Formalization of E-Hospital Via Colored Petri Nets (CPN)

[1]Muhammad Zubair Awan*, [2]Muhammad Tauseef Hanif, [3]Saira Latif, [4]Awais Saeed*
[5]Abdullah Faisal*,[6]Imran Siddique*

[1,2,3] Department of Computer Science, COMSATS University Islamabad (Lahore Campus), (Pakistan).
[4] Department of Electrical Engineering, Lahore Leads University, Lahore (Pakistan).
[5,6] Department of Computer Science ,Lahore Leads University, Lahore (Pakistan).
***Lecturer Department of Computer Science and Information Technology Afro Asian Institute, Lahore**

Abstract- Hospital Management Systems (HMS) can be large and hence difficult to analyze with specifically targeted objectives. Even with the ever increasing technological advancements, automation requirements of (HMS) systems to improve a hospital's efficiency and customer based approach demand attention. This paper attempts to combine Colored Petri-Nets (CPN) model with CPN tool for evaluating the behavior of three (HMS) scenarios. State-Space analysis was also performed to verify the behavioral properties of the system.

Key Index: Hospital Management Systems (HMS), Colored petri nets (CPN), Electrocardiography (ECG), State Space Graph.

Introduction

Colored Petri Nets is a graphical oriented language for modelling of systems in which concurrency, communication and synchronization are significant. CP-Nets are a discrete event modelling language combining the capabilities of Petri-Nets with the capabilities of a high level functional programming language Standard Machine Learning (Darabi & Galanter). Standard ML provides the basis for definition of data types, for describing how to manipulate the data and for creating compact and parameterizable models. CPN model of a system consists of places which depict the various states of the system, transitions which depict the various actions. Each place contains a set of markers called tokens. Each of these tokens carries a data value, which belongs to a given type. A distribution of tokens over the places of a net is called a marking. Arcs running between places and transitions describe the flow of tokens and tell how actions modify the state and when they occur. The Arc Expression describes how the state of the CP-Net changes on the occurrence of a transition. (Dumebi, 2017)

Literature Review

The automation of the processing and activities of Hospital Management System (HMS) can invariably contribute greatly to the success, profitability and customer-based approach of such an organization. The use of formal specification creates a formal approach for specifying the underlying functions and Properties of the system. [1] While the use of information technologies is becoming increasingly widespread in healthcare organizations, such as large hospitals, to date these organizations lack unified information systems providing a comprehensive view of the organization's state [2] ,[3] Proposed a timed Petri net model for analyzing and simulating the workflow of a hospital department – starting from the arrival of patients to their discharge- while considering the drug distribution system management, a key process in the department workflow.

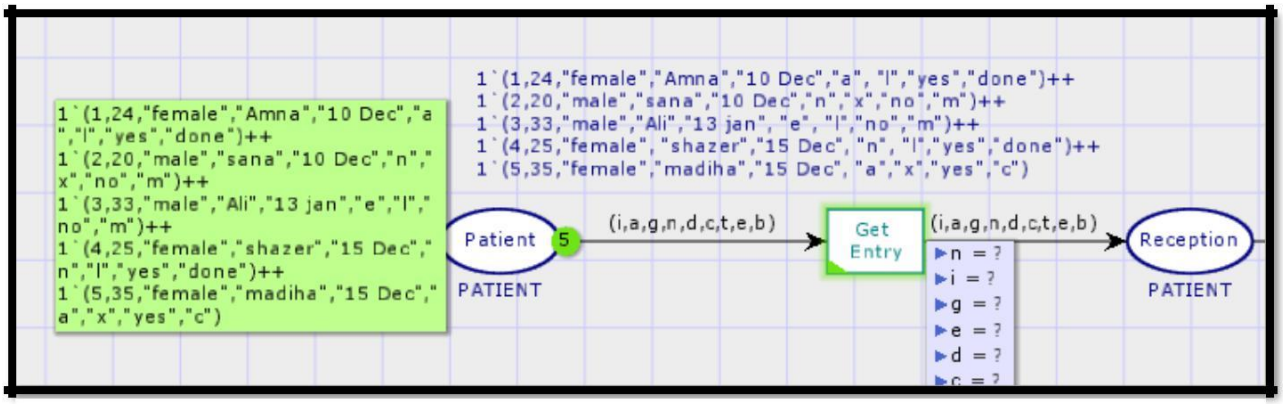
Hospital Management System

Colored Petri nets model application of Hospital Management System flows via places, transitions, and tokens. Each transition moves tokens from the input places to the output places. The placement of a token in a place indicates the location of control within the application thread (Modeling and management of a hospital department via Petri nets, 2010). Hospital Management System CPN model is covering the main cases a patient will go through in a hospital. The initial case is to check in at reception after entering and provide the details about him and his case. Now according to the guard condition, model will identify either the patient will be admitted in the emergency ward or will be assigned a particular time for appointment else will have to go through few lab tests. If the patient case is in emergency zone, he will be directly allocated to the On-Duty Doctor. Will be assigned with the bed in on time, get the treatment and after the treatment will get discharged.

If the patient defines its case as appointment, the details will be checked if actually he has fixed an appointment with someone, and after that he will be allowed to visit the specialist and after the checkup the patient will be discarded. Furthermore if the case of the patient will be laboratory test case than he will either go for Lab test ECG or X-Ray. After ECG or X-Ray, wither the patient reports will directly checked by the specialist or he will go through few more tests like ECG or MRI. Finally, the report will be generated and patient will be examined by the doctor and after examination he will be discharged.

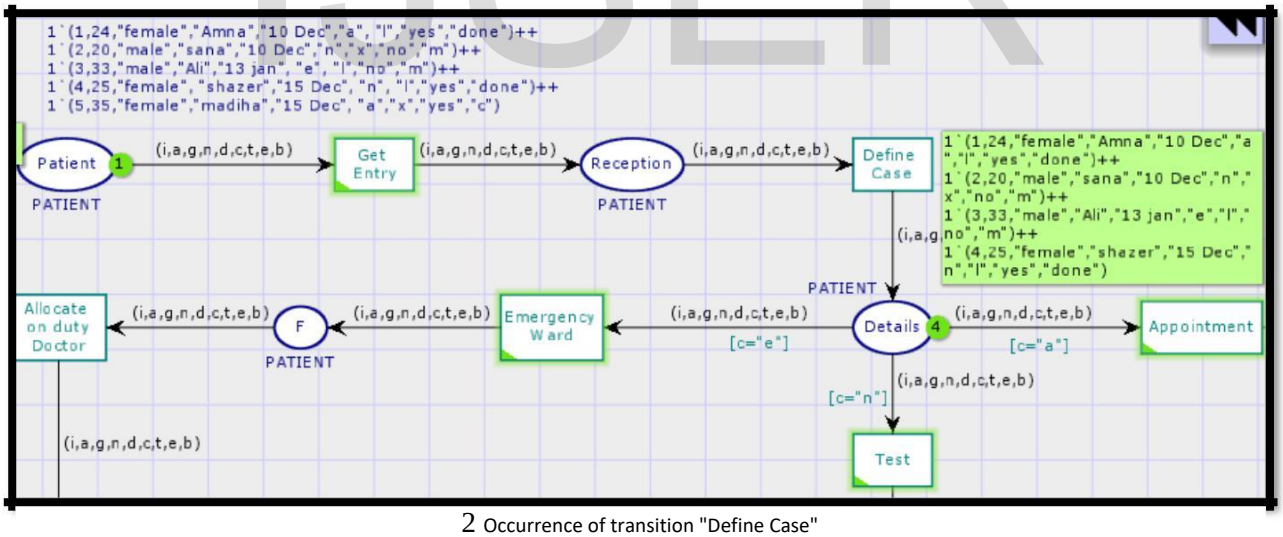
Flow of Model

In CPN Model of Hospital Management System, a number of patients are waiting for the entry to contact to the receptionist. State Patient has 5 tokens and initially “Get Entry” transition is enabled.



1 Initially Enabled Transition "Get Entry"

When Get Entry will be fired, token will be moved to Reception making “Define Case” enabled. Patient will be left behind with 4 tokens. Get Entry and Define Case both will be enabled at the same time. If Define Case will be fired, token will move according to the guard

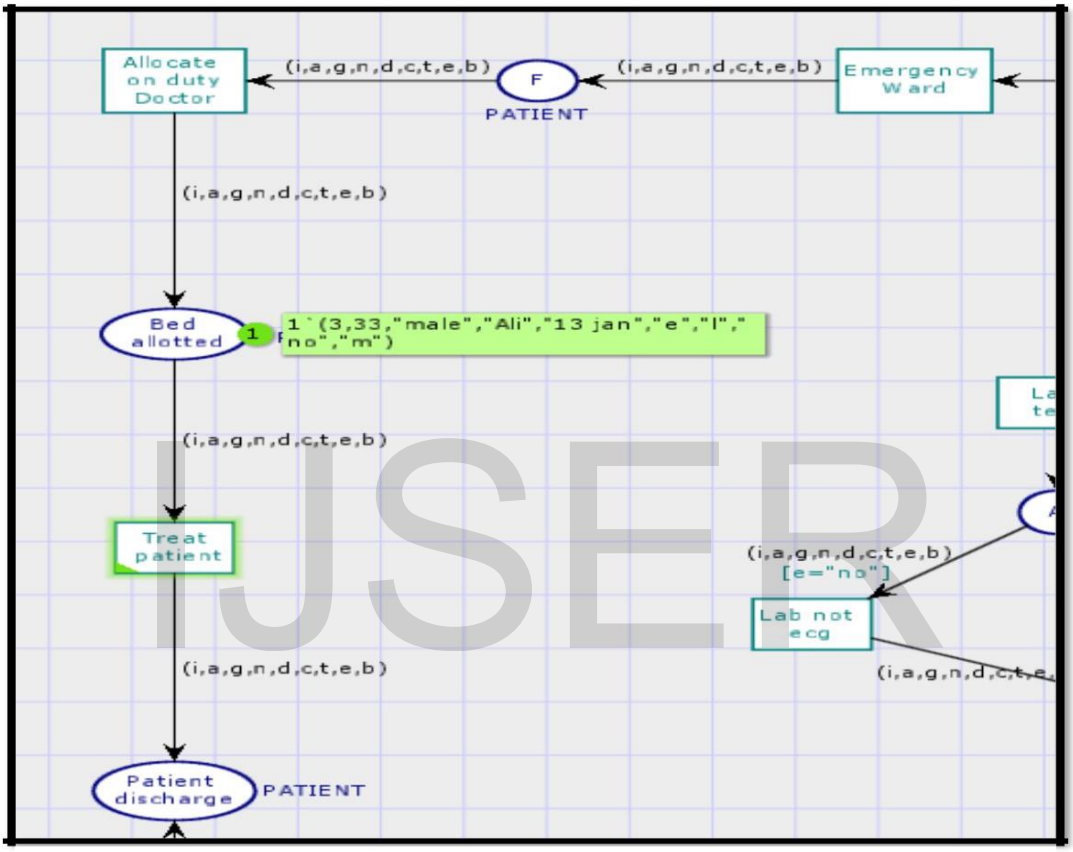


2 Occurrence of transition "Define Case"

Statement [“c=a”] or [“c=n”] or [“c=e”]. In the figure, given below, all the guard statements are

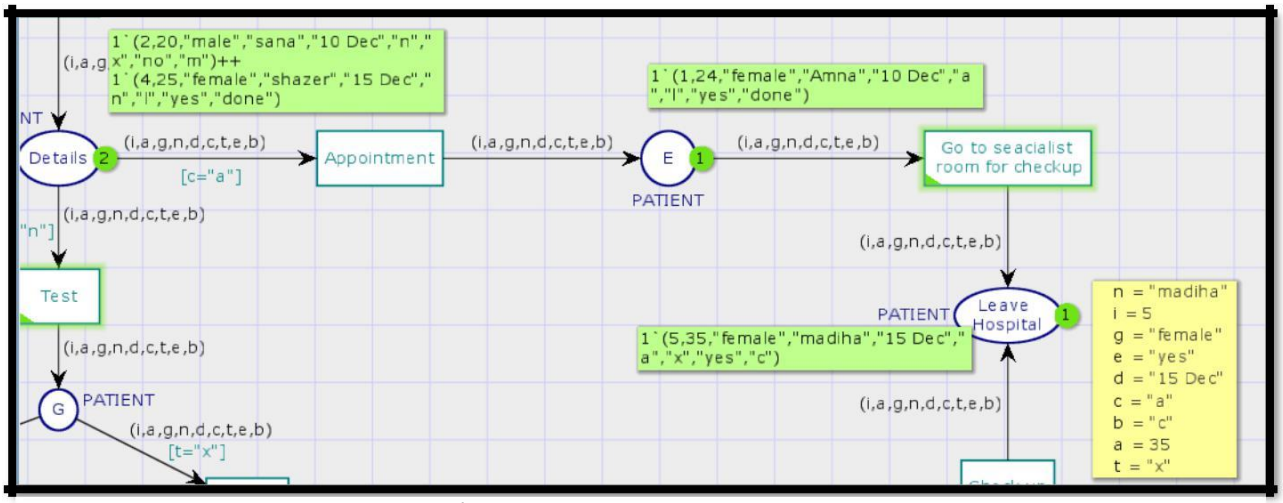
Fulfilled by different patient tokens. So Patient State is left with 1 token behind, where as “Get Entry”, “Emergency Ward”, “Appointment” and “Test” transitions are enabled.

When the transition “Emergency Ward” is fired, Patient will be allocated to the doctor currently available on duty. The bed will be allocated to the emergency patient. Doctor will treat him and after treatment he will be discharged.



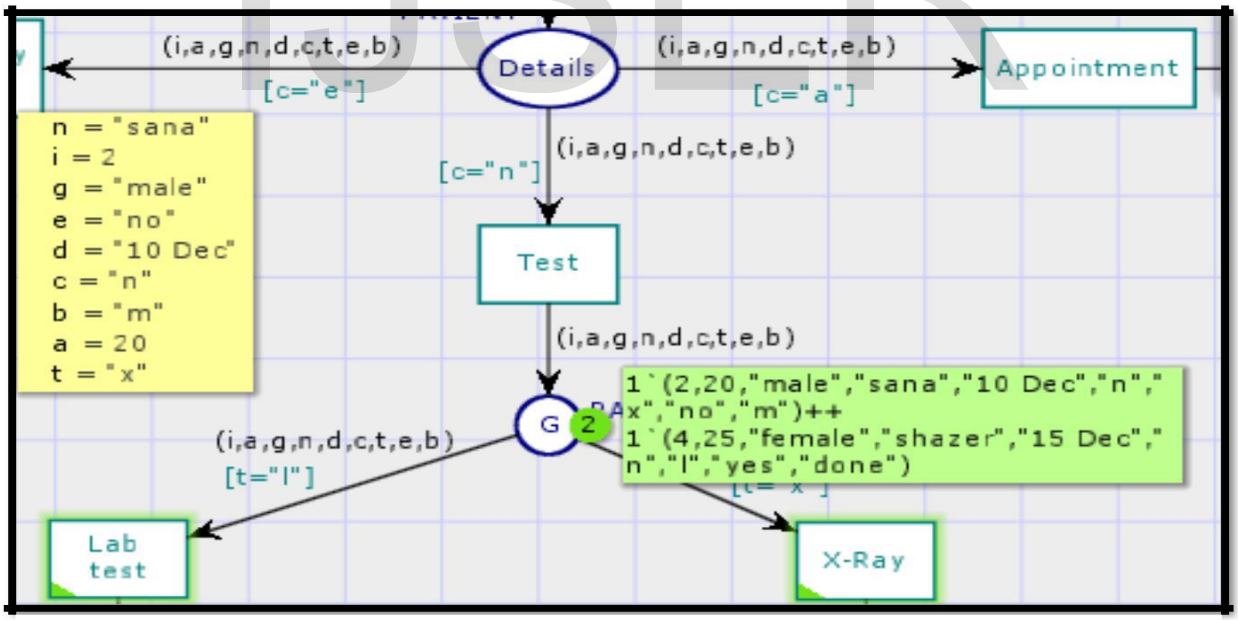
3 Further Processing in Case Emergency Ward.

In case of "Appointment", Patient will be allowed to visit his specialist for checkup and after checkup he will leave the hospital.



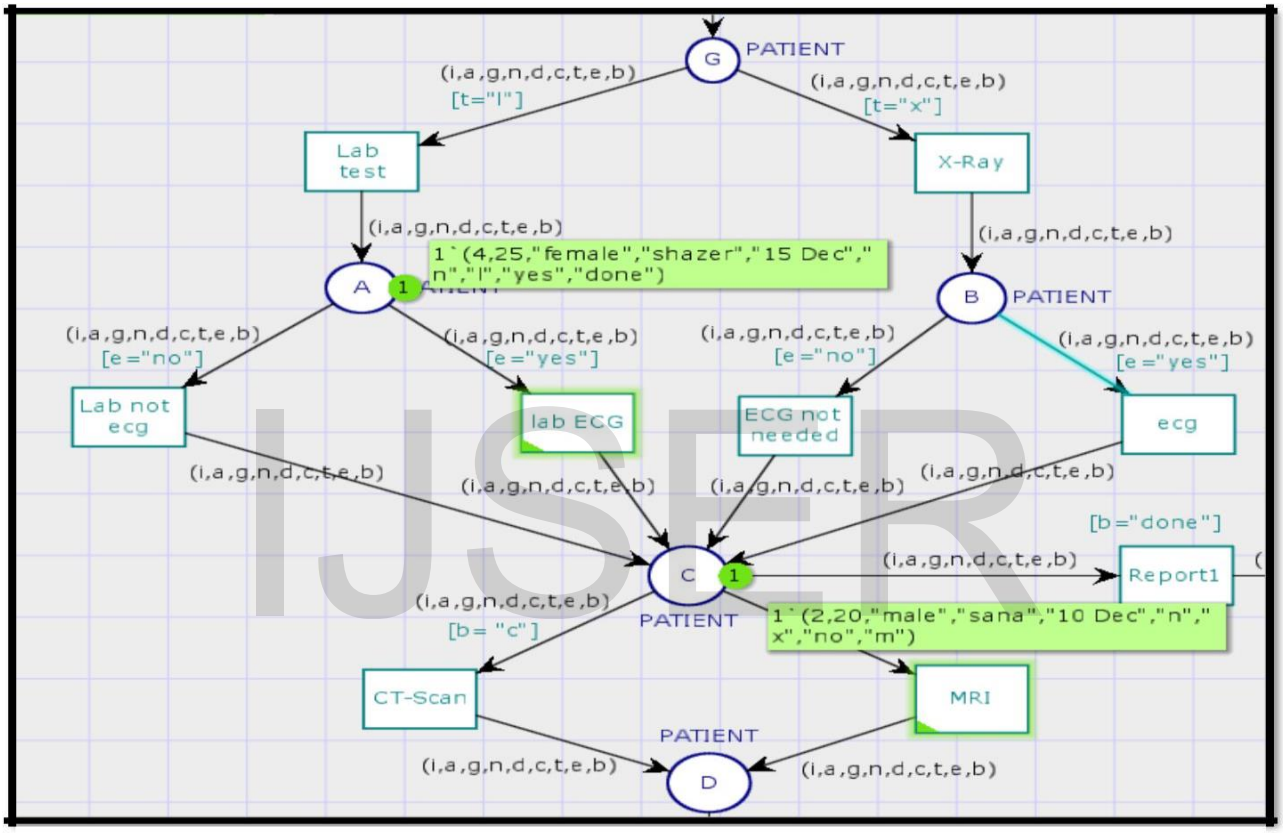
4 Further Processing in Case Appointment.

In guard case [c= "n"], Test transition will be enabled. Patient of guard condition [t= "x"] will move towards X-Ray and [t= "l"] will move towards Lab Test and both transitions will be enabled at this point.



5 Firing of Test Transition

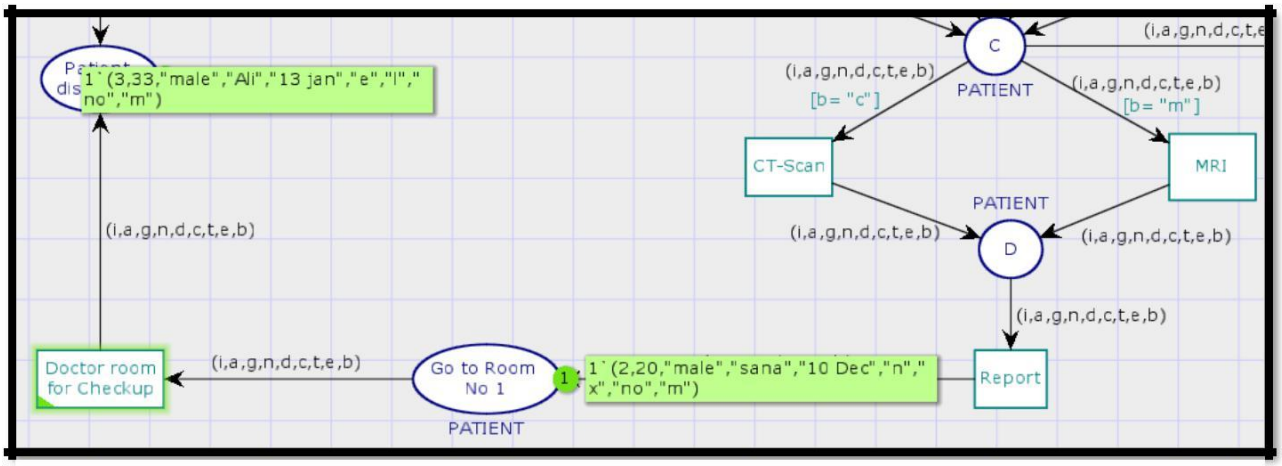
On occurrence of “Lab Test” transition, Guard condition will be checked again if [e= “yes”] or [e= “No”]. In Yes case the patient will have to take ECG test. Similarly, when X-Ray transition is fired, Guard Condition will be checked if [e= “yes”] or [e= “No”]. In No case the patient will not have to take ECG test after the X-Ray. At state C, the patient will either get its report or leave the hospital. Or for further test like MRI or CT-Scan guard will be checked. Guard condition will be checked at three states, if done, [b= “done”], for MRI, [b= “m”] and for CT-Scan, [b= “c”].



6 Further Processing when LAb ECG and XRay ECG is Fired

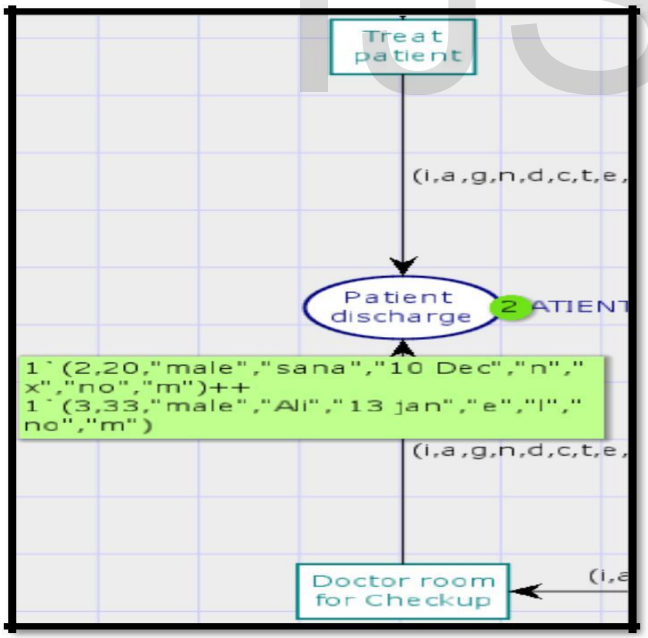
After CT-Scan or MRI, Report will be generated which will be provided to the doctor in Room 1 for further consultation and checkup. After checkup patient will leave. And if CT-Scan and MRI

is not required, then in this case patient will directly check the doctor and leave after the checkup and consultation.

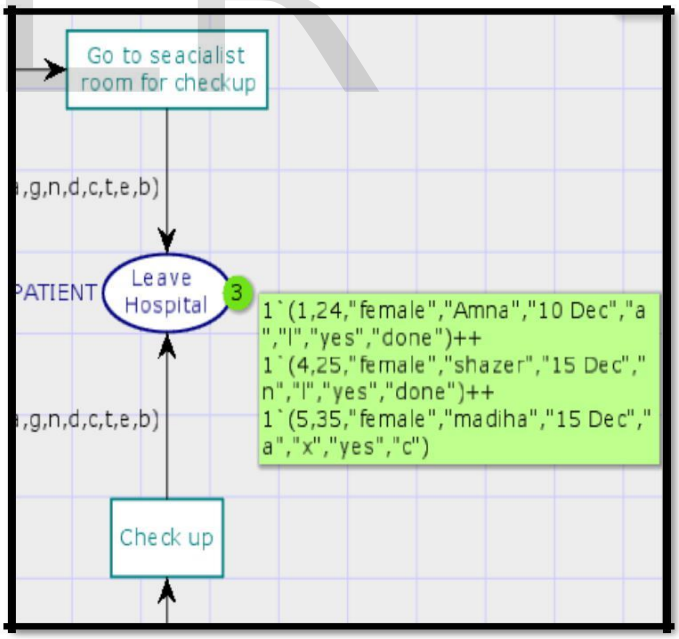


7 Processing after Report transition is fired

The end marking when the whole model is completed will be shown as follows.



8 End Marking1



9 End Marking 2

CPN ML Programming Language

The language used for modeling a Colored Petri-Net is known as CPN ML Programming Language. There are some color sets and variables that are declared before making a model. Following are the color sets and variables defined in this model.

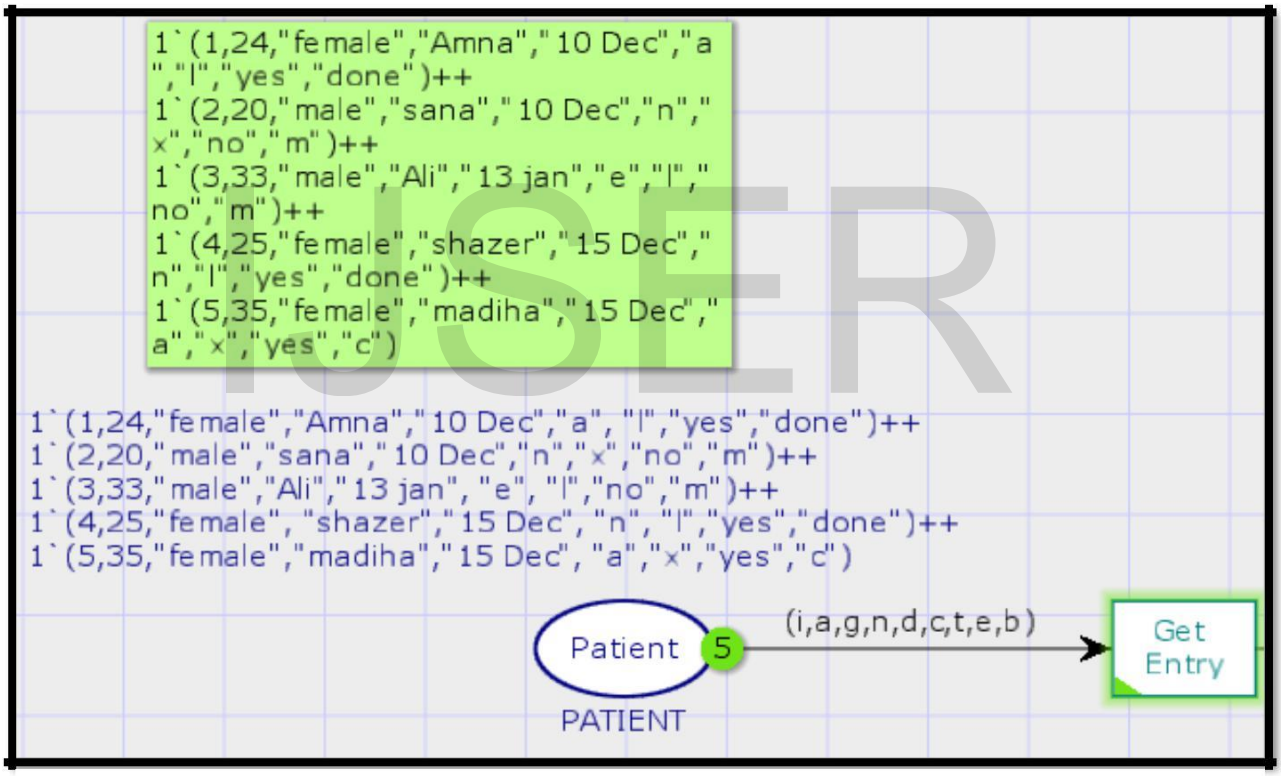
```
Declarations
  Standard priorities
  Standard declarations
    colset ID = int;
    colset AGE = int;
    var count : ID;
    colset GENDER = string;
    colset NAME = string;
    colset DATE = string;
    colset TEST= string;
    colset Case = string;
    colset ECG=string;
    colset Mtest= string;
    colset PATIENT= product ID*AGE*GENDER*NAME*DATE*Case*TEST*ECG*Mtest;
    colset STRING = string;
    var e: ECG;
    var b: Mtest;
    var t : TEST;
    var i: ID;
    var a: AGE;
    var g: GENDER;
    var n: NAME;
    var d: DATE;
    var patient: PATIENT;
    colset UNIT=unit;
    var c : Case;
    colset BOOL=bool;
    colset INTINF= intinf;
    colset REAL= real;
    colset TIME= time;
```


10 Colsets and Variables

Initial Marking

The initial marking of a CPN model is the tokens that are initially present in one or few of the place(s). The presence of these token makes one or few of the transition(s) enabled. Firing such transition(s) start the flow of the model.In this model, there is one place which has tokens initially. However, only one of the transitions is enabled initially. Firing this transition enables rest of the transitions in a sequential way. Initial marking of this model is given as follows.

- Five tokens at place patient enabling “Get Entry” transition.
- Three patient tokens are of female type and two are of male category.
- Arc inscription is in the order of ID, Age, Gender, Name, Date, Case (Emergency, Test, Appointment), Test, ECG, Mtest

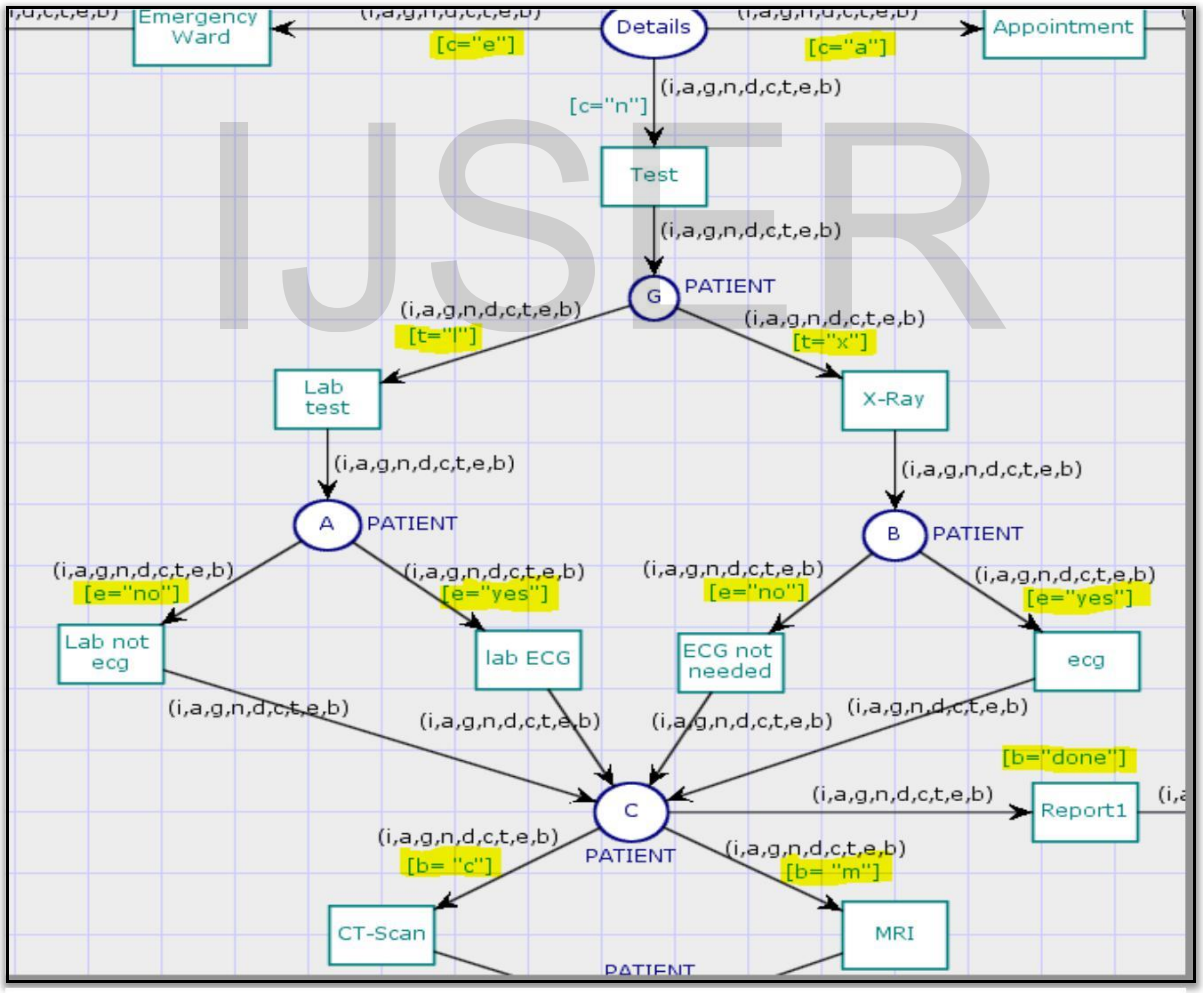


11 Initial Marking

Guards in Transitions

Guards are the conditions in transitions. Fulfilling these conditions will make the respective transitions enabled. In this model, 12 transitions have guard condition. These are:

1. Emergency Ward [c= “e”]
2. Appointment [c= “a”]
3. Test [c= “n”]
4. Lab Test [t= “l”]
5. X-Ray [t= “x”]
6. Lab test: ECG not Required [e= “No”]
7. Lab test: ECG Required [e= “Yes”]
8. X-Ray: ECG not Required [e= “No”]
9. X-Ray: ECG Required [e= “Yes”]
10. CT-Scan [b= “c”]
11. MRI [b= “m”]
12. Report [b= “done”]



12 Guards in the CPN Model

State Space Graph

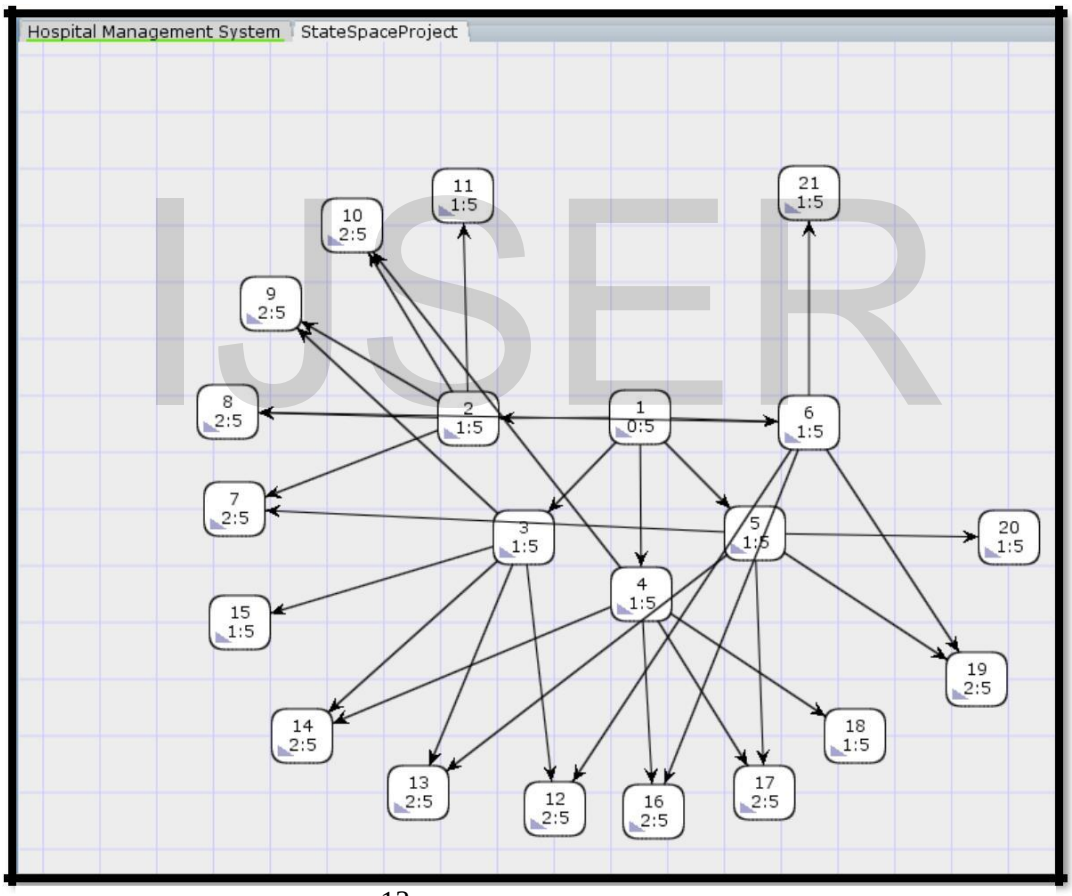
Following are the statistics of State Space of the current model project.

State Space

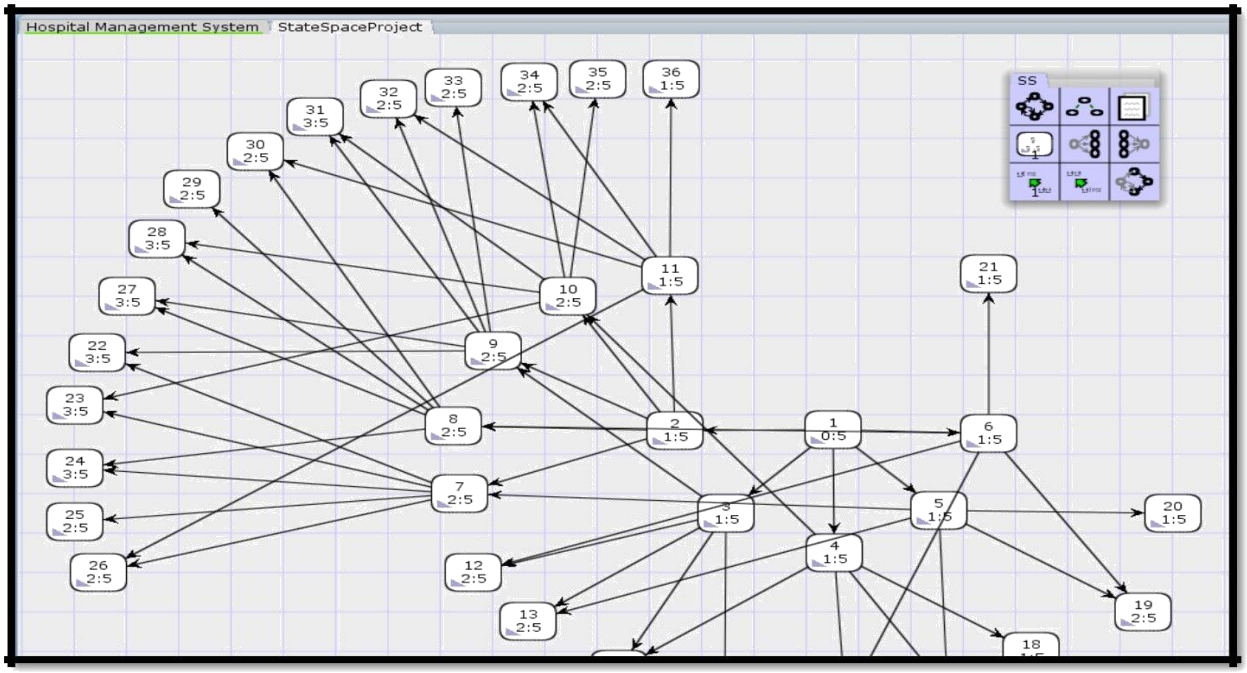
Nodes: 12150
Arcs: 51165
Secs: 91
Status: Full

Scc Graph

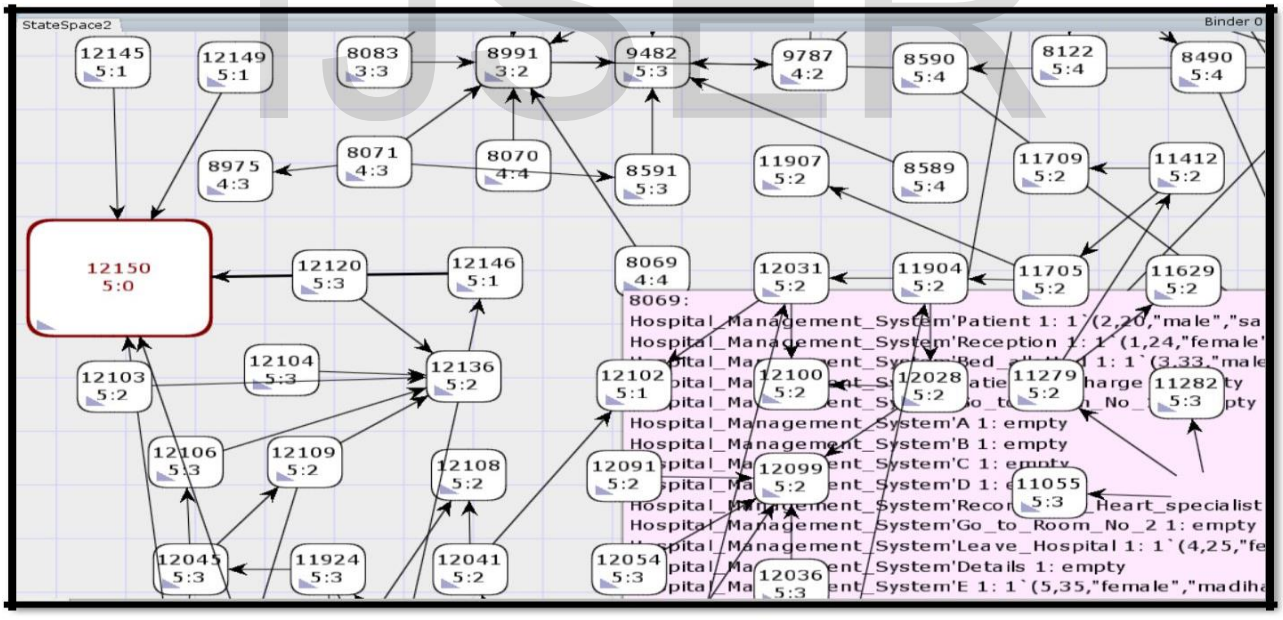
Nodes: 12150
Arcs: 51165
Secs: 2



13 State Space [NodesInPath 1:21]



14 State Space NodesInPath [1:36]



15 State Space NodesInPath

Conclusion

To conclude, the CPN models are actually developed to check the behavior of a system and identify any errors occurring in it. This is done with the help of token flow. Hospital management system is a large system. In this model, few modules have been shown through token flow. Generating its CPN model helps in better understanding of how system works and making it more optimal by adding more services and removing flaws.

References

1. Darabi, Houshang, et al. "Modeling and integration of hospital information systems with Petri nets." 2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics. IEEE, 2009.

2. Dumebi, Okpor Margaret. "FORMALIZATION OF A Hospital MANAGEMENT SYSTEM." International Journal of Computer Science, Engineering and Information Technology (IJCSEIT).

3. Dotoli, M., et al. "Modeling and management of a hospital department via petri nets." 2010 IEEE Workshop on Health Care Management (WHCM). IEEE, 2010.

4. White, Gale, et al. "Patient medication IV delivery pump with wireless communication to a hospital information management system." U.S. Patent No. 6,790,198. 14 Sep. 2004.

5. Jensen, Kurt. "Coloured Petri nets and the invariant-method." Theoretical computer science 14.3 (1981): 317-336.

6. Jensen, Kurt, and Lars M. Kristensen. Coloured Petri nets: modelling and validation of concurrent systems. Springer Science & Business Media, 2009.

7. Ratzer, Anne Vinter, et al. "CPN tools for editing, simulating, and analysing coloured Petri nets." International Conference on Application and Theory of Petri Nets. Springer, Berlin, Heidelberg, 2003.

8. Chiola, Giovanni, et al. "A symbolic reachability graph for coloured Petri nets." Theoretical Computer Science 176.1-2 (1997): 39-65.

Authors' Profile

Muhammad Zubair Awan received his Master Degree (MCS) in Computer Science from University of Lahore, Pakistan and pursuing MS in Computer Science from COMSATS University (Lahore Campus), Pakistan.
zubairawan.uol@gmail.com

Imran Siddique received his B.Sc. degree in Computer Engineering from Bahauddin Zakariya University Multan, Pakistan and pursuing MS in Electrical (Electronics & Communication) Engineering from Lahore Leads University, Lahore.
Imrancpe4u@gmail.com

Abdullah Faisal received his BS degree in Computer Science from University of Sargodha, Pakistan and pursuing MS in Computer Science from Lahore Leads University, Lahore (Pakistan)
Abdullahfaisal495@gmail.com

Awais Saeed received his BS degree in Computer Science from University of Lahore, Pakistan and pursuing MS in Computer Science from Lahore Leads University, Lahore (Pakistan)
awaissaeed111@gmail.com

Saira Latif received her bachelor Degree in Computer Science from Capital University of Science and Technology Islamabad, Pakistan and pursuing MS in Computer Science from COMSATS University (Lahore Campus), Pakistan. Saira.latif93@gmail.com

Muhammad Tauseef Hanif received his bachelor Degree in Computer Science from Punjab University College of Information and Technology (PU-CIT) Lahore Pakistan and pursuing MS in Computer Science from COMSATS University Islamabad (Lahore Campus), Pakistan.
Tauseefhanif05@gmail.com