



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: III Month of publication: March 2018

DOI: <http://doi.org/10.22214/ijraset.2018.3487>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Analysis of Implementation of Distributed Database using Colored Petri Nets (CPN)

Yashaswini J¹, Garima Varshney², Pranoti Joshi³

^{1,2,3} School of computer science and engineering, VIT University, Vellore, India

Abstract: As internet is everything nowadays, the need for storing the information and retrieving the information from different locations is very difficult. To overcome this problem the concept of distributed database came into existence. Here the copy of data is stored in different locations and time required to retrieve data from those locations is less. In this paper we describe communication in the distributed system among some managers. Here the managers should save their database unique. Whenever there is an update it should be reflected to remaining managers in the system and inform them to perform the same update. Here we illustrate three basic concepts i.e., concurrency, conflict and causal dependency. This idea is implemented using the concept colored petri nets (CPN). The implementation of this is analyzed by the software called CPN tool which uses Colored Petri Net ML for assertions and net impressions. This paper also consists of the case study of distributed database and explains the colored petri net concept.

Keywords: Distributed database, colored petri nets, state space analysis.

I. INTRODUCTION

The most important developments in past decade is Distributed Databases. It is a bunch of numerous distributed logically linked databases which are in the computer's network. [1]. The system of distributed database management is defined as the collection of program that permits the database system's management to make distribution clear to end users. Data systems, information and interface components are central in centralized system as shown in Fig. 1.1.

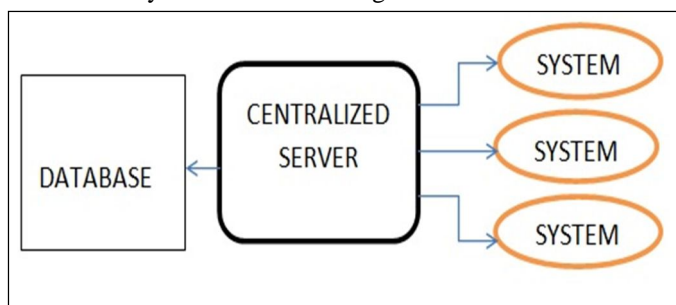


Fig. 1.1. Centralized Database System

Process, Interface components and information are distributed to many locations in the computer network in distributed system as show in Fig. 1.2 which reduces the workload.

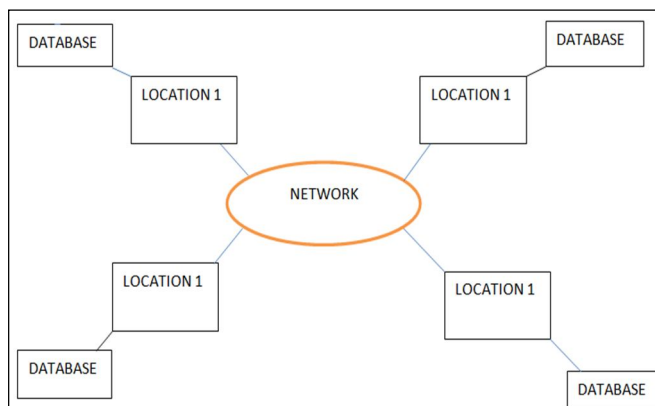


Fig. 1.2. Distributed database

A. Few Conclusions on The Topic of the Distributed Database Systems Are

- 1) Data is stored at numerous sites. Single processor is logically assumed in those sites. Even though some sites are multiprocessor systems, the distributed database systems are not eager with the storage of and management on parallel databases.
- 2) Rather than a multiprocessor arrangement, the processors at these sites are related to each other by a network of computer. The main element here is to highlight on those loosely interconnected processors who operate on their own operating systems.
- 3) The Distributed Database is not just “collection” of files that a nod of computer network can store, it is a database. There is the difference among a Distributed Database and a group of files managed by a distributed file system. To form a Distributed Database, the data that has distributed must be related logically, from where some structural formalism describes the relationships, and data access must be via a common interface at a higher level. The relational model which is used for settlement of the logical connection. Most accessible DDB system explores relational system by assumption.

These conclusions are applicable in today's technology base. On top of local area networks most of the existing distributes systems are built on which each site is usually a single computer. The distributed database is distributing in such way that a site handle one single database managed by DMS as shown in Fig. 1.2. Message passing technique is used for interaction between the hardware and software component's events. Database is an indexed information such that it allows easy retrieval, updating and output of stored data. Different database may have types of database management system and architectures which allocate the implementation of transactions. In distributed database the storage devices are not connected to a common processing unit, instead it is separated physically inside the computer network and interconnected devices.

The structures to evaluate the performance are given by Colored petri nets. The fragment of petri nets is to plot the graph and in other words, time and transition are the two main elements which pays the major role in the graph These nets are based on status and not on events and it makes the precise modeling of status of each possible case. The petri nets model gives discrete event system to any system. The Petri nets besides that provide the validation and evaluation and reliability of the system. For the complex asynchronous processing system, exacter models given by the colored petri nets are used. The nets of the colored petri nets are similar to the petri nets, as known they also takes tokens which is different from each other, because every token in the color petri nets defines some data values that is why for representation of the data some colors are used to differentiate. The color property is added because the respected tokens may vary. The lines of the transitions include the mathematical formulas or phases. In color petri nets, the guards i.e. the Boolean expressions, in which each place defined by that one, arcs and transitions will be considered by their own color.

With the continuation of the paper structure is as follows, Section 2 describes the related literature survey of the distributed database concept and colored petri nets. Section 3 explains the concept of colored petri nets in detail. Section 4 describes how to implement distributed database in colored petri nets briefly and Section 5 shows the implementation part of the paper with screenshots on it description respectively. Section 6 is the conclusion about the paper.

II. LITRATURE REVIEW

In the past years of study, the distributed database is done with the help of queuing Petri Net, in this, model can have explained by the distributed e-business system performance. Although the current available tool and technique [3]. However, currently available tools and techniques are having so much of limitations. In the recent trend the computer should be very active and accurate in terms of time complexity which will be done by the distributed database systems which organize the data in the proper way manner [4]. The graphically represented language to design is the colored Petri Net which is also used for the verification and validation of most of the systems [1]. The system which mainly requires synchronization, communication and concurrency can be designed by the colored Petri Net. The research based on the colored Petri Net. Petri Net defines the several applications and properties based on this Petri Net. The application consists of the protocol engineering and on which phase the colored Petri Net used in the protocol engineering. As known that the nature of concurrent system is non-deterministic that is why the system's execution may proceed in single direction only and it is the most difficult task to make it work. Therefore, the system will consist of the possible execution in the infinite numbers, this will create the gaps and the malfunctions in the designing of the systems, if the person responsible to design miss any of the characteristics which is important [6]. To implement the applications like disaster relief, aircraft controller, power plant of the nuclear is way too complex and the time will be consumed, to avoid this colored Petri Net is to be used. Another way by which the problem of the concurrency control system is managed is by the making the blueprint of the model using colored Petri Net. The main aim of this model is to give or to represent the engineer the better way to look at the how the actual software is going to perform and the way by which the errors and the bugs will easily be debug and detected. Colored Petri Net defines the exact model's functionalities of the system how it is to be build and analyzed and then simulated. The concept of the colored Petri

Net the most advanced technique which proves the correctness of the various varieties of the system. From the researched, it has been studied that the CPN is applied to the model which will be effective in the various use. The communication protocol such as the ad-hoc protocol has been used and modeled to give, the prove of the availability to discovers the error which is existing. In the distributed systems, interactions are fundamental do the two different sets of protocols which is considered is: a) standard protocols which primarily concern communication protocols, b) protocols which are specified. [3]. the limitation of the protocol is also present, that as the number if user and sites increase it will difficult to maintain the same. These nets have been used from last several decades still the survey is going on to make it better and used in the form of the applications [5]. The general testing is also done in the form of the boundedness and liveness in the beginning cases and in the later cases invariant testing of the validation is preferred. In the recent published years, the implementations used on the protocols have been researched. The collection of the data that describes the activities in one or more related organization or the institutes with specific purpose and accurate design of the structure is termed as the database. The huge number of the data is maintained by the or managed by the database management system. [2]. The system in which the co-ordination of the software along with the hardware is connected through the medium and can communicate through the network is known as the distributed database system. The data which are stored in the different place are managed by the system known as the distributed database. The researches done on the distributed system describes the pros and cons of the system along with their types like replication, fragmentations etc [5]. By the distributed database system, the retrieval of the information is easy. in the system of the database the data are systematically organized and the repository of the structured of the information by the index are easy to modify the data after it is being analyzed. The DDMS may include the various types of the systems' architecture and for execution of the transaction. The data may be stored in the multiple computers. DDMS may contain the un-tightly coupled sites by not sharing the physical components [2]. The major problem in the distributed database system is that the deadlock, according to the research work, the detection and avoidance of the deadlock is done in the last past years.

III. CONCEPTS OF CPN

The synchronization of the parallel activities in the automata is best described by the use of the nets called Petri nets. Although the complexity while using it in the real life, also exists [3] and some models of petri nets are:

A. *Transitions of nets based on the place*

The place transition nets are the sequential automata given by the transitions and the events and the tokens based on the events [5]. The representation is formally called as the state machines for which there is a place which contains only one input state and will give relatively output places.

B. *Colored Petrinets*

The communication pattern are more clearly describe with the use of the nets of colored Petri, which gives the process to the information and the patterns are controlled by the using of this net called colored Petri. The colored Petri Net uses the mathematical approaches to show the graphical and crystal-clear diagrams. It is the two-part directional graphs consisting of the number of elements are two i.e. time along with the directions, based on the framework for the process of analysis till the maintenance. This is based on the explicit modeling in the status [9]. The structural and the aspects of the behavior of the discrete systems is given by colored Petri nets models, the limitation of the petri net is well known, to reduce this complexity based in the structures are given by the colored petri nets by the asynchronous systems of the systems of processing. In this Petri net , token are used and are differentiated from each other in the different manner. This nets are named as colored because of this very property of the petri net that the tokens in this consists of the traits, providing the details of the extractor asynchronous models for the complexity process. As the tokens are different from each other the colored property is added to the token to provide the crystal-clear view to the developer and the tester who is going to the test the complexity. The color sets are combined which is based on the mathematical phases give arise to the arcs and the relation of the variables among themselves. The activation of the input arc is given by the transition which is given by the Boolean expression known as guard, which one of the input arc's. On the color petri nets, the transitions and the arcs are depending on the token's colored of their own places.

C. *Fundamental Details of Colored Petri Nets*

Colored Petri Nets is focused on graphical representation for modeling of language which concentrates on concurrency, communication and synchronization of the system. By analyzing their properties, it also helps to validate them [11]. For modeling concurrency, a communication and synchronization Petri net gives the basis of graphical notation and primitives. Describing how

the data should be handled and how it should be defined are afforded on basis of Standard ML [8]. Communication protocols, distributed algorithms, data network and embedded systems are some of the typical applications of colored petri net [8].

1) *CPN model of systems are composed of:*

- a) Various states of system are represented by Ellipses/circles and these states are called places.
- b) Different actions or events that the system will face are represented by rectangles and these are called transitions.
 - Places contain various markers which are known as tokens which are of any non-negative numbers. The data value which are in these tokens lies in specific type.
- c) When tokens are distributed in a net over places it is called as marking.
- d) The flow of tokens and tell how actions which causes modification of the states are described by arcs which are running between places and transitions.
- e) When a transition occurs is described by arc expression how the state of CP-net get change.

When the token exists at all input places and all tokens have same color that of Arc Expression then a transition gets enabled. After this firing occurs and the transition places new tokens at all the places from the output [4]. Many transitions can be enabled after one point of time since the petri nets execution is non-deterministic.

2) *Properties of Colored Petri Nets to show the use CPN*

- a) *Graphical representation:* The graphical forms of anything are very simple to understand. Any person who is not aware or knowledgeable about the concepts of CP-nets can also interpret it. A communication protocol or any algorithm can also be shown by drawing graphs where node depicts as states and action, while are results in describing about the execution of some of the actions and how they go from one state to another.
- b) *Well-characterized set of rules:* These rules uniquely describes about the behavior of each CP-net.
- c) *Description of large systems:* CP nets are applied in many informal systems like any communication protocols, VLSI
- d) It consists of very few, but influential, primitives.
- e) *It has a different description of both states and actions:* many systems have only description only about the states or the actions but not both.
- f) *Offers hierarchical descriptions* That indicates we can create a large CP-Net by relating small CP-Nets. This is similar to that of methods, functions and small module of any programming languages. All this enable us to make very large system in a manageable way.
- g) *CP-nets can be extensive with the concept of time:* This describes that it is likely to use the same modeling language for the arrangement/support of logical properties (such as nonappearance of deadlocks) and routine properties (such as average waiting times).
- h) *Interactive simulation:* CP-nets are represented directly with the help of CPN diagram.
- i) *CP-nets contains a big number of formal analysis methods through which properties of CP-nets can be proved:* It consists of basically four classes of formal analysis methods: building occurrence graphs (which represents all reachable markings),interpretation and calculation of system invariants (place and transition invariants), reductions (which shrink the net without changing a selected set of properties) and checks the structural properties (which assures certain negotiating behavioral properties).

D. Dynamic Behavior of Colored Petri Net

CPN execution occur when a transition is fired. A transition can occur hence following conditions are satisfying: -

- 1) Transition's input places have appropriate number of tokens as per the requirement of the input arc expressions. [3] Each of the input places of the transition if it has variables, then there must be a bound between the input place's tokens and input arc expression.
- 2) The same binding of input arc expressions the transition's guard is evaluated to true.Two actions are taken when a transition gets ready to fire
- 3) Consumption of the token occur from every transition's input places as it is specified by its own evaluated expression of the input arc [3]
- 4) The output places of transitions now contain newly created tokens as described by the equivalent expressions of output arc. If any variables are present, then the token which have appeared in the output place to which the arcs are attached are found by the evaluation of expression of arc with the same binding that has been used to calculate the expressions of input arc and the guard.
- 5)

IV. DISTRIBUTES DATABASE CASE STUDY USING CPN

A. Description of Distributed Database Model in CPN

In this paper we have considered a case study on distributed database to explain the functionalities of CPN. The following case tells a plan distributed database with n different sites (here n is a integer >=3). Every site have a replica of each and every data and it is handled by manager of local database.

$$DM = \{d1, d2, d3, d4, d5, d6, dn\};$$

To make any update to the database, it is permitted to the database manager to make changes but, after making changes it must inform that to all sites on the particular system. Now those managers will do the same changes to their database and send acknowledgment to the sender manager. In this case study we are not at all concerned about the complete message. This is the message set:

$$MES = \{(sender, receiver) \mid sender, receiver \in DM \wedge sender \neq receiver\};$$

In the following message structure database managers are sender, receiver. Now if sender makes any changes in its database, it will update the same in remaining sites as shown below:

$$Message(sender) = \sum_{Receiver \in DM - \{sender\}} (sender, receiver)$$

Here we are making n-1 elements, by using summation, each contains single element.

The following declaration is defined when we put everything together:

Numbers: n: int (n >= 3);

Color set: DM = {d1, d2, d3, ..., dn-1, dn};

MESSAGE = {(sender, receiver) | sender, receiver ∈ sender ≠ receiver};

E = {e};

Functions: Message(sender) = ∑ 1 (sender, receiver);

r ∈ DM - {sender}

Variables: sender, receiver : DM;

The manager of database abstains three separate states: waiting_state, Inactive_state, Performing_state. Four states are available for messages unused state, Sent_state, received_state, Acknowledged_state. Status of the system can be active_status or inactive_status. At first messages are not used. The initialization expressions are then indicated. Multi set of managers are got by DM.all(), and Multi set of messages are got by MES.all(). When updating and sending of message is decided by sender and manager, state change occur from inactive to waiting, while for messages function Message(sender) changes to Sent from unused. Until all the managers have acknowledged the update, a manager has to wait. When one of the manager got the update message its status changes to performing and its message status changes to received. Now that manager have to send acknowledgement to the sender, when it send the ACK, its status changes to inactive and message status changes to acknowledged. After when sender got ACK message from all the managers it changes its state to inactive from waiting and message status changes to unused after acknowledging.

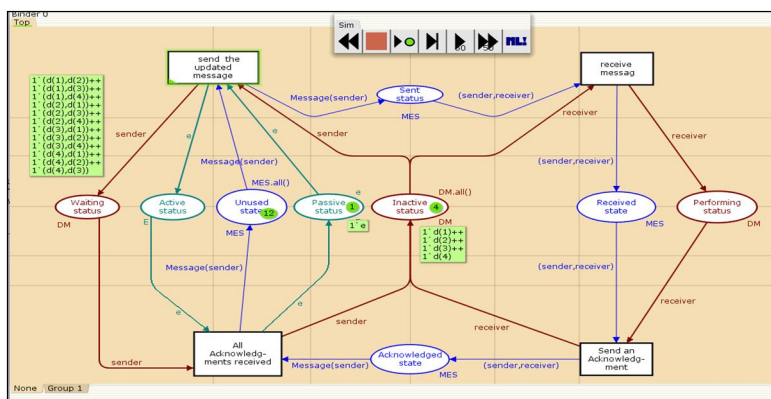


Fig. 4.1. Distributed Database model using CPN

Before initiation of any update it has to be initiated by all other managers. This working is managed by place passive [9]. Passive place initially have e-tokens which are not colored means tokens which contain none of the information. By using this tool we can

implement some of the concepts of distributed database like loss of message , mutual exclusion. Using this tool we can remove the unwanted message as save and use the current message. Consider this example where we can delete not used states. This will result in presence of only those messages which are currently being used. We can also delete Active status and performing status. Sometimes, the redundant place gives more detailed description about various states of the system. when one transition has occurred for any manager and update send has been performed parallel and this concept is called concurrency [8]. We have defined three different ways to define the arcs and places. it have just to make the net more understandable to human other than this it doesn't have any formal definition or meaning. the state changes if data base managers and managers are shown in Fig. 4.1.

B. Explanation of why declarations of CPN ML are equivalent to the declarations outlined

```

▼ Declarations
  ▼ val n = 4;
  ▼ colset DM = index d with 1..n;
  ▼ colset Prod = product DM * DM;
  ▼ fun difference(x,y) = (x<>y);
  ▼ colset MES = subset Prod by difference;
  ▼ colset E = with e;
  ▼ fun Message(s) =
    Prod.mult(1`s,DM.all() --1`s);
  ▼ var sender, receiver: DM;
▶ Monitors
  
```

Fig. 4.2. Variable Declaration for Designing the Model

All Variable declarations are shown in the Fig. 4.2 and are explained line wise in the following:-

- 1) In line1,n is declared as constant and it is assigned with value 4. The use of constant is same as it is in the languages of the programs which indicates that changes in the database managers quantity by changing the declaration of n.
- 2) The line 2, defines the set of the color of DM is declared. There is a inbuilt constructor of color set which makes it possible to declare the indexed color set more easily. The sets of the form {yj, yj+1, .., yn.1, yn} where y is an identifier (string), while integers are j and n. CPN ML doesn't identify variety of styles in font and thus receiver can be identified as y(receiver) or as y. At the initialization level, the inactive the function DM. all() is being considered to produce the multi-set having a one form of each element in DM. The DM.all() which is a function of color which is habitually describe for the set of color DM. This DM can also be defined by the mean of enumeration color set, but by that way, this would have become difficult to define without worrying about the exact value of n
- 3) n line 3 to 5, the color set MES is declared. This declared color set is declared in such a way that it is Cartesian product of DM with itself. Function ML is automatically declared by the CPN ML compiler. we can multiply with multi those two DM multi-sets with in order to receive a PR multi-set. As an example, we have PR. mult(2`d3+1`d4, 1`d2+3`d3) = 2`(d3,d2)+ 6`(d3,d3) + 1`(d4,d2) +3`(d4,d3).
- 4) In line 7, of the declarations we can see the usage of PR.mult.
- 5) The declaration of the function diff., consists of two arguments and examines if those two arguments are different from each other (<> means ≠).
- 6) The declaration of the MES which is the subset of PR which have precisely that only elements in which diff(sender, receive) is true. CPN ML compiler also produces the function MES.all() and it calculates a constant multi-set. This function can be used in the initialization expression.
- 7) In line 6, the color set E is declared, which will be used for allocation of the resource system is done.
- 8) In line 7, multi-sets of messages are being taken from the database manager of mapping message. this can be done by the already defined function of the multiplication i.e. PR.mult that was taken when PR was defined in 3rd line. Multi-set subtraction is represented by using the minus sign.
- 9) In line8, declaration of the two variables sender and receives of the same type as DM.

V. IMPLEMENTATION

In this Section we describe how the model is implemented and how the data is being updated in all the DB sites using CPN tool [7].

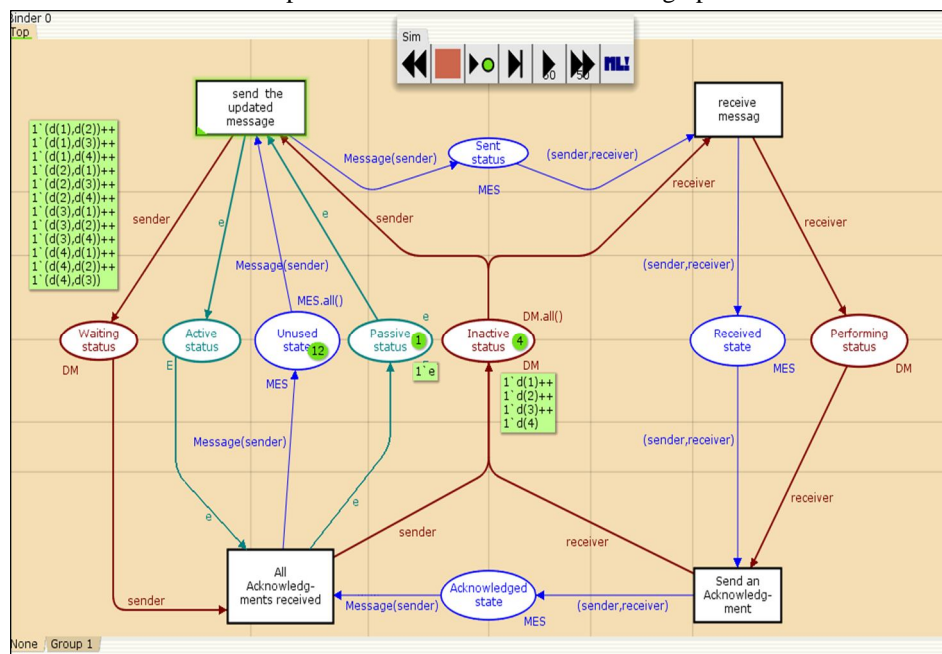


Fig. 5.1. Step1 of implementation

A. Description of step1

In this case study we have considered there are four database managers in a distributed environment and if any one of the database managers updates any data in its database, that update has to be reflected in all the other DB managers. In Figure 5.1 we have initialized four database managers as 1, 2, 3, and 4. And the system is in inactive state. All the variable declarations are described in section 4.

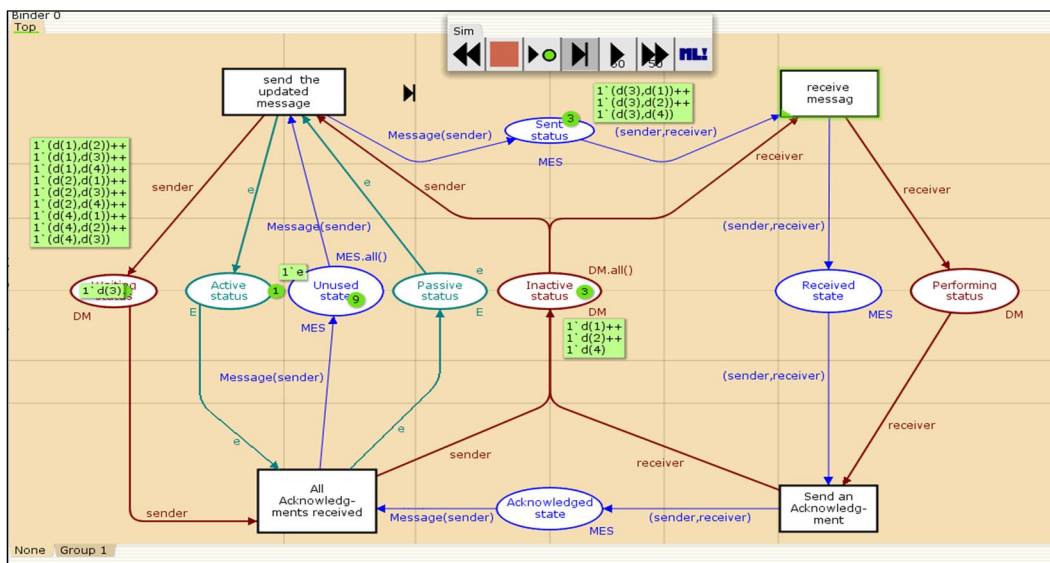


Fig. 5.2. Step2 of implementation

B. Description of step 2

On firing the transaction, randomly selected DM3 as the sender manager and all others are receivers which were selected from the set of product of database managers. It can be seen that DM4 status has been changed to waiting status. And the system is in active state.” sent status” is showing that DM4 has to send its update to DM1 DM2 DM4. This can be show in Figure 5.2.

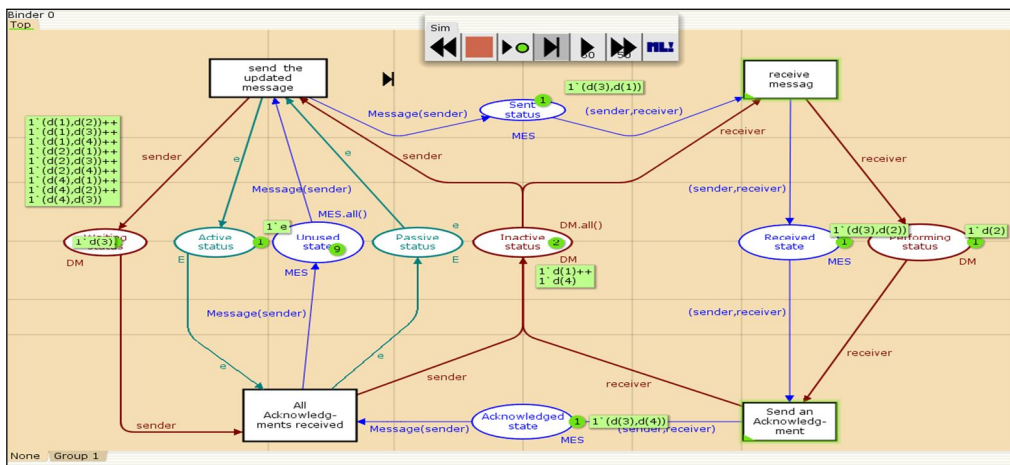


Fig. 5.3. Step3 of implementation

C. Description of step 3

In Figure 5.3, DM3 sent the update message to DM2. Therefore the status of the DM2 has been changed to performing and the status of the message to DM2 is received status. And this will be repeated to all the other DM i.e, DM1 and DM4.

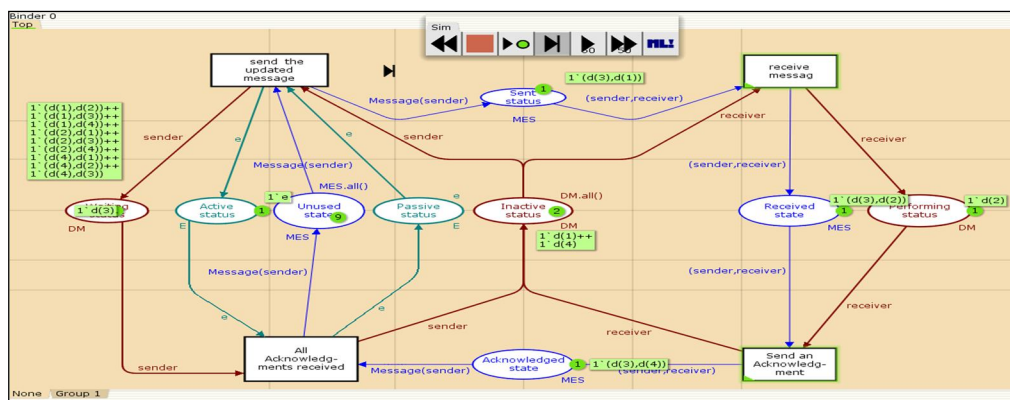


Fig. 5.4. Step4 of implementation

D. Description Of step 4

Sender DM has to get the acknowledgment, after every updation it has sent to other DM. Therefore now in this Figure 5.4, it is shown that DM3-DM3 has changed the state to acknowledgment status and sends the acknowledgment to the sender DM3. Likewise all the other DM's do the same.

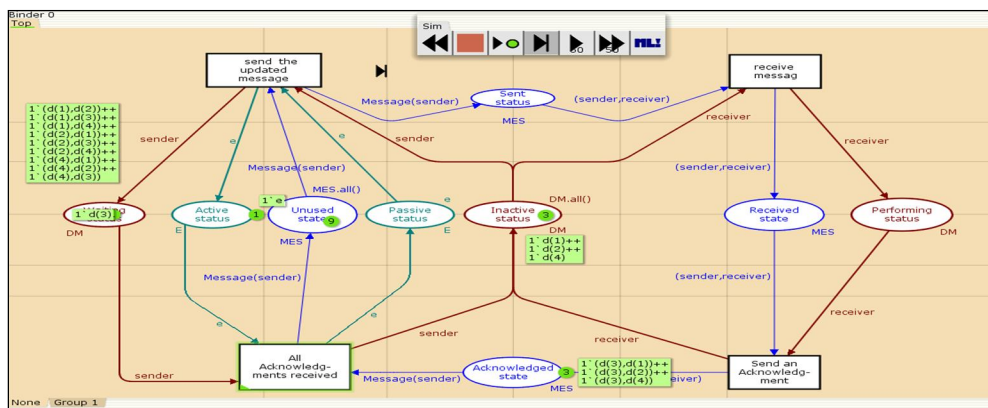


Fig. 5.5. Step 5 of implementation

E. Description Of Step 5

In this Figure 5.5, it can be seen that all the DM's has sent the acknowledgment to the sender DM3. Now the process is complete. From the statistics it can be show that there one component which is strongly connected means all the reachable states are reachable from each other(Figure. 5.6).

```

Statistics
-----
-----

State Space
Nodes: 109
Arcs: 224
Secs: 0
Status: Full

Scc Graph
Nodes: 1
Arcs: 0
Secs: 0
    
```

Fig. 5.6. Statistics of the model

From the Fig 5.6 we can see that at most only one process can be in waiting state which tells that none of the new update should start till all the DB managers have completed the processing of former update .All the integer bounds and multi set bounds are expected.

```

Best Integer Bounds
Top'Acknowledged_state 1      Upper      Lower
Top'Active_status 1      3          0
Top'Inactive_status 1      4          0
Top'Passive_status 1      1          0
Top'Performing_status 1 3      0
Top'Received_state 1      3          0
Top'Sent_status 1      3          0
Top'Unused_state 1      12         9
Top'Waiting_status 1      1          0
Best Upper Multi-set Bounds
Top'Acknowledged_state 1
Top'Active_status 1
Top'Inactive_status 1
Top'Passive_status 1
Top'Performing_status 1
Top'Received_state 1
Top'Sent_status 1
Top'Unused_state 1
Top'Waiting_status 1
Best Lower Multi-set Bounds
Top'Acknowledged_state 1 empty
Top'Active_status 1 empty
Top'Inactive_status 1 empty
Top'Passive_status 1 empty
Top'Performing_status 1 empty
Top'Received_state 1 empty
Top'Sent_status 1 empty
Top'Unused_state 1 empty
Top'Waiting_status 1 empty
    
```

Fig. 5.7. Boundedness property

This below report tells that all the transactions are live and there are no dead transaction instances

```

Liveness Properties
-----
-----

Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All
    
```

Fig. 5.8. Liveness property

VI. CONCLUSION AND FUTURE WORK

As seen the super deal of the works has been conducted and developed by the Petri net in the colored Petri net. It appears that DDMS is the most promising in field of application of theory of nets. The concepts of the Petri net are easily understood by the engineer within less amount of the time using graphical way. Although the better data for the account of representations of the data can be further extended. This Petri net will be used a medium of model to interact with all the models. AS traditional Petri nets have some disadvantages which can be improved in many ideas. After studying and research, the basic algorithm like first come first serves, shortest jobs first, windows system can be applied while distributing the data which will lead to the fast distribution and the will give more clear way to distribute the data.

REFERENCES

- [1] "Distributed Database System Concepts", <http://www.springer.com/us/book/9781441988331>
- [2] "Overview of Distributed Databases", <http://it.toolbox.com/blogs/enterprise-solutions/overview-of-distributed-databases-1622>
- [3] Diaz, Michel. "Modeling and analysis of communication and cooperation protocols using Petri net based models." *Computer Networks* (1976) 6.6 (1982): 419-441
- [4] Mehdi Effatparvar and Seyedeh Solmaz Madani "Evaluation of Fault Tolerance in Cloud Computing using Colored Petri Nets", (IJACSA) *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 7, 2016.
- [5] Kounev, Samuel, and Alejandro Buchmann. "Performance modelling of distributed e-business applications using Queuing Petri Nets." *Performance Analysis of Systems and Software*, 2003. ISPASS. 2003 IEEE International Symposium on. IEEE, 2003
- [6] M. Tamer Özsu and Patrick Valduriez "Distributed Data Management : Unsolved Problem and New Issues".
- [7] "CPN TOOL", <http://cpntools.org/>.
- [8] REF--K. Jensen, L.M. Kristensen, and L. Wells, "Colored Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems". *International Journal on Software Tools for Technology Transfer (STTT)*, 9(3-4):213-254, Springer-Verlag 2007.
- [9] "Approach for software reliability", <http://www.sciencedirect.com/science/article/pii/S0898122103901167>.
- [10] G. Berthelot, G. Roucairal, R. Valk, "Reduction of nets and parallel programs", *Net Theory and Applications*, Lect. Notes in Computer Science, 45, Springer Verlag, 1977.
- [11] B. Berthomieu, "Methods for carrying proofs on Petri nets using their structural properties", to be published, *IEEE Trans. on Software Engineering*, 1982.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)