Modeling using Petri grids

Muhammad Abdul Hafiz Shehab
haneen.d.hameed@aliraqia.edu.iq
Shaheen Ibrahim Kamil
ishaheen712@gmail.com
Bushra Abbas Mohammed
bushraaabbas7@gmail.com
Al-Iraqia University/ College of Education for Women

ABSTRACT

Modern systems result in a number of challenges that can be considered and studied by choosing the appropriate modeling techniques, some of these techniques aim to represent behavior within what is known as process modeling, while other techniques aim to analyze and study performance quantitatively and qualitatively, which is called performance modeling. The importance of the latter is due to the increasing complexity of current systems, the difficulty of tracking performance at the source code level, and the inability to predict virtually all possible situations.

Petri networks are the most famous performance modeling techniques, which have graphical capabilities to represent the characteristics of systems as well as a mathematical representation of them. Since their innovation to the present day, these networks have shown their great capabilities in modeling, analyzing, scheduling and controlling different types of information systems, transport and communication systems, which are characterized by characteristics that possess synchronization, competition, distribution and randomness. Petri network types have varied since their inception to meet the successive evolution of systems complexity. It's often important to provide the ability to dynamically convert between models, helping to rid analysts of the need to have enough expertise for each type of model and tool. In this research, due to the importance of Petri networks, we have worked on finding solutions to problems related to their structural analyses by means of transformations to generalized random Petri networks, and we have done an opposite conversion process to take advantage of simulation capability.

Key words: Tracking performance, petti networks, transformations, modeling.

النمذجة باستخدام شبكات بيتري

محمد عبد الحفيظ شهاب

شاهين ابراهيم كامل

بشرى عباس محمد

الملخص

تترتب على النظم الحديثة عدد من التحديات التي يمكن النظر فيها ودراستها من خلال اختيار تقنيات النمذجة المناسبة، حيث تهدف بعض هذه التقنيات إلى تمثيل السلوك ضمن ما يعرف بنمذجة العمليات، بينما تهدف تقنيات أخرى إلى تحليل ودراسة الأداء كميا ونوعيا، وهو ما يسمى نمذجة الأداء. ترجع أهمية الأخير إلى التعقيد المتزايد للأنظمة الحالية، وصعوبة تتبع الأداء على مستوى الكود المصدري، وعدم القدرة على التنبؤ بجميع المواقف المحتملة تقريبًا.

تعتبر شبكات بيتري من أشهر تقنيات نمذجة الأداء، حيث تتمتع بقدرات رسومية لتمثيل خصائص الأنظمة بالإضافة إلى التمثيل الرياضي لها. وقد أظهرت هذه الشبكات منذ ابتكارها إلى يومنا هذا قدرات كبيرة في نمذجة وتحليل وجدولة والتحكم بأنواع مختلفة من نظم المعلومات وأنظمة النقل والاتصالات، والتي تتميز بخصائص تمتلك التزامن والمنافسة والتوزيع والعشوائية. تنوعت أنواع شبكات بيتري منذ بدايتها لتواكب التطور المتتالي لتعقيد الأنظمة. غالبًا ما يكون من المهم توفير القدرة على التحويل ديناميكيًا بين النماذج، مما يساعد على تخليص المحللين من الحاجة إلى الخبرة الكافية لكل نوع من النماذج والأدوات. في هذا البحث، ونظراً لأهمية شبكات بيتري، عملنا على إيجاد حلول للمشاكل المتعلقة بتحليلاتها الهيكلية عن طريق التحويلات إلى شبكات بيتري العشوائية المعممة، وقمنا بعملية تحويل معاكسة للاستفادة من إمكانية المحاكاة.

الكلمات المفتاحية: تتبع الأداء، الشبكات الصغيرة، التحولات، النمذجة

Chapter One

Boot

1.1 Introduction:

Petri networks are an effective means of modeling different types of systems and in multiple fields in order to verify their characteristics, study their performance and ensure the correctness of the desired results. These networks have been evolving since they were invented in 1961 in the doctoral thesis of Carl Adam Petrie. Among the most prominent of these networks, we find deterministic and random Petri networks used in the study of systems with time parameters such as packet transfer time in distributed systems, lead times in real−time systems and repair time in systemsForgive errors, colored Petri grids In order to increase the expressive ability to represent Petri networks must have the ability to interoperability among themselves, in the sense that researchers and analysts are able to share models so that they can use different tools in analyzing the same model. Hence the importance of the Petri Net Markup Language (PNML) standard, which works to share models by configuring them in the form of XML Stylesheet files. This characterization allows the use of Extensible Markup Language (XML) technology and Petri networks with ardeners that contribute to the study of stubborn and programmatic characteristics together in one model. In the process of conversion between them Language Transformation (XSLT)

It is often necessary to secure the ability to exchange models that are built using Petri networks between tools that may be distributed in different geographical areas. This allows users to take advantage of advanced capabilities in other means such as analysis, simulation and verification. This necessity arises depending on the different capabilities of the tools, so someone may want to build the model in one tool, verify it in a second tool, and simulate it in a third. Because there are no common interfaces and methods for conversion, tools can't communicate with each other or even exchange forms. The distributed modeling methodology for a collaborative environment requires a lot of coordination, as the problems of the security domain in using open networks and trust raise the question of which coordination mechanism to follow.

In the past twenty years, tools began to appear that support different types of Petri networks, and became available for use and development, and with the multiplicity of patterns and methods of analysis, it becomes difficult to have a single tool that supports all the analysis techniques used in the study of structural and non-structural properties of computer systems. These tools dealing with Petri networks vary in terms of their expressive ability to represent additional structures other than traditional structures in Petri networks, which are represented in parentheses, places and transitions.Transitions and Tokens allowing the representation of different states in a simplified manner, and the most prominent of these structures are the retarding and zeroing arcs that are not available in the tool.

Queuing Petri Net (QPN), invented in 1993 by Falco Bass, is an effective modeling method for studying distributed systems and analyzing their performance and scalability, by integrating the expressive capability of both Petri and frame networks. Based on this integration, QPNS networks) Dealing with standard and programmatic considerations in a single model, as well as the scheduling policies that support it and representing competition for resources, also enables us to model synchronization, asynchronous processing and competition at the software level. These considerations call into question the performance of modern software systems.

The recent development in terms of technologies and information systems requires that operations must be faster than ever, for example, the Internet of Things technology has made the performance of suppliers more effective by carrying out tasks very quickly, and modern means of transportation such as drones need great speed in carrying out operations. Many process modeling techniques have been developed with the aim of descripting, analyzing and evaluating them, the most common of which are: Business Process Model and Notation (BPMN) which can describe the structure of complex processes, the organizations that implement these operations, and the way resources are managed. Descriptive elements can be used to represent many concepts, but this will make the form of models more complex and difficult to verify them, so we are looking for analytical tools that can be used with BPMN (BPMN). These tools should be equipped with capabilities to simulate and evaluate time considerations in BPMN models.

1-2 The importance of research:

Petri networks are a powerful tool for modeling complex systems. Used to describe and analyze the control flow and information flow in the system, especially those systems with asynchronous and concurrent activities Used to represent positions (places) and transitions and the circle represents the position (place), and the token in the circle indicates that the condition is met. The bar represents the transition. The Petri grid graph is a two-part vector graph used to illustrate many Petri network concepts.

1-3 Research problem:

It is the process of developing a model abstract or graphical representation using realistic concepts or ideas. During petrometric modeling, different assumptions are made regarding how the system works. The models also illustrate the prevailing processes in the system and how they are related. These processes may include factors known to drive change in the system, or they may include the consequences of change in the factors themselves.

1.4Research Methodology

The descriptive approach was extracted in writing scientific research

1.5 Data collection tools

1- International information networks (Internet)

2. Some libraries

3- Some books

Chapter Two

Modeling using Petri grids

2.1 Introduction to Petri net modeling

Petri Net Modeling is a powerful technology used to model and analyze complex systems. It provides a graphical representation of the behavior of the system, allowing us to study its dynamics, identify potential problems and improve its performance. This section will provide an introduction to Petri Net modeling, exploring its basic concepts and how they can be applied to different domains [1]

At its core, Petri Net Modeling consists of two basic elements: places and transitions. Places represent system states, while transformations represent events or actions that cause a change in system state. These elements are linked by parentheses that define the flow of tokens (representing resources or information) between places and transitions.

2.2 Modeling examples

To better understand the Petri Net model. Such a simple manufacturing process includes three stages: production, quality control and packaging. We can represent this system using Petri networks, where places represent the different stages and transitions that represent actions between them. By analyzing the PetriNet model, we can identify bottlenecks, improve the process and improve overall efficiency

Another example could be a traffic intersection. Places can represent different lanes, while transitions represent variable traffic signs. By modeling the intersection using Petri Nets, we can simulate and analyze traffic flow, identify potential congestion points, and improve signal timing to reduce traffic jams [2]

2.3Petri network components

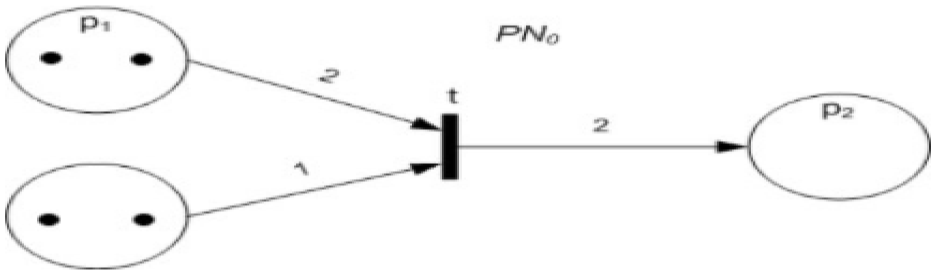
Petri's network is based mainly on a group of elements that are either

1- Places represent the states or conditions of the system. It can contain a certain number of symbols that represent resources, data, or entities.

2- Transitions represent events or actions that can occur in the system. Represents changes in situations or interactions between places.

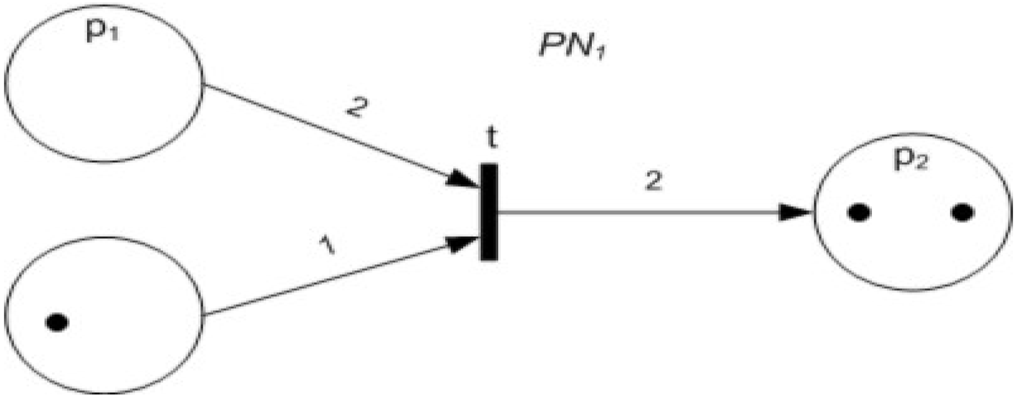
3- Symbols: represent the presence or absence of resources, data or entities in places

Example of a simple Petri grid



This simple example shows us a grid of 3 places and one transition let's say it represents a pulling machine so that for two units of place p1 and one unit of place p3 will give you two units of a p2 type product.

There are a large number of types of networks that you can adopt colored and object-oriented and and and a large number of methods of installing networks as needed



2-4Performance modeling:

We use performance modeling to predict the behavior of systems during different stages of their life, and it can also be used to plan for capacity and understand many changes, such as changing scheduling rules or predicting the impact of deleting or adding loads. There are many methodologies that make building performance models an automated process, which must be quick and accessible to designers among these methodologies. [3]

1- Extracting forms from registration paths within what is known as process mining

2- Conversion between models.

3- Feedback in software design

2.5Concepts of the process of conversion between models

We will introduce a number of concepts that are needed to understand conversion processes when talking about them, including:

1- Model: It is a simplified (abstract) representation of the actual system. Own models using domain-specific languages or general-purpose languages such asUnified Modelling Language (UML). They often have a pictorial form [4]

2- Meta-Model: The descriptive model of Model X has the structure for which this model is valid. The descriptive model must be precisely defined as a grammar.

3- Meta-meta Model: It is a descriptive model that describes the descriptive model of a model. There are criteria and evidence for the representation of descriptive models, for example (Meta Object Facility (MOF or Ecore), which is the most common investigation and is synthesized using the modeling environment in Eclipse. Eclipse Modeling Framework (EMF)

4- Model Transformation Description: This description expresses how to convert one or more source models into one or more target models by writing them using the conversion language, if the language depends on rules, then we say that the description is a set of conversion rules that can also be used instead of the word description of terms such as program, code or definition

5- Transformation Rule: It is a description that determines how to convert part of the source model into part of the target model. The rule contains the source form and the target form, and every occurrence of the source form is converted to the target form.

6- Transformation Engine Tool to implement the rules or description that was written using the conversion language, we need a tool or engine to implement these rules The property of the ATL that we will rely on in this thesis.

2.6 Definition of Petri networks

Petri grids are mathematically represented according to the trilogy (N = (P), TF, where it consists of the following components: Positions: It is one of two basic components in human networks, and is graphically expressed in an oval or circular shape, and is described as negative components, meaning that the position stores, assembles or shows things. [5]

- T transitions: They are the second component in Petri grids and have a rectangle or square. Transition is the active component as it works to produce, consume, move or change objects.

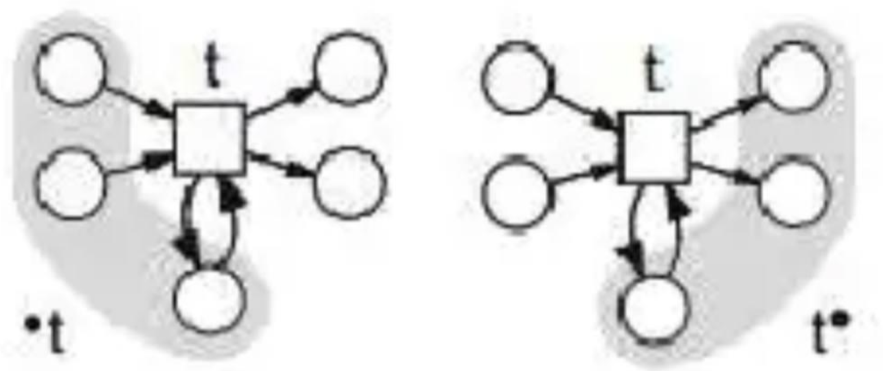
– Brackets F: Positions are connected to transitions by directed arcs, and the arc does not have any of the components of the systems, and the arcs cannot connect two positions or two transitions together. Parentheses are described according to the following relationship:

$$F \subset (PxT) \cup (TxP)$$

The rear groups are known as x• and front.x of element x according to the two relationships:

$$\cdot x = \{y|yFx\} \qquad x = \{y|xFy\}$$

The two elements, in the N grid, form a ring if y€x & x€y as in Figure (1)



We call the distribution of symbols in positions during a particular crossing Marking, so that we have an initial determination before crossing any transition. We say about transitions in Petri grids that they are activated if they are transitable – when the number of symbols in the input positions of the transition is not less than the weight on the input brackets for it – and once they are crossed the symbols are deleted from the input positions and it is enough to do a task, and the symbols are placed according to the weights on the brackets in their output in the output positions perform these The process leads to the selection change in the input and output positions of the transition and therefore the state of the network as a whole. It should be noted that the transit of the transition in normal Petri networks is an atomic process, in the sense that the transit process occurs in one step that is not interrupted, in addition to that, the implementation of Petri networks is not inevitable, meaning that several transitions can be activated and crossed simultaneously and in a different order[6] PxT→N ,I^{out} ,I^{in} Consequences determine how positions communicate with transitions, ifP:t)>0)

Then there is an arc from position p to the transition, and the number of symbols of P:t is omitted)) When crossing the transition .. if P:t)>0) We say that there is a connection from the transition to the position, and when crossing it a number of symbols equal to P:t)) In position p. It can be said that the consequences of occurrence place integer values on the parentheses that connect the transitions and positions, which are referred to as parentheses weights. When each position in the transition input has at least as many symbols as the weight on the arc it connects to the transition, then we say that the transition is activated. Crossing the transition) changes the selection in the grid from M to 'M in the following cases:

-M'(P)=M(P)+ $(p<t_o)$

-M'(P)=M(P)+ $(p<t_o)$

2.7 Benefits and applications of Petri networks

Petri networks are suitable for information systems that are asynchronous, competing, distributed, non-deterministic and that may also be random. The wide spread of Petri networks in various fields is due to the following[8]

1- Petri networks easily illustrate the relationships of precedence and interaction between asynchronous and competing activities as well.

2- You can simulate and verify models in addition to the possibility of including time constraints and their ability to show the dynamic behavior of systems through the Token Game.

3. It is an appropriate tool for systems that include communication, synchronization and sharing of resources.

4- Petri networks have a mathematical definition that helps verify important properties in systems such as stability analysis.

2.8Expansions of Petri networks:

The expansions of Petri's networks from their inception to the present time can be viewed according to several considerations, including:

- The way behavior is represented.
- Elements that support it.
- Integration with other technologies.
- Analysis methods.

The evolution of behavior representation using Petri networks is as follows:

1- The flow of basic control of systems and simple behavior in them.

2 Time-dependent behavior.

3. Behavior that involves the presence of certain data and functions.

4- Petri networks in a pyramidal shape.

5- Object-oriented modeling.

6- Agent-based modeling.

7 - Modeling based on software structure

The expansion of Petri networks dates back to the seventies of the last century, where these expansions mostly cause a variety of transition rules and lead to an increase in expressive ability. For example, the amplitude in one place, which expresses the maximum number of symbols in that position can be simulated with the presence of another complementary position, also testing the absence of any symbols in one position facilitates the generation of formal languages, as well as temporal constraints help to increase Expressive Ability

Petri networks initially appeared as causal models without any time inclusion or even probability. Adding time to many applications is important. Without declaring tenses, important performance parameters such as productivity and utilization cannot be measured. In the early seventies, talk began about adding time in Petri networks, and accordingly many methodologies appeared in its inclusion and many special methods of analysis Time can be added to symbols, transitions or positions The most commonly used method is to put delays Temporal on transitions because they have activities that can take time to implement Some consider that inclusion should be on positions where we preserve the original meaning of transit transitions, i.e. activation and transit are done together in many high-level Petri networks. Tenses can be placed with symbols, which indicate the time when the code becomes ready for consumption [9].

2.9 Petri network types:

Many patterns of Petri networks have emerged as a result of their continuous expansions, which are accompanied by specific tools and analysis methods. We will give a glimpse of the most common patterns that have been widely used in recent years in different fields. These patterns include: [10]

1- Timed Petri Nets (TPNS).

2- Stochastic Petri Nets (SP).

3- Benzi random generalized networks (GSPNs)

4- Colored Petri Nets (CPN)

5- Petri networks with QPN

Chapter Three

use the concept of Petri network to organize resource sharing between processors

3.1 Petri Network

It is a graphical interface and a mathematical modeling tool, and it consists of places and transitions , which are linked to each other by arcs, places contain tokens and represent the current state of the system, and represent symbols in the form of numbers. [11] The activation process (Firing) is done by the transition of the code from the initial state of the network and is called (Initial(0)Marking) to the outlines directly of the Petri network and is weighted There are several concepts in addition to places and transitions such as the concept of the independent Petri network and non-independent[12].

Places are represented by circles, transitions are represented by squares or boxes connected to each other by parentheses, places express conditions and transitions are the event), Table 1 shows some of the ideal change processes for the transition operation.

الجدول(1) عمليات التغيير[4]

Table (1) Change processes (4)

In[illegible]		
Precondition	Event	Post Condition
Input Data	Computation Step	Output Data
Input Signal	Signals Process	Output Signals
Resources Needed	Task or Job	Resources Released
Buffers	Processor	Buffers

٣.٢Basic concepts in Petri networks: There are some concepts that are frequently discussed in the field of Petri networks, including the following:

1. Accessbility: This feature helps to identify unwanted or erroneous selections if accessed from the initial selection. This property is verified using an access scheme that is a directed statement in which nodes have Petri network states, and parentheses are the transition between two states. We get the diagram from the initial state M_o) and form the

root node, and then add all the accessible states. For each case we review all the transitions that can be crossed from [13]

2. Liveness :Petri networks are described as having four levels of vitality. A network has a L_k level where $k \leq 4$ if all network transitions are

3. Boundness We say about the Badri network that it is limited to the degree K when each position in the network in each case contains the most K symbol, and if all its access schemes are finished.

4. Safeness is safe if the number of symbols does not exceed one for any selection, and the Petri network is secure if all transitions are secure.

5. Conservation to achieve this property if the sum of the symbols in the grid is the same for any selection.

6. Reversible: We say that the Bahri network is reversible in case it is always possible to return to the initial state from any other caseد.

7. Conflict: A conflict occurs when the same position is in the input of two or more transitions. We call a conflict an effective conflict if the number of symbols in the common position is less than the number of transitions in the output.

8. Persistence (permanence): Petri network achieves continuity property for each accessible M determination then the following property is realized for each pair and T activated in M then (TT forms a transit chain in the TT chain as well) meaning that the activated transition is deactivated by crossing this transition only[14]

٣-٣The mathematical formula for the Petri network Petri network consists of five symbols (PN=(P,T,F,W,M0) where:- P={P1,P2,P3,………,Pm} which is a finite set of places T={t1,t2,t3,……,tn} A finite set of transitions W=F→{1,2,3,……… N} weight of function M0=P→{1,2,3,……..,M}marking start $P \cap T = \emptyset$ and $P \cap T \neq \emptyset$

1- To describe the marking process in the Petri network, there are rules for transition or activation, which are: - The transition (T) is activable if the entry (P) of (T) is at least PT) token) where (W(PT) is the weight of the arc (Arc) from (P) to (T) - activability (T) can occur or not occur depending on what is required of the system [15]

The activation process is the activation of (T) and moving W(PT) token molecule from each input (P) for (T) and adding (W(PT) symbol for each output (P) for (T) so that PT token) is the weight for the transition process from (T) to (P), the rules can be clarified by taking the well-known chemical equation of the water-forming reaction as an example where there are two symbols in each of the places as in the illustration (1) (a) where two molecules of each of hydrogen and oxygen appear and after the activation process The initial state will change according to the three rules mentioned as in the illustration (1) (b)

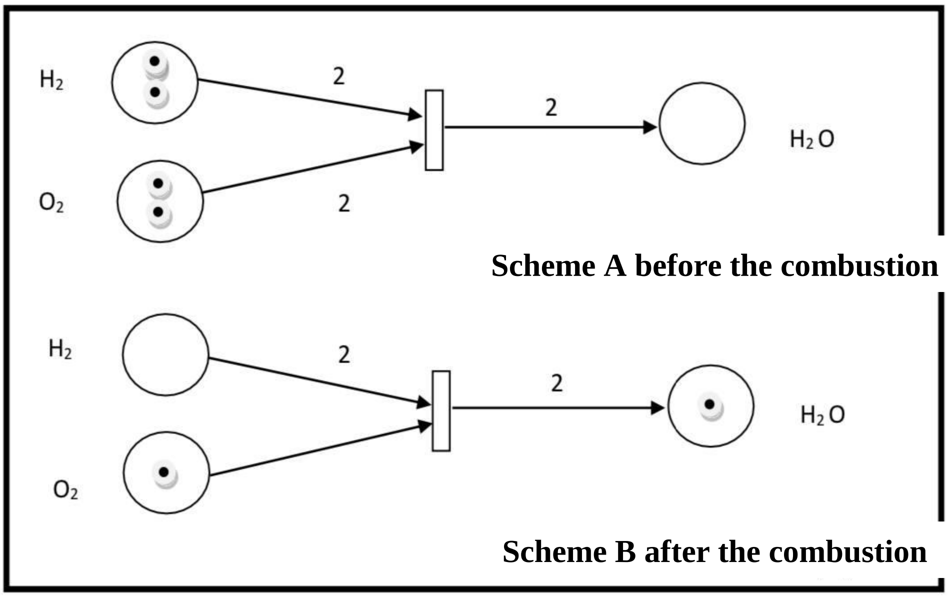


Diagram (1) Illustrate the Chemical Water Equation Using Petri

There is also a concept related to a network, which is that each place has a specific capacity of symbols, meaning that they must have a fixed number (p), which is the upper limit of the symbol numbers that places can store at any time. From this point of view, a fourth rule of activation rules can be formulated, which is that the number of symbols in each transition from (P) to (T) must not exceed the amplitude of (p) after the activation process [16]

٣.٤Representing the problem using a Petri grid The problem of a system that contains more than one processor and requires sharing sources can be represented at the same time, if we take, for example, the existence of each of them separately, the representation is in the following diagram (2)

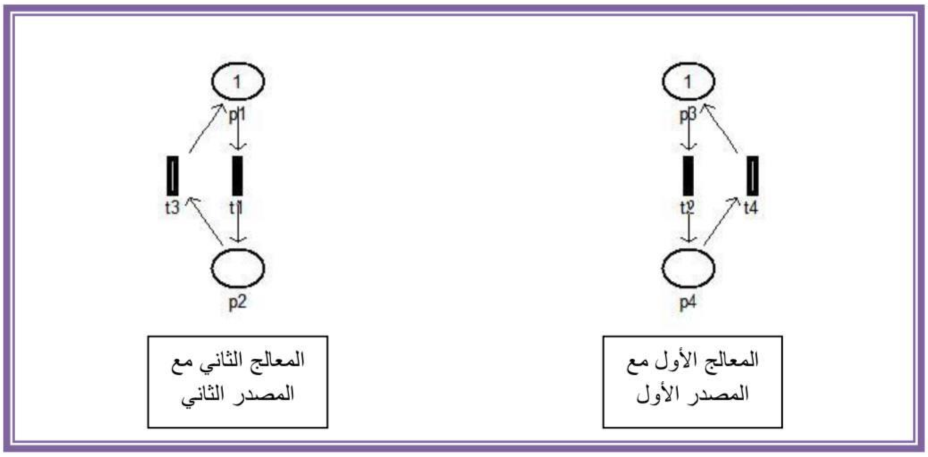
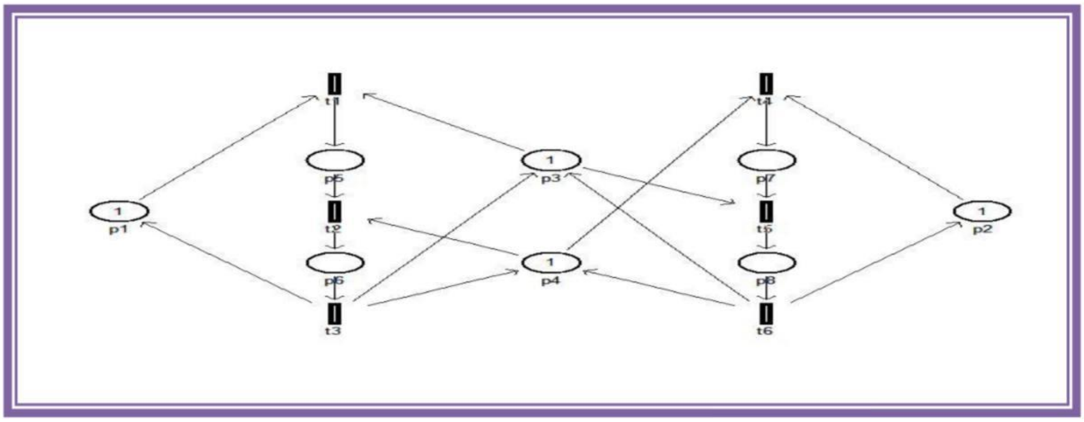


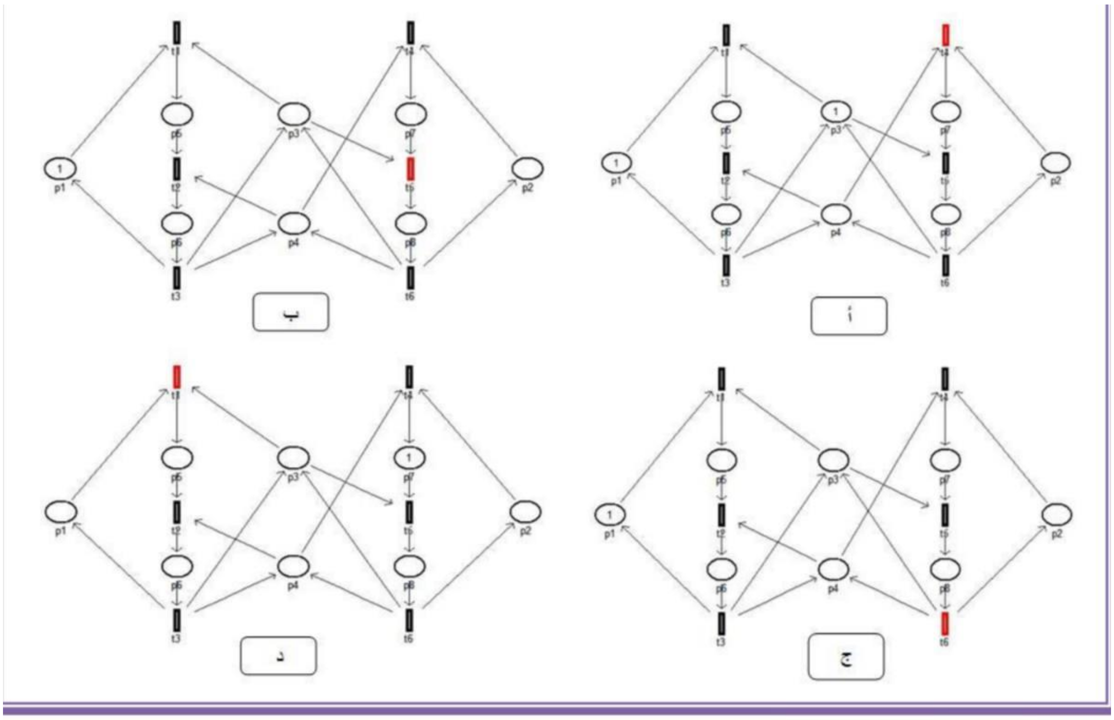
Chart 2 representing two processors with two sources using Petri Net

The processors were represented by P1,P3) and the two sources were represented by P2,4 and the number one within the circuit is the symbol (Token), which represents the process of starting the source request by the processor and {11,12,13,14) represents transitions, which is necessary within the concept of Petri network to complete the application process,(12) The above illustration is suitable for representing the use of the source by the processor in autonomous systems that do not contain several processors and sources, which avoids falling into the fatal lock ((Deadlock 3−5) Scientific application using Petri net representation of the same process on multiprocessor systems will be as in the following



Scheme 3 Process Sharing Exporters for Both Processors

The system was represented by a group of places, namely {P1,P2,P3,P4,P5,P6,P7,P8} And transitions {t1,t2,t3,t4,t5,t6} and to start the process of requesting exporters by processors at the same time, a code (Token) was added in each of {p1,p2,p3}, where a series of activation process will take place as in the following plan (4)



Scheme 4 Execution of a series of firing operations

The activation process will take place either in both the t1 or t4, and the following is a series of steps using the Petri network model: - 1- Activate t4 as in (a) The code will be reduced from (p2,p4) and a code will be added to (p7). 2- Activate t5 as in (b) the code will move to (p8)

3 – activate t6 as in (c) the codes will move to each of (p2, p3, p4) 4– activate pt as in (d) the codes will move to (p5) and activate t4 again as in point number (1) then we will notice that there are two symbols in both (p5,p7), which means that the continuity of the system fails and it falls into the problem of deadlock because t2 cannot be activated, t5 means the maximum activation of t is (5) times only, as in the following diagram

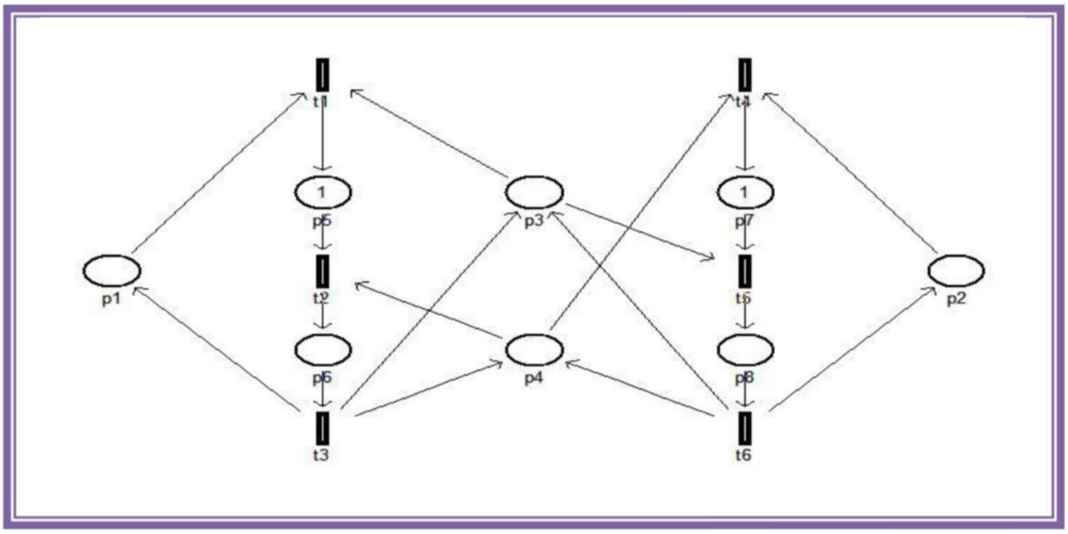


Diagram 5 shows the occurrence of the system in the Deadlock state

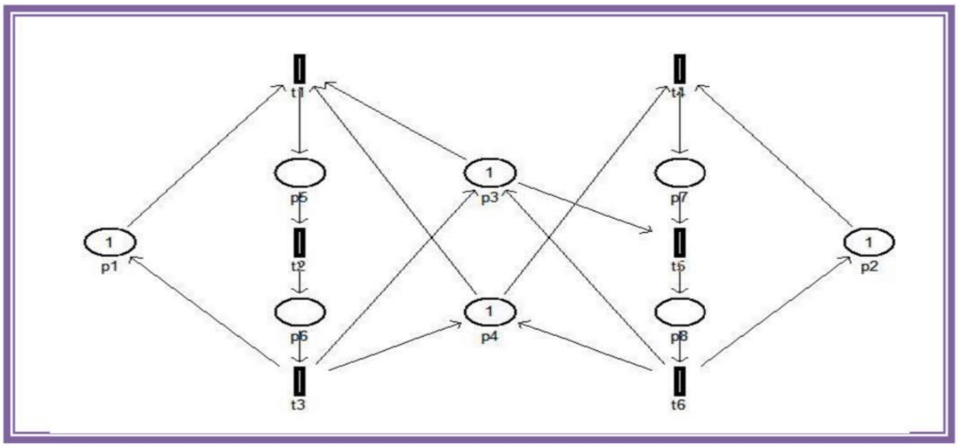
The following table (2) is a statistic that represents the process of reducing and adding symbols from places where: – Event: Represents the number of times the event occurs, that is, the number of times the t is activated. Arrival Sum:– Represents the number of symbols that have been added to places. Throughput Sum: Represents the number of shorthand symbols of places

Table 2 Places Statistics through 5 Event

جدول (2)إحصائية الـ Places خلال 5 Event

Place Name	Arrival Sum	Throughput Sum
p1	0	1
p2	1	2
p3	1	2
p4	1	2
p5	1	0
p6	0	0
p7	2	1
p8	1	1

٦–٣Using the Look head Feedback method to get rid of the Deadlock problem The Petri network can be relied on to solve and organize many problems by using and pairing them with different techniques and methods, and using the feedback method and applying it to the Petri network will provide the possibility of getting rid of the deadlock obstacle by providing a single path for symbols (Tokens) that allows the system to continue working and avoid failure, this means that the system continues indefinitely or stops according to the programming of the system by the user, such as using the number of times the event For (t) or use a specific time for it, the following is a paraphrasing of the question with diagram (5) with feedback[17].



Scheme 5 Using the feedback method to abandon Deadlock

The feedback process was done by linking an arc betw een the p and t instead of the previous formula, which was p4 and t2, and thus we succeeded in creating an actual and practical balance between requests by therapists and accurately determining the process of sharing the two sources by any processor without falling into the case of a fatal lock, and this is a series of events in which the two sources were used unhindered represented by an illustration (6) below (single arrows represent the active transitions). The activation process is done either in the t4 or t1, the following is a series of steps using feedback in a model of the Petri network: 1- Activate t4 as in (a) The codes will be reduced from (p2,p4) and a code will be added to (p7). 2- Activate t5 as in (b) the code will move to (p8)

3−Activate t6 as in (c) The code will go to p2, p3, p4 4− Activate t1 as in (d) The code will move to (p5) 5− Activate t2 as in (e) The code will move to (p6) 6− Activate t3 as in (f) The codes will move to p1p,3p,4

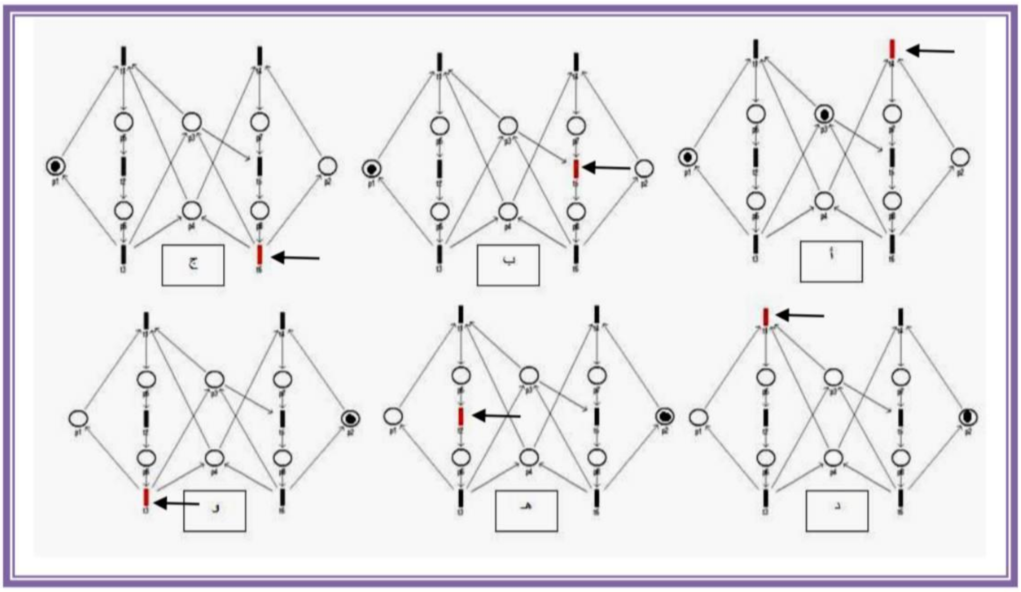


Chart 6 Series of firing operations using feedback

Thus, the process will be repeated again, starting from point (1) to (6) without an intersection in the implementation process, the following is a table (3) statistic representing the process of reducing and adding symbols from places where the number of events (Event)

Table 3 Places Statistics during 10 Event

جدول(3)إحصائية الـ Places خلال 10 Event

Place Name	Arrival Sum	Throughput Sum
p1	1	2
p2	2	2
p3	3	4
p4	3	4
p5	2	1
p6	1	1
p7	2	2
p8	2	2

Chapter Four

Conclusions Petri Grid can be used as visual communication similar to a flow chart or as a diagram template or in grids [18], and is also used in describing both the practical and theoretical part. For example, from the global scale and called the verbal model (Semantic Model) The second part is concerned with the language model (High-level Petri Nets) depending on the difference between the tools in the language and is called as there are several versions of the Petri network we will address the part (Petri Net Markup Language (PNML) There is a Petri network called (Baukasten), which will support and develop programs and applications and this support includes the following tasks:

1- Creating an appropriate petri network for the development of programs.

2- Using Petri network technology that serves the application development area

3- Finding a suitable type of Petri network suitable for applications

4- Finding the appropriate tool

5- Provide solved examples that help application developers. [19] The most important problem that occurs in systems that work simultaneously and in parallel is the problem of deadlock, which is a common problem due to the presence of a certain number of processors Prevent Avoidance Detection and Recovery

Conclusion

Systems that suffer from the problem of deadlock due to mismanagement of the use of resources that must be available to implement programs, whether it is a set of files, input and output devices, etc., can be treated in a way that makes it impossible for the system to fall into this problem. As a result of the application of the research, it was concluded that the use of the Petri network in addressing the fatal lock by integrating the network with the lookhead feedback method led to positive results that made the use of the hybrid network concept with operating systems achieve results at a high level of accuracy, and this is exactly what real time systems need from the sensitivity of using sources and the time accuracy required in these systems. Petri network compared to previously known methods provides solutions that make it impossible to fall into the fatal lock during any moment of the system's operation due to the way the network is designed and made to operate according to an accurate pattern that does not accept error

Sources

[١]Maria and K. Brosig, “Architecture-Level Software Performance Models for Online Performance Prediction,” PhD Thesis, Universität des Landes Baden-Württemberg, 2014.

[٢]R. Jain, the Art of Performance Evaluation. New York: Wiley-Interscience, 1991.

[٣]S. Liu, "Formal Modeling and Analysis Techniques for High Level Petri Nets," PhD Thesis, Florida International University, 2014.

[٤]W. Reisig, Understanding Petri nets: Modeling techniques, analysis methods, case studies. Berlin, Germany: Springer-Verlag, 2013.

[٥]P. J. Haas, Stochastic Petri Nets: Modelling, Stability, Simulation. New York: Springer Series in Operations Research, 2002.

[٦]I. Al-azzoni, "On Utilizing Model Transformation for the Performance Analysis of Queueing Networks," J. Softw. Eng. Appl., pp. 435-457, 2018.

[٧]M. Silva, "Annual Reviews in Control Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field," Annu. Rev. Control, vol. 37, no. 2, pp. 191-219, 2013.

[٨]V. Devedz, "Petri net ontology," Elsevier, vol. 19, no. 4, pp. 220-234, 2006.

[٩]J. Billington, S. Christensen, K. Van Hee, E. Kindler, O. Kummer, and L. Petrucci, "The Petri Net Markup Language: Concepts, Technology, and Tools," Int. Conf. Appl. Theory Petri Nets, vol. 2679, no. Springer, Berlin, Heidelberg, pp. 483-505, 2003.

[١٠]D. Vanderhaeghen, S. Zang, A. Hofer, and O. Adam, "XML-based Transformation of Business Process Models - Enabler for Collaborative Business Process Management 1 Collaborative Business Process Management," Proc. 2nd GI Work. XML4BPM, no. XML for Business Process Management at 11th Conference Business, Technologie, and Web, pp. 81-94, 2005.

[١١]S. Kounev and P. Meier, "Introduction to Queueing Petri Nets: Modeling Formalism, Tool Support and Case Studies," Proceeding ICPE '12 Proc. 3rd ACM/SPE, pp. 9-18, 2011.

[١٢]R. Shiraki and Y. Shinkawa, "Temporal Verification of Business Processes Using BPMN and CPN," Inf. Eng. Express, vol. 3, no. 4, pp.

[١٣]U. Mutarraf, K. Barkaoui, Z. Li, and N. Wu, "Transformation of Business Process Model and Notation models onto Petri nets and their analysis," Adv. Mech. Eng., vol. 10, no. 12, pp. 1- 21, 2018.

[14] E. Badouel, L. Bernardinello, and P. Darondeau, Texts in Theoretical Computer Science An EATCS Series Petri Net Synthesis. Berlin Heidelberg: Springer- rlagVe Berlin, 2015.

[15] L. Wells, "Performance Analysis using Coloured Petri Nets," PhD Thesis, University of Aarhus, 2002.

[16] M. S. OBAIDAT and N. A. BOUDRIGA, FUNDAMENTALS OF PERFORMANCE EVALUATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS, no. John Wiley & Sons, Inc., Hoboken, New Jersey. 2010.

[17] P. J. H. E. M. Fortier, Computer Systems Performance Evaluation and Prediction. USA: Elsevier Science, 2003.

[18] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, Queueing Networks and Markov Chains, Second Edi. New Jersey: A JOHN WILEY & SONS, INC., 2006.

[19] A. B. Bondi, Foundations of Software and System Performance Engineering. Pearson Education, Inc., 2015.

[20] J.-Y. Le Boudec, PERFORMANCE EVALUATION OF COMPUTER AND COMMUNICATION SYSTEMS, 2.3. EPFL, 2018.

[21] S. Fernandes, Performance evaluation for network services, systems and protocols. Cham, Switzerland: Springer International Publishing, 2017.

[22] T. De Gooijer, "Performance Modeling of ASP. Net Web Service Applications: an industrial case study," Master Thesis, 2011.

[23] A. Augusto et al., "Automated Discovery of Process Models from Event Logs: Review and Benchmark," CORR, abs/1705.02288, pp. 1-18, 2017.

[24] B. F. Van Dongen, A. K. A. De Medeiros, and L. Wen, "Process Mining: Overview and Outlook of Petri Net Discovery Algorithms," Springer-Verlag Berlin, pp. 225-242, 2009.

[25] A. Aftab, A. Jilani, and M. Usman, "Model Transformations in Model Driven Architecture," Univers. J. Comput. Sci. Eng. Technol., vol. 1, no. 1, pp. 50-54, 2010.

[26] M. Biehl, "Literature Study on Model Transformations," R. Inst. Technol., no. July, pp. 1-28,2010.

[27] K. Chen, Performance Evaluation by Simulation and Analysis with Applications to Computer Networks. Hoboken: John Wiley & Sons, Inc.

[28] I. Ramos, "A Discrete-Event Simulation Metamodel for Obtaining Simulation Models from Business Process Models," Inf. Syst. Dev. Springer, Cham, 2014.