

# Constrainedness Measurement of Petri Net Models

DOGUS BEBEK and HUREVREN KILIC

**Abstract**—It is shown that Shannon’s information capacity calculation metric (originally developed for discrete noiseless channels) can be used for constrainedness quantification of Petri Net based models if they are k-bounded and deadlock free. The proposed approach includes generation of finite marking reachability graph from a given restricted Petri Net based model followed by capacity calculation using the graph by setting up an analogy between Petri Net transitions and letters of Shannon languages.

**Keywords**— constrainedness quantification, information capacity calculation, Shannon language, Petri Nets

## I. Introduction

Petri Nets are mathematical tools for modeling, formal analysis and design of engineering systems. They are being used for years to realize such systems especially showing parallel and concurrent processing requirements [1]-[3]. Petri Net applications range from communication protocols [4], distributed software systems [5], service composition [6][7], peer-to-peer systems [8], security [9] and data flow oriented computational systems [10] to software engineering [11][12], formal languages, compilers, operating systems and local area networks etc.

For any engineering system modeled using Petri Nets, we can position the places and transitions as primitives describing constrainedness of system under development. The set of markings on the other hand, reveals the constraints of the Petri Net based view to the system [13][14]. Therefore, the marking reachability graph of the system defines a form of constrained based view to the system. Information processing capacity calculation based on combinatorial execution of transitions realized over marking reachability graph defines a quantified value about system’s capability.

As a consequence, system designer may enhance his/her design (before its implementation) based on the computed and a target constrainedness values. Shannon’s combinatorial capacity calculation metric [15] can be used to realize such quantification. In [16], it has been shown that the metric can suitably be used for component based systems’ combinatorial capacity calculation.

In this paper, we suggest an approach to apply Shannon’s information capacity calculation metric to the Petri Net based models. Using our proposed procedure, Petri Net designer can figure out the degree of freedom (or constrainedness) of his/her model. Lower values of the metric imply smaller degree of freedom or highly constrained system. In order to obtain more (or less) constrained system, the designer can apply necessary compression (or relaxation) on his/her model by looking at the computed value. Note that the original combinatorial capacity calculation proposed by Shannon uses finite state automata model to represent discrete noiseless channels. This leads us to make k-bounded and deadlock free Petri Net model assumption for our purpose. The approach developed in this work has also been applied in Multiagent Systems context [17].

Sections 2 and 3 include background information about Petri Nets and Shannon’s information capacity calculation metric, respectively. In Section 4, our proposed measurement algorithm is introduced and explained by an example. Section 5 is the conclusion.

## II. Petri Nets

A Petri Net is a particular bipartite digraph with three types of objects namely places, transitions, and directed arcs. Formally, one can define them as below [18].

**Definition 1:** A Petri Net PN is a five-tuple  $(P, T, I, O, M_0)$  where

- $P = \{p_1, p_2, \dots, p_m\}$  is a finite set of places,
- $T = \{t_1, t_2, \dots, t_n\}$  is a finite set of transitions  $P \cup T \neq \emptyset$  and  $P \cap T = \emptyset$ ,
- $I : (P \times T) \mapsto \mathbb{N}$  is an input function that defines directed arcs from places to transitions where  $\mathbb{N}$  is a set of nonnegative integers,
- $O : (P \times T) \mapsto \mathbb{N}$  is an output function which defines directed arcs from transitions to places, and
- $M_0 : P \rightarrow \mathbb{N}$  is initial marking.

---

Dogus Bebek, MSc.  
Business Intelligence and Datawarehouse Systems Department, Akbank  
T.A.S. Istanbul, TURKEY

Hurevren Kilic, PhD  
Computer Engineering Department, Engineering and Architecture Faculty,  
Gediz University, Menemen, Izmir, TURKEY

Places represent conditions and transitions represent events. Places are differentiated as input and output places. An input place can be defined as a place which has a directed arc pointed towards a transition. Similarly, an output place can be defined as a place which has an incoming directed arc from a transition. A transition has finite number of input and output places which represent pre-conditions and post-conditions of an event, respectively [19]. A token presented in a place indicates that truth condition of that place is met. In other words, k-tokens in a place mean that k data items or k resources are available at that place. At any given time instance, distribution of tokens on places represented by a vector of m places is called  $i^{\text{th}}$  Petri Net marking  $M_i$ . If a marking assigns to place p with positive integer k, it is said that p has k tokens. A marking  $M_i$  is reachable from  $M_0$  if there exist a sequence of firings that transform  $M_0$  to  $M_i$ . The set of all possible markings reachable from  $M_0$  is called reachability set and denoted by  $R(M_0)$ .

Note that finiteness of the reachability graph and applicability of transition sequences through place visits without any deadlock occurrence are two basic requirements for our purpose. We define these restrictions resulted in expected behavioral properties of Petri Nets under consideration as below [18]:

- **Bounded Petri Net:** A Petri net is said to be k-bounded if the number of tokens in any place p, is always less than or equal to k (where  $k \geq 0$ ) for every marking M reachable from  $M_0$ .
- **Live Petri Net:** A Petri Net is said to be live if, no matter what marking has been reached from  $M_0$ , it is possible to ultimately fire any transition of the net by processing through some further firing sequence. This means that, Petri Net guarantees deadlock free operation, no matter what firing sequence is chosen. Note that for our purpose, L4-liveness is a requirement since we need for any marking reachable from marking  $M_0$ , it should be possible to ultimately fire any transition by executing some firing sequence.

### III. Combinatorial Capacity Calculation

The following definitions from 2 to 11 and the theorem are compiled from [20]. For the original proof of Theorem 1 and background information see [15].

**Definition 2:** A *discrete noiseless channel* is a channel which allows the noiseless transmission of a sequence of symbols chosen from a finite alphabet A (called q-letter alphabet), each symbol, say  $a \in A$ , having certain duration  $\tau(a)$  in time, possibly different for different symbols.

**Definition 3:** A *word of length k* over A is a finite string of k letters from A. If  $\alpha = a_1a_2\dots a_k$  is a such word, its duration is defined to be

$$\tau(\alpha) = \tau(a_1) + \tau(a_2) + \dots + \tau(a_k)$$

**Definition 4:** A *language L* over A is a collection of words over A. The discrete noiseless channel associated with L, the *L-channel* for short, is the channel which is only allowed to transmit sequences from L, where it transmits them without error.

**Definition 5:** *Shannon language* is defined by a directed graph whose edges are labeled with letters from the alphabet A. The corresponding language L is then defined to be the set of words that result by reading off the edge labels on paths of the graph.

**Definition 6:** Let L be a Shannon Language, the *combinatorial capacity of the L-Channel* is defined as

$$C_{comb} = \limsup_{t \rightarrow \infty} \frac{1}{t} \log(N(t))$$

where  $N(t)$  is the total number of words in L of duration t.

In Shannon's original work [15], an algebraic method of computing  $C_{comb}$  has been given. In [20], on the other hand, a more detailed and simpler (at least for the authors of this paper) calculation of  $C_{comb}$  has also been elaborated. Therefore, we prefer to continue with definitions from [20] which are finalized by Theorem 1 defining the  $C_{comb}$  calculation.

**Definition 7:** A *directed graph* (or digraph) G has vertex set  $V = \{v_1, \dots, v_M\}$  and branch set  $B = \{b_1, \dots, b_N\}$ . Each branch  $b \in B$  has an *initial vertex*  $\text{init}(b) \in V$  and a *final vertex*  $\text{fin}(b) \in V$ . The set of branches with initial vertex v and final vertex w is denoted by:

$$B_{v,w} = \{b \in B : \text{init}(b) = v, \text{fin}(b) = w\}$$

**Definition 8:** A *path P* of length k from v to w in G is a sequence of k branches  $P = b_1b_2\dots b_k$  with  $\text{init}(b_1) = v$ ,  $\text{fin}(b_1) = \text{init}(b_2)$ ,  $\dots$ ,  $\text{fin}(b_{k-1}) = \text{init}(b_k)$ ,  $\text{fin}(b_k) = w$ . We write  $\text{len}(P) = k$ ,  $\text{init}(P) = v$ , and  $\text{fin}(P) = w$ . Then, the set of paths of length k from v to w is denoted by:

$$B_{v,w}^k = \{P : \text{len}(P) = k, \text{init}(P) = v, \text{fin}(P) = w\}$$

Using a directed graph G and an alphabet A, we can generate a language by labeling each branch of G with an element of A.

**Definition 9:** Let  $\lambda : B \rightarrow A$  be a labeling.  $\lambda$  is *right-resolving* iff for each vertex v, the labels on all branches with  $\text{init}(b) = v$  are distinct.

**Definition 10:** The set of all possible path labels is called the *Shannon language* generated by G, and denoted by  $L_{G,\lambda}$ .

**Definition 11:** Let s be nonnegative real number, and for a given pair of vertices (v, w), *branch duration partition* function is defined as:

$$P_{v,w}(s) = \sum_{b \in B_{v,w}} e^{-s\tau(b)}$$

The functions  $P_{v,w}(s)$  can be thought of as entries in an  $M \times M$  matrix P(s). The *spectral radius* (the magnitude of the largest

eigenvalue) of the matrix  $P(s)$  is represented by  $\rho(s)$  and it is also called the *partition function* for the language  $L_{G,\lambda}$ .

**Theorem 1:** The *combinatorial capacity* of the  $L_{G,\lambda}$  language is given by

$$C_{comb} = \ln(s_0)$$

where  $s_0$  is the unique solution to the equation  $\rho(s) = 1$ .

Alternatively,  $C_{comb}$  is the greatest positive solution of the equation,  $q(s) = \det(I - P(s)) = 0$ . In our computations, we used this alternative definition.

#### IV. Proposed Algorithm

Originally, the Shannon's metric calculates maximum amount of information that can be transmitted over a discrete noiseless channel per unit cost. If the cost (due to symbol transmissions) is measured in unit of time (being additive), say in seconds, the calculation results in bits per second. For our purpose, we establish an analogy between description of discrete noiseless channel and marking reachability graph of k-bounded, L4-live Petri Net. Each different Petri Net transition appearing in the Net's reachability graph represents different symbols of Shannon's alphabet whose associated costs are interpreted as the contribution of the transitions  $t_i$  to the degree of being constraint  $d_i$  over the model under evaluation. Note that, in our calculation, we do not work on the model directly but on its marking reachability graph since each marking represent a global state of the system that changes during event transitions having different contributions to the degree of system's constrainedness. As a consequence, lower channel capacity is supposed to imply more constrained system or vice versa. Below is our proposed algorithm for the purpose.

**ALGORITHM:**

**Inputs:** A k-bounded, L4-live Petri Net model P; A vector of degree of being constraint  $d_i$  for transition  $t_i$ ;

**Output:** Degree of constrainedness F of the model (i.e.  $C_{comb}$ );

1. Obtain "marking reachability graph" G by using input P.
2. Label the edges of G as Shannon alphabet letters by using given transitions  $t_i$  and corresponding  $d_i$  values.
3. Calculate combinatorial capacity  $C_{comb}$  of the constructed Shannon language  $L_{G,\lambda}$  using Theorem 1. Set F to  $C_{comb}$ .

One can intuitively conclude that a lower bound for F is nothing but zero. Also, we need to figure out an upper bound for F. The upper bound is attained when the marking reachability graph defines a free Shannon language. In other words, the system is not constrained due to different execution sequences of transitions. The free Shannon language is simply the set of all possible strings (i.e. paths) of all lengths over its q-letter alphabet A which is constituted by transitions, in our

case. The combinatorial capacity of a standard q-ary alphabet with a fixed unit cost say c is defined by the unique solution to equation  $P(s) = qe^{-cs} = 1$  which is  $s = \ln(q) / c$ . Detailed description of the equation can be found in [20]. Below, we give an example for the application of the algorithm.

**Example:** Given a k-bounded and L4-live Petri Net model in Figure 1-(a), we first generate corresponding marking reachability graph shown in Figure 1-(b).

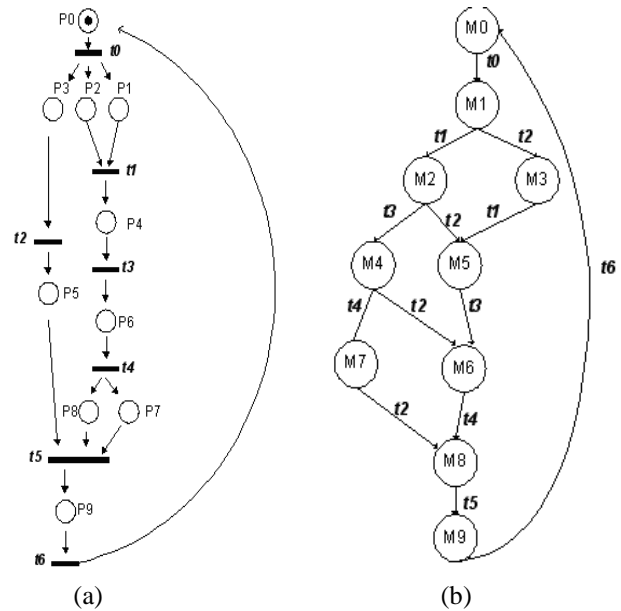


Figure 1. Example Petri Net model (a) and its Marking Reachability graph (b).

Then, we complete Shannon language description by using example  $d_i$  values say 1, 2, 5, 12, 8, 4, 1 for  $t_0$  to  $t_6$  showing individual contributions of each transition to the degree of being constrained.

Finally, we calculate the degree of constrainedness of the model under consideration. For this purpose, we construct the matrix  $P(s)$  depicted in Figure 2 by using the obtained Shannon language (see Definition 11).

$$P(s) = \begin{bmatrix} 0 & e^{-s} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{-2s} & e^{-5s} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-12s} & e^{-8s} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-4s} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-1s} & e^{-2s} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-5s} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-12s} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-8s} \\ e^{-1s} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2. Constructed  $P(s)$  matrix for the example Petri Net model.

The greatest positive solution of equation  $\det(I - P(s)) = 0$  gives constrainedness degree 0.0915 bit per unit constraint for the example model.

## v. Conclusions

In this work, we developed an approach to calculate degree of constrainedness of Petri Net based models. For our purpose, we established an analogy between Shannon's languages and marking reachability graphs of Petri Nets. A restriction of the approach is its applicability to only k-bounded and L4-live models. Using our approach, net designer can figure out an upper-bound for run-time degree of constrainedness of his/her model. Accordingly, he/she can apply relaxation (i.e. increase degree of freedom) or compression on the model under study until a target value is attained.

A drawback of the developed procedure is calculation inefficiency that appears especially when the number of marking states is high. In future, an attack for this problem can be to investigate the applicability of more efficient approaches including parallelization of the algorithm. Furthermore, development of an approach for constrainedness calculation for unbounded Petri Nets can be interesting.

### Acknowledgment

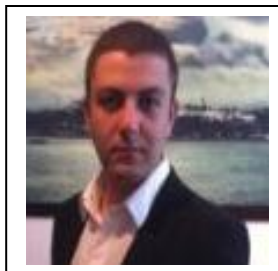
This paper is by-product of Dogus Bebek's M.Sc. Thesis done at Atilim University, Ankara, Turkey. Author Hurevren Kilic would like to thank to Dr. Md. Haidar Sharif from Gediz University Computer Engineering Department for his valuable comments and discussions about Petri Net and Information Theory related parts of this work.

### References

- [1] Girault, C. and Valk, R. Petri nets for systems engineering. Springer-Verlag, 2003.
- [2] Brauer, W. and Gold, R. "Concurrent processes and Petri nets". In: Schwichtenberg, H. (Ed.), Proof and Computation, Springer, 1994, pp. 1-64.
- [3] Gold, R., "Petri Nets in Software Engineering". Working Paper, [https://www.thi.de/fileadmin/daten/allgemein/dokumente/Working\\_Paper/ABWP\\_05.pdf](https://www.thi.de/fileadmin/daten/allgemein/dokumente/Working_Paper/ABWP_05.pdf), as of August, 2014.
- [4] Berthelot, G. and Terrat, R., "Petri Nets theory for the correctness of protocols". IEEE Trans. Communications, Vol. 30, Issue 12, December 1982, pp. 2497-2505.
- [5] Berthomieu, B., Decitre, P. and Azéma, P., "The design and validation by Petri nets of a mechanism for the invocation of remote servers". Proc. of the International Federation for Information Processing Congress (IFIP), Amsterdam: North Holland, 1980, pp. 599-604.
- [6] Fan, G., Yu H., Chen, L. and Liu, D. "Petri net based techniques for constructing reliable service composition", Journal of Systems and Software, Vol. 86, Issue 4, April 2013, pp. 1089-1106.
- [7] Xu, Y. and Huang H., "A Petri Net-Based Model for Aspect-Oriented Web Service Composition", Proceedings of International Conference on Management and Service Science, (MASS), 2009, pp. 1-4.
- [8] Oztoprak, K. and Kilic, H., "Protocol and Connectivity Based Overlay Level Capacity Calculation of P2P Networks", Proceedings of IEEE/WIC/ACM International Workshop on P2P Computing and Autonomous Agents (IEEE/WIC/ACM P2PAA 2006), Hong Kong / China, Dec. 2006, pp. 447-450.
- [9] Yang N., Yu H., Qian Z., Sun H., "Modeling and quantitatively predicting software security based on stochastic Petri nets", Mathematical and Computer Modelling, Vol. 55, Issues 1-2, January 2012, pp 102-112.

- [10] Kluge, W.E., "Reduction, data flow and control flow models of computation". Proceedings of an Advanced Course, Bad Honnef, Lecture Notes in Computer Science vol.255, Springer-Verlag, 1987, pp. 466-498.
- [11] Denaro, G. and Pezze, M. "Petri Nets and Software Engineering", Lecture Notes in Computer Science (LNCS) Vol. 3098, 2004, pp 439-466.
- [12] Hu W., Yang, X., Zuo, K. "Based Aspect-oriented Petri Nets in Software Engineering", Physics Procedia, Vol. 22, 2011, pp. 646-650.
- [13] de Assis, F.M. and Barros, T.C., "Evaluation of Petri Nets Compressibility". Proceedings of International Symposium on Information Theory (ISIT), pp. 323, Ulm, Germany, 1997, pp. 323.
- [14] Gurjao, E.C., de Assis, F.M., Perkusich, A. and Pimentel, C., "Petri Nets Compressibility and Capacity of Concurrent Systems". Proceedings of International Symposium on Information Theory (ISIT), Yokohama, Japan, 2003, pp. 250.
- [15] Shannon, C.E., "A mathematical theory of communication". Bell System Tech. J., Vol. 27, pp. 379-423, July 1948, pp. 379-423.
- [16] Seker, R. and Tanik, M. M., "An Information-Theoretical Framework for Modeling Component-Based Systems". IEEE Trans. on Systems, Man and Cybernetics- Part C: Applications and Reviews, Vol. 34, Issue 4, November 2004, pp. 475-484.
- [17] Bebek, D., "An Information Theory Based Capacity Calculation Metric For Multi-Agent and Petri Net Modeled Systems", M.Sc. Thesis, Nov. 2009, Atilim University, Ankara, Turkey.
- [18] Zurawski, R. and Zhou, M.C., "Petri Nets and Industrial Applications: A Tutorial". IEEE Transactions on Industrial Electronics, Vol. 41, No.6, December 1994, pp. 567-583.
- [19] Murata, T. "Petri Nets: Properties, Analysis and Applications". Proceedings of the IEEE, Vol. 77, No. 4, April 1989, pp. 541-580.
- [20] Khandekar, A. McEliece, R. and Rodemich, E., "The discrete noiseless channel revisited". In Coding, Communications, and broadcasting, P. Farrel, M. Darnell, and B. Honary, Eds. Baldock, U.K.: Research Studies Series Ltd., 2000, pp. 115-137.

About Authors:



Dogus Bebek received his B.Sc. degree in Computer Engineering and M.Sc. degree in Software Engineering both from Atilim University, Ankara, Turkey in 2005 and 2009, respectively. His main expertise areas are Decision Support Systems and Business Intelligence architectures and consultancy.



Hurevren Kilic is an Associate Professor of Computer Engineering at Gediz University, Izmir, Turkey. He received his B.Sc., M.Sc. and PhD degrees all in Computer Engineering from Middle East Technical University, Ankara, Turkey, in 1989, 1992 and 1998, respectively. He is an author/co-author of 50-around journal and/or conference publications. His current research interests include Automata Theory and its applications to Software Modelling, Metrics and Search-Based Software Engineering field.