COLORED PETRI NETS

2.1 The Example of Modeling Context-Free Language with Colored Petri Nets

In Petri Nets, it is impossible to remember an arbitrarily long sequence of arbitrary characters. In Petri Nets a sequence of limited length can be remembered (this is also possible in finite automata) [1,2].
However, Petri Nets do not have the "capacity of pushdown memory", which is necessary for the generation of context-free languages. The interrelation of languages of Petri Nets with other classes of languages investigated by Ven [1], is shown in Figure 2 in the form of a diagram.
We modeled $L^* = L \cup LL \cup LLL...$ language (star Kline) by CPN, in particular $\{L = a^n b^n / n \geq 1\}$.

Figure 3 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, which are used in a figure, it is necessary to give a declaration. 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 [3-9]. In Figure 3 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 (Figure 4).

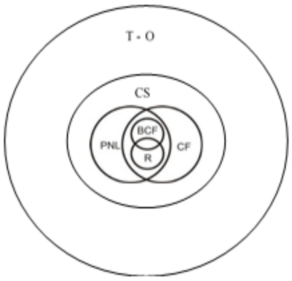

Figure 2. Interrelation of Petri Nets and Traditional Languages (T-0-the General type of languages, CS-Context-sensitive languages, PNL-Petri nets languages, BCF-bounded Context-free languages, R-Regular languages).

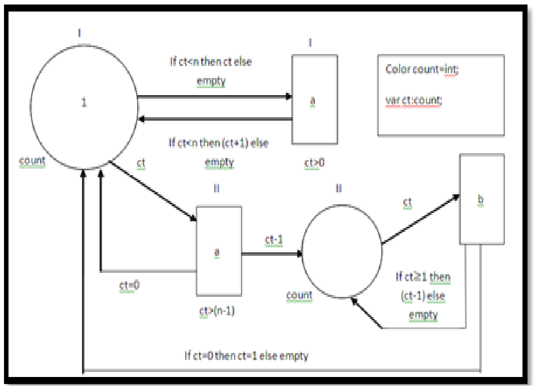

Figure 3. Modeling $L^* = L \cup LL \cup LLL...$ **language by Colored Petri Net.**

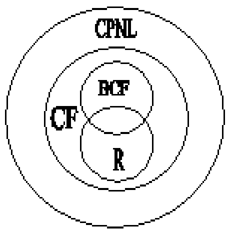

Figure 4. Interrelation of Colored Petri Nets and Traditional Languages. CPNL (language of Colored Petri Net).

2.2 The Example of Modeling Consumers Process with CPN

Let us suppose that there are two processes of producers and consumers [1, 2].
Process 1: The P_1 producer creates data for the first process of the consumer C_1, and P_2 producer does it for the consumer C_2. Data, which are produced, but aren’t

used yet, are placed in the buffer. The B_1 buffer is for (P_1, C_1) and B_2 is for (P_2, C_2) pair. Transmitting the data from buffers to consumers is done by the same channel. The channel during one séance can deliver one element (from an arbitrary buffer to an arbitrary consumer). The producers should insert the data into the buffer, and the consumers should coordinate their actions by usage of the channel. The following picture shows the above-mentioned process (Figure 5).

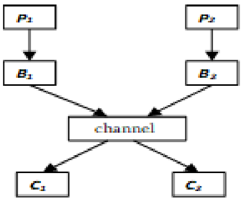

Figure 5. The Consumers process with the common usage and buffer is an action.

In the described system, there is a problem of distribution. The (P_1, C_1) pair should have priority towards (P_2, C_2) in the meaning of the channel usage. This means the following: the channel shouldn't report data from B_2 buffer to C_2 consumer as long as the B_1 buffer is not empty. The idea of priority doesn't let carry out a model of the mentioned system by Classical Petri Nets, as it is allowed in nature. The proof of the mentioned fact is described in [1, 2].

To solve that problem, it's needed to extend Petri Nets several properties in such a manner that the proposed properties are headed towards the opportunity of checking the zero in Petri Nets.

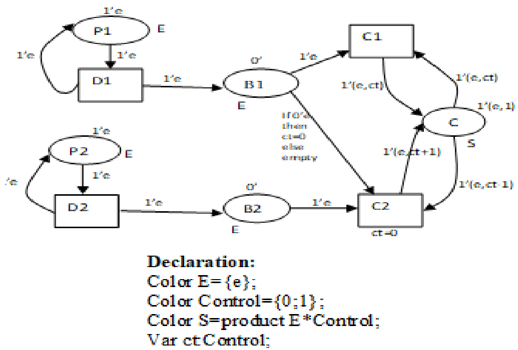

Figure 6. The modeling of Consumer problem with Colored Petri Net.

In Fig. 6, the E type declaration is used for positions, which include the produced elements. The Control type includes 0 or 1, through which the priority idea has been decided. The S type is the Cartesian product of the E and the Control types. The net (Fig. 6) is the model of the solved problem of priority usage.

3. ACKNOWLEDGEMENT

The paper deals with issues that relate to the comparative analysis of the Classical Petri Nets and the Colored Petri Nets, where, using illustrative examples, it is found that the Classical Petri Nets have limited properties of modeling with the comparison of the Colored Petri Nets. In the future, we will consider more complex systems that can be modeled with other extensions of Petri Nets.

REFERENCES

[1] J. L. Peterson, "Petri Net Theory and the Modeling of Systems," Prentice Hall, Upper Saddle River, 1981.

[2] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, Vol. 77, No. 4, 1989, pp. 541-580. doi:10.1109/5.24143

[3] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use," Springer-Verlag, Berlin, 1992. doi:10.1007/978-3-662-06289-0

[4] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Volumn 1. Basic Concepts. Monographs in Theoretical Computer Science," Springer-Verlag, Berlin, 1997.

[5] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Volumn 2. Analysis Methods Monographs in Theoretical Computer Science," Springer-Verlag, Berlin, 1997.

[6] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Volumn 3. Practical Use. Monographs in Theoretical Computer Science," Springer-Verlag, Berlin, 1997.

[7] K. Jensen, "Coloured Petri Nets: A High-level Language for System Design and Analysis," In: G. Rozenberg, Ed., *Advances in Petri Nets* 1990, *Lecture Notes in Computer Science*, Vol. 483, Springer-Verlag, Berlin, 1991, pp. 342-416.

[8] K. Jensen, "Coloured Petri Nets: A High-level Language for System Design and Analysis," In: K. Jensen and G. Rozenberg, Eds., *High-Level Petri Nets. Theory and Application*, Springer-Verlag, Berlin, 1991, pp. 44-122.

[9] J. D. Ullman, "Elements of ML Programming," Prentice- Hall, Upper Saddle River, 1998.

[10] M. Bevilacquaa, F. E. Ciarapicaa, M. Giovannia, "Timed Coloured Petri Nets for modelling and managing processes and projects", *11th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '17, Procedia CIRP*, pp. 58 – 62. v. 67, 2018.