Modeling of Customer Centric eGovernment system using Petri Nets

Ramani. S,
ISRO Headquarters, Bengaluru – 5600094/
Research Scholar, Dr. M.G.R Educational and Research Institute University, Chennai-600095, India.

Y.S.Kumaraswamy,
Dayananda Sagar College Engineering, Bengaluru - 560078, India.

ABSTRACT
A customer centric e-Governement system provides services through delivery channels and is being operated by internal users (designated functionaries for processing), whenever required. The customers could be citizen, employees, government users, business users, etc . This paper proposes Petri Nets (PN) as modeling language for Customer centric eGovernment system, which is modeled as Discrete event driven system. The graphical representation of PN describes dynamic behaviour of system. Petri Nets enable the visualization of the modeled system state changes. The process cycle can be expressed in terms of occurrence of events and conditions in a system. A simulation tool has been used to develop PN model for eGovernment systems and presented as illustration. The system types considered are Single Function Single Processing Stage (SFSPS), Single Function Multiple Processing Stages (SFMPS) and Multi Functions Multiple Processing Stages (MFMPS). The model developed can be simulated to exhibit system behaviour without construction. The designers and system analysts can build necessary information system based on PN model with desired technology platforms.

Keywords
eGovernance, eGovernment, Discrete event driven system, Petri Nets, system model, process, request, response

1. INTRODUCTION
eGovernment systems are evolving over the years and deployed to customers. The complexity existing in government processes poses challenges in developing eGovernment applications and successful implementation. Hence, there is a need to study eGovernment system in modeling perspective, which will help to understand this real life system and to suggest improvements prior to construction. A model is developed to reproduce the physical system mathematically, in a laboratory or through computer simulation. If the model represents the physical system accurately, then the study of the system behaviour under different conditions or scenarios with different parameter values or input conditions becomes easier. If it is not possible to represent correctly then attempt is made to represent closely with certain assumptions and analysis is made to verify the behaviour of the system. Hence, it is required to understand the characteristics of the physical system clearly and select a suitable mathematical tool for modeling and analysis.

This paper proposes Discrete Event Driven System model for customer centric eGovernment system and Petri Nets as a formal modeling language for graphical representation of system. Section 2 describes customer centric eGovernment system. The problem statement covering scope of study is included in section3. Related works published literature has been highlighted in section 4. Section 5 discusses about discrete event driven system and modeling language Petri Nets. Section 6 proposes the model for eGovernment system based on the study and the models generated using Petri Nets. Section 7 presents PN model developed for issue of Trade licence for a local city government. Conclusions are presented in section 8.

2. CUSTOMER CENTRIC EGOVERNMENT SYSTEM
A customer centric e-Governement system provides services through delivery channels to customers and to be operated by internal users (designated functionaries for processing) of the system, whenever required. The customers could be citizen, employees, government users, business users, etc. The following system characteristics considered for study of customer centric eGovernment system:

- System consists of Hierarchical structure of functions, processes and activities.
- Functions are the functional requirements of system envisaged.
- Processes embed the procedures of government
- Activities contain the actions to be performed based on policies/rules/acts
- External users (customers) request the system for services like information, certificates, licences, permits, payment of taxes/fees, etc.
- Requests are processed by the system
- Internal users perform necessary activities during processing
- Responses generated as outcomes for customer requests are delivered

3. PROBLEM STATEMENT
The research study is aimed to propose a system model for customer centric eGovernment system. Then, a suitable modeling formal language will be proposed to represent system. The applicability of model will be illustrated using typical system structures. The analysis techniques and properties of modeling language will be used to analyse dynamic properties of system. The proposed model can be used for visualization to non-

technical users, system analysts and developers to understand the system clearly. The design and development of information system with this model can be constructed with selected technology platform.

4. RELATED WORKS

The emergence of eGovernment, perceptions of citizens', expectation of citizens' better services from government, more transparent way of doing business with government, a two way path of consultation and collaboration are being discussed in the literature. The six dynamic stages of eGovernment are Information publishing/dissemination, official two way transactions, multi-purpose portals, portal personalisation, clustering of common services, full integration and enterprise transformation. By harnessing the advances in technology, making services more accessible through multiple channels and more responsive by providing 'joined-up' services, the citizen has access to information relating to services through one point of contact (Rachel Silcock 2001). The internet encourages transformation from the traditional bureaucratic paradigm to eGovernment paradigm. Many cities have already adopted one stop shopping and customer oriented principles in web design (Alfred Tat –Kei Ho 2002). The impact of new technology and its assessment in public sector service delivery and citizens' attitudes about government are debated. The interactive features of the World Wide Web to improve service delivery, democratic responsiveness and public outreach have been examined in the content of e-government (Darrel M West 2003).

Petri Nets have been used to model systems like computer networks (Ajmore Marsan, Balbo and Conte 1986), Communication systems (Merlin and Farber 1976; Wang 2006), manufacturing systems (Venkatesh, Zhou and Caudill 1994), discrete event based system(Pan Xinglong, HeGuo 2011), e_government systems(Wang Ge, Ren nan 2008), multi-agent systems(Jose R. Celaya, Alan A. Desrochers and Robert J. Graves 2009), distributed systems (Y.Narahari, K.Suryanarayanan and N.V.Subba Reddy 1989) and E-services(Massimo Mecella, Franscesco Parisi Presicce and Barbara Pernici 2002). There are different model types for modeling business processes. Petri-nets (Jensen 1985) and added-value chain diagrams (Porter 1990) are some of the models reported in literature.

5. DYNAMICAL SYSTEM [6]

The dynamical system concept is a mathematical formalisation for any fixed "rule", which describes the time dependence of a point's position in its ambient space. A dynamical system has a state determined by a collection of real numbers or more generally by a set of points in an appropriate state space. The evolution rule describes what future states follow from the current state. The rule is deterministic: for a given time interval only one future state follows from the current state. A system consists of interacting components and is associated with a function it is presumably intended to perform.

5.1 Discrete state systems

The state variables x(t) represents the state at the time instant t. The path traversed by the system representing various states over the time represents state space(S). In discrete- state models, the state space is discrete set. The dynamic behaviour of discrete-state systems is often simpler to visualize. This is because of state transition mechanism is normally based on simple logical statements of this form " if something specific happens and the current state is x, then the next state becomes x'."

5.2 Discrete Event Systems

When the state space of a system is naturally described by a discrete set like {0,1,2,…} and state transitions are only observed at discrete points in time, we associate these state transitions with "events" and consider a discrete event system. An event should be thought of as occurring instantaneously and causing transitions from one state value to another. An event may be identified with a specific action taken. We can define an event set E whose elements are all these events $\{e_1,e_2,e_3,\ldots\}$. Clearly E is a discrete set.

5.3 Event – driven system

In discrete – state systems, we see that the state changes only at certain points in time through instantaneous transitions. With each such transition we can associate an event. Consider two possibilities, for which there exists a clock through which the time is measured. A) At every clock tick an event e is to be selected from the event set E. If no event takes place, the null event is selected to mean there is no state change. B) Some event e announces that it is occurring at various time instants, which are not necessarily known in advance and not necessarily coinciding with clock ticks.

Every event $e \in E$ defines a distinct process through which the time instants are determined based on the occurrence of the event e. State transitions are the result of combining these asynchronous and concurrent processes. Moreover, these processes need not be independent of each other. This gives rise to the term event-driven system. In discrete – state systems, state transitions are synchronized by a clock or occur asynchronously.

5.4 Discrete Event Driven System (DEDS)

A Discrete Event Driven System (DEDS) is a discrete-state, event-driven system, ie. Its state evolution depends entirely on the occurrence of asynchronous discrete events over time. Many systems, particularly technological ones, are in fact discrete-state systems. For many applications of interest, a discrete state view of a complex system may be necessary.

For the DEDS, the state space is some discrete set $X = \{S_1, S_2, S_3, S_4, S_5, S_6\}$. The sample path can only jump from one state to another whenever an event occurs. Sometimes, an event may take place, but may not cause a state transition. Discrete Event modeling formalisms can be untimed, timed or stochastic, according to the level of abstraction of interest. Two discrete event modeling formalisms available in the literature are a) automata b) Petri nets. Analysis and synthesis issues are then typically addressed by making use of the structural properties of the transition structure in the model.

5.5 Petri Nets[7]

Petri Nets are a tool for the study of systems. Petri Nets are a basic model for parallel and distributed systems. Systems with independent components working together for realizing the goal can be modeled by a Petri net. The basic idea is to describe state changes in a system with transitions. Petri net theory allows a system to be modeled by a Petri Net, a mathematical representation of the system. Analysis of the Petri Net can then reveal important information about the structure and dynamic behavior of the modeled system. This information can then be used to evaluate the modeled system and suggest the improvement or changes. Petri nets (PN) are a graphical tool for the formal description of the flow of activities in complex systems. PN are particularly suited to represent in a natural way logical interactions among parts or activities of a system. Typical situations can be modeled by PN are synchronization, concurrency, sequentiality and conflict. The classical PNs do not convey any notion of time. However, a Class of Timed PN (TPN) can be used for the qualitative analysis of the performance and reliability of the system versus time. The time variables associated to the PN can either deterministic variables leading to the class of models called deterministic PN or random variables leading to the class of models called Stochastic PN (SPN)

6. MODELING OF EGOVERNMENT SYSTEM

Customer centric eGovernment system is a dynamic system and is proposed as **Discrete event driven system** due to the following characteristics observed in the system:

- System can be described in terms of events and actions to realize its functions
- Customers' Requests are events, which occur randomly. Events are actions which take place in the system. Events are discrete and finite.
- Event triggers processing or activity of a particular function
- The state of the system can be described as a set of conditions.
- System moves from current state to next State on occurrence of event
- There are finite number of discrete states in a system
- The preconditions and post conditions of an event control the flow of activities through a series of states until request processing is complete.
- Internal user(s) perform the necessary actions at a particular state. On completion of actions, system moves to next state.
- Response is generated at the end of process cycle.

Petri Nets has been selected as a modeling formal language for this Discrete event driven system. The graphical representation of Petri Nets describes the dynamic behaviour of system. Petri Nets enable the visualization of the modeled system state changes. System can be viewed easily using Petri Net graphical representation. Petri Nets help to describe system hierarchy in the model. The process cycle can be expressed in terms of the occurrence of various events and conditions in a system. [7] Thus, Customer centric eGovernment system is a marked Petri Net, which is a quintuple (P,T,I,O,M) where:

$P = \{p_1, p_2, p_3, \ldots p_{np}\}$ is the set of np places drawn as circles in the graphical representation;

$T = \{t_1, t_2, t_3, \ldots t_{nt}\}$ is the set of nt transitions drawn as bars;

I is the transition input relation and is represented by means of arcs directed from places to transitions;

O is the transition output relation and is represented by means of arcs directed from transitions to places;

$M = \{ m_1, m_2, m_3, \ldots m_{np} \}$ is the marking. The generic entry m_i is the number of tokens (drawn as black dots) in place p_i in marking M.

The graphical structure of PN is a bipartite directed graph: the nodes belong to two different classes (places and transitions) and edges (arcs) are allowed to connect only nodes of different classes (multiple arcs are possible in the definition of the I and O relations). The dynamics of a PN is obtained by moving the tokens in places by means of the following execution rules:

- A transition is enabled in a marking M if all its input places carry at least one token;
- An enabled transition fires by removing one token per arc from each input place and adding one token per arc to each output place.

Given the initial marking M_0, the reachability set $R(M_0)$ is the set of all markings that can be obtained by repeated application of the above rule. More formally, it can be said that t_k is enabled in marking M if : for any $p_i \varepsilon I(t_k)$, $m_i \geq 1$;

Marking M' obtained from M firing t_k, is said to be immediately reachable from m and the firing operation is denoted by the symbol $(M - t_k - M')$. The marking of the Petri Net represents the state of the net. The transition change the state of the Petri Net in the same way an event changes the state of a DEDS (Discrete Event Driven System). The concept of Reachability is essential for the study of dynamic properties of a Petri Net.

6.1 Petri Nets - Events and conditions [8]

The simple Petri Net view of a system concentrates on two primitive concepts: events and conditions. Events are actions which take place in the system. The occurrence of these events is controlled by the state of the system. The state of the system can be described as a set of conditions. A condition is a predicate or logical description of the state of the system. A condition in that state may either holds (may be true) or not hold (may be false). Events are actions and they occur when conditions satisfy (are true). These are called preconditions for the events to occur. The events may cause other conditions, post conditions to become true. Hence, for an event, there may be preconditions and post conditions.

Conditions are modeled by Places in a Petri Net; events are modeled by Transitions. The inputs of a Transition are the preconditions of the corresponding event; the outputs are the post conditions. The occurrence of an event corresponds to the firing of the corresponding Transition. The holding of a condition is represented by a token in the Place corresponding to the condition. When the transition fires it removes the enabling tokens representing the holding of the preconditions and creates new tokens which represent the holding of post conditions. Each Transition and Place with corresponding event or condition can be labeled. This view of a system can be modeled as a Petri Net. A Petri Net execution, which exhibits the system behaviour of the model, is viewed as a sequence of discrete events. The Petri Net structure itself contains all necessary information to define the possible sequence of events. The order of occurrence of the events can be suitably modeled.

A simulation tool was surveyed for generating and analysing PN model for this study in the literature and a suitable tool P3 version 1.0 developed by Gasevic Dragon was found [21]. This tool has been used to generate various case studies for this study. The following are some of the net configurations, which can be used in system modeling to represent the system structures.

6.2 PN- net configurations

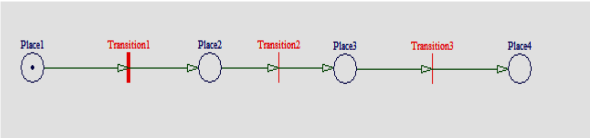

Figure 1: Sequential

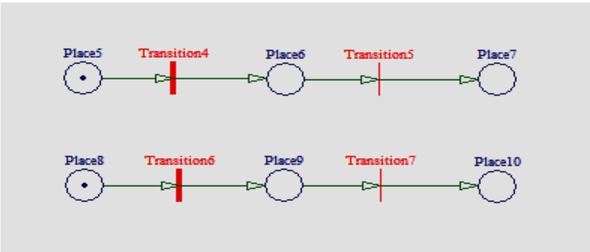

Figure 2: Concurrent

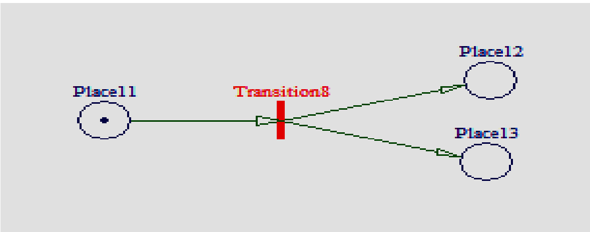

Figure 3: Fork

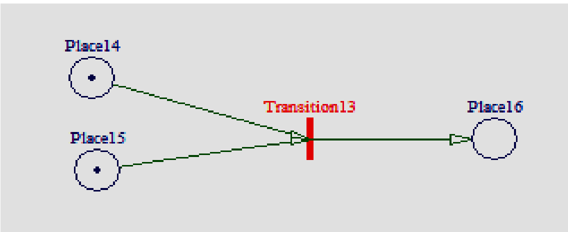

Figure 4: Join

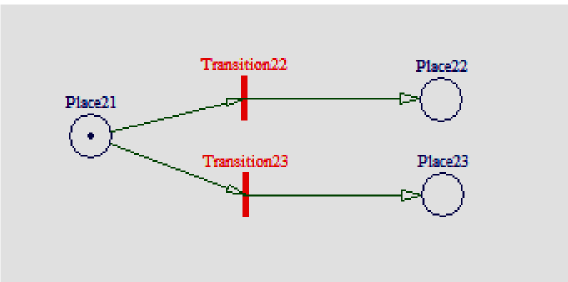

Figure 5: Choice-options

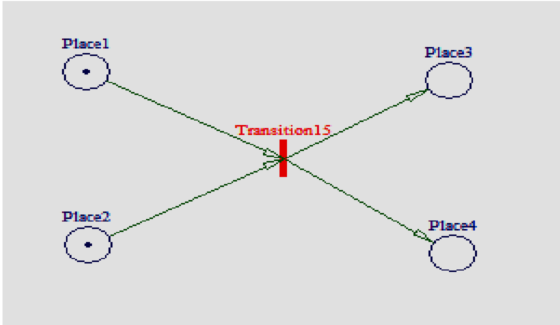

Figure 6: Synchronous

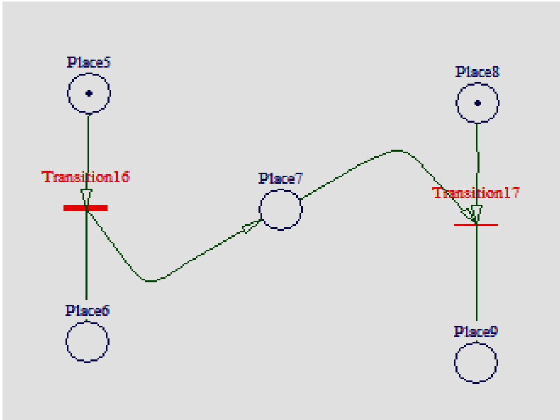

Figure 7: Asynchronous

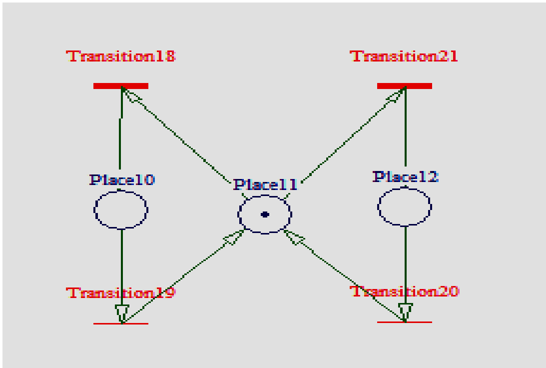

Figure 8: Mutual exclusion

6.3 Analysis on Petri Net model

The analysis provides how closely and correctly the system has been modeled. The analysis techniques available in Petri Nets are Reachability Tree, Matrix-equation and Simulation. The properties of Petri Nets are safeness, liveness, boundedness and conservation, which are used to derive system properties.

6.3.1 Safeness

A place in a Petri Net is safe if the number of tokens in that place never exceeds one in any marking. A Petri Net is safe if all places in the net are safe.

6.3.2 Liveness

A transition is potentially firable, if a sequence of transition firing exists. It leads to a marking in which the transition is enabled. A transition is live if it is potentially fireable in any marking. A transition is dead if it is not potentially fireable. The dead transition cannot fire.

6.3.3 Boundedness
A simple generalisation of safeness is the concept of boundedness. A place is bounded with bound k, if the token count does not exceed k in any marking. A Petri Net is k bounded if each place is k-bounded.

6.3.4 Conservation
A Petri Net is strictly conservative if the total number of tokens is constant in each marking

6.3.5 Reachability Tree
The reachability set of a Petri Net is generated by means of reachability tree. The initial marking M_0 is the root of reachability tree. Starting from the root, the enabled transitions are noted; the firing of an enabled transition produces a new marking, which is represented as a new leaf in the tree. The procedure is repeated to obtain reachability tree.

6.3.6 Matrix – equation
In a Petri Net, there are n transitions and m places. The incidence matrix A is defined as an nxm matrix of integers and its entries are given by $a_{ij} = a_{ij}^{+} - a_{ij}^{-}$ where $a_{ij}^{+} = O(t_i,p_j)$ is the weight of the arc from transition I to its output place j and $a_{ij}^{-} = I(t_i,p_j)$ is the weight of the arc from input place j to transition i.

6.3.7 Simulation

Simulation is a method to run the net with the execution flow to verify the system properties. This can exhibit the dynamic properties of the system and system behaviour. If any undesirable behaviour is observed, it can be corrected appropriately. The advantage is to simulate the system without construction and derive the system behaviour under realistic conditions.

6.4 eGovernment system models
The Customer centric eGovernment system may be of different types. The four generic types with following structures have been taken for modeling in this paper:

6.4.1 Single Function Single Processing Stage (SFSPS)
There is a single function in the system. There is only one stage of processing. The request obtained for this function is processed and response is delivered to customer. There is no action required by internal user during processing. The response is generated automatically by the concerned process. It is a simple system

Request → Process → Response

Examples: Downloading application form, status checking, Downloading information, etc.

The events and conditions of the system has been identified and mapped onto Petri Net model in the following way.

Events
E1. Request is generated
E2. Request processing starts
E3. Request processing completes
E4. Response sent for delivery

Conditions
A - Request arrived and waiting for processing
B - Request is being processed
C - Response generated
D - Process is idle
M – External user idle

Event	Preconditions	Post conditions
E1	M	A
E2	A,D	B
E3	B	C,D
E4	C	M

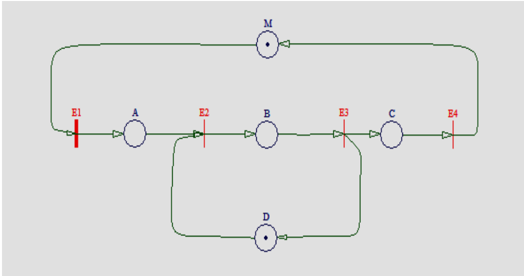

Figure 9: PN model for SFSPS

Analysis: This model has been analysed for system behaviour. The process cycle has been verified and found to be meeting the system requirements. All the places in this net are safe and firing sequence is as per the requirements. There is no dead transition and all are live. The firing sequence is as mentioned below:

E1[ID:t1] → E2[ID:t2] → E3[ID:t3] → E4[ID:t4] → E1[ID:t1]

6.4.2 Single Function Single Processing Stage (SFSPS) including action by internal user
There is a single function in the system. There is a single stage of processing. The request obtained for this function is processed and response is delivered to customer. The action is required by internal user during processing. The process includes activity to be carried out.

Request → Process → Response

Examples: Request for Birth Certificate, Death certificate, Khata Certificate etc.

The events and conditions of the system has been identified and mapped onto Petri Net model in the following way.

Events
E1. Request is generated
E2. Request processing starts
E3. Request processing completes
E4. Response sent for delivery

Conditions

A - Request arrived and waiting for processing
B - Request is being processed
C - Response generated

Event	Preconditions	Post conditions
E1	EU	A
E2	A,F,IU1	B
E3	B	C,F,IU1
E4	C, G,IU2	D
E5	D	E,G,IU2
E6	E	EU

D - Process is idle
EU – External user idle
IU - Internal user idle

Event	Preconditions	Post conditions
E1	EU	A
E2	A,D,IU	B
E3	B	C,D,IU
E4	C	EU

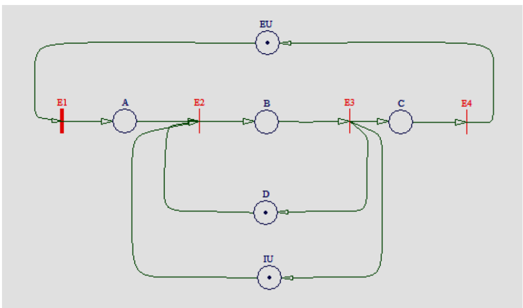

Figure 10: PN model for SFSPS with internal user

Analysis: This model has been analysed for system behaviour. The process cycle has been verified and found to be meeting the system requirements. Both external user and Internal user can interact with the system as required by the process. All the places in this net are safe and firing sequence is as per the requirements. There is no dead transition and all are live. The firing sequence for this model is as follows:

E1[ID:t1] ⟶ E2[ID:t2] ⟶ E3[ID:t3]

⟶ E4[ID:t4] ⟶ E1[ID:t1]

6.4.3 Single Function Multiple Processing Stages (SFMPS) including actions by internal users

There is a single function in the system. There are more than one stage of processing. The request obtained for this function is processed through various stages of processing and then response is delivered to customer. Possibly, some stages of processing require certain actions to be carried out by internal user(s) as per the process/procedure . The processes may or may not include activity as required.

Request ⟶ Process1 ⟶ ....

ProcessN ⟶ Response

Examples: Request for property registration, property bifurcation, etc.

The events and conditions of the system has been identified and mapped onto Petri Net model in the following way

Events

E1 - Request is generated
E2 - Process 1 processing starts
E3 - process 1 processing ends
E4 - Process 2 processing starts
E5 - Process 2 processing ends
E6 - Response sent for delivery

Conditions

A - Request arrived and waiting for processing
B - Process 1 is being processed
C - Outcome of process 1 generated
D – Process 2 is being processed
E - Response is generated
F - Process 1 is idle
G - Process 2 is idle
EU – External user idle
IU1 - Internal user1 idle
IU2 – Internal user2 idle

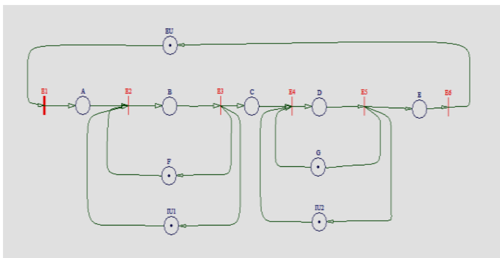

Figure 11: PN Model for SFMPS with internal users

Analysis: This model has been analysed for system behaviour. Both external user and Internal users can interact with the system as required by the process. All the places in this net are safe and firing sequence is as per the requirements. There is no dead transition and all are live. The matrix equation for this model is as shown below in the form incidence matrix.

	p1	p2	p3	p4	p5	p6	p8	p9	p10	p11
t1	-1	1	0	0	0	0	0	0	0	0
t2	0	-1	1	0	-1	-1	0	0	0	0
t3	0	0	-1	1	1	1	0	0	0	0
t4	0	0	0	-1	0	0	1	-1	-1	0
t5	0	0	0	0	0	0	-1	1	1	1
t6	1	0	0	0	0	0	0	0	0	-1

Table : Incidence matrix for SFMPS

6.4.4 Multi Functions Multiple Processing Stages (MFMPS) including actions by internal users

There are many functions in the system and each function could be broken down to one or more stages based on processing requirements. The request obtained for any function is processed through required number of stages of processing and then response is delivered to customer. Possibly, some stages of processing require certain actions to be carried out by internal user(s) as per the process/procedure. The processes may or may not include activity as required. It is a complex system.

Examples: eProcurement, Leave system, Advances, etc.

The events and conditions of the system has been identified and mapped onto Petri Net model as mentioned below:

Events

E19 – Request is generated
E1 – Request1 is selected
E2 – Process1 processing starts
E3 – Process1 processing ends
E4 – Process2 processing starts
E5 – process2 processing ends
E6 – Response is sent for delivery
E7 – Request2 is selected
E8 – Process3 processing starts
E9 – Process3 processing ends
E10 – Process4 processing starts
E11 – process4 processing ends
E12 – Response is sent for delivery
E13 – Request3 is selected
E14 – Process5 processing starts
E15 – Process5 processing ends
E16 – Process6 processing starts
E17 – process6 processing ends
E18 – Response is sent for delivery

Conditions

A - Request1 arrived and waiting for processing
B - Process 1 is being processed
C - Outcome of process 1 generated
D – Process 2 is being processed
E - Response is generated
H - Request2 arrived and waiting for processing
I - Process 3 is being processed
J - Outcome of process 3 generated
K – Process 4 is being processed
L - Response is generated
O - Request3 arrived and waiting for processing
P - Process 5 is being processed
Q - Outcome of process 5 generated
R – Process 6 is being processed
S - Response is generated
V - Request arrives and selected
F - Process 1 is idle
G - Process 2 is idle
M - Process 3 is idle
N - Process 4 is idle
T - Process 5 is idle
U - Process 6 is idle
EU – External user idle
IU1 - Internal user1 idle
IU2 - Internal user2 idle
IU3 - Internal user3 idle
IU4 - Internal user4 idle
IU5 - Internal user5 idle
IU6 - Internal user6 idle

Event	Preconditions	Post conditions
E19	EU	V
E1	V	A
E2	A,F,IU1	B
E3	B	C,F,IU1
E4	C, G,IU2	D
E5	D	E,G,IU2
E6	E	EU
E7	V	H
E8	H,M,IU3	I
E9	I	J,M,IU3
E10	J, M,IU3	K
E11	K	L,N,IU4
E12	L	EU
E13	V	O
E14	O,T,IU5	P
E15	P	Q,T,IU5
E16	Q,U,IU6	R
E17	R	S,U,IU6
E18	S	EU

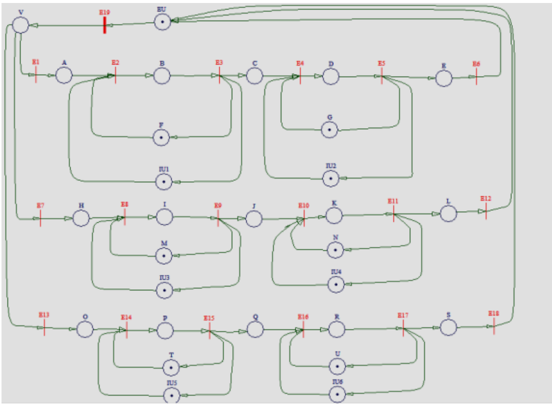

Figure 12: PN model for MFMPS with internal users

Analysis: This model has been analysed for system behaviour. The process cycle has been verified and found to be meeting the system requirements. Both external user and Internal users can interact with the system as required by the process. This model explains the net describing multiple choices (Three options) for processing based on the selection of request at place V. On selection, the process cycle is executed as per the flow. The cycle continues. All the places in this net are safe and firing sequence is as per the requirements. There is no dead transition and all are live. The firing sequence for this model is as follows:

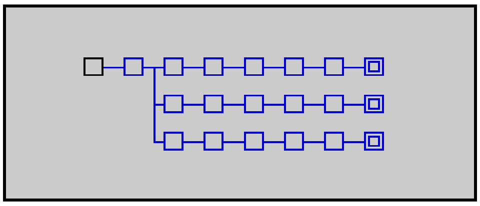

Figure 13 Reachability Tree for MFMPS

7. A PETRI NETS MODEL FOR A CITIZEN CENTRIC EGOVERNMENT SYSTEM

The model (Fig. 14) has been generated as case study to illustrate the effectiveness of Petri Nets model with Discrete Event driven system. This system is used to issue Trade licence for business users for a local city government. This system supports on-line application for issue of trade licence by business user (Place1). The application is scrutinized by the authorised department user (Place2) and it is forwarded to Inspector (department user-Place3) for verification and site inspection. The inspection report has been entered into the system and his recommendation. If the application is rejected, the applicant is informed about rejection. If the application is approved, the authorised department user generates the approved trade licence and issued to applicant. This process cycle can be visualized easily through this model. The PN model found to be very useful in understanding the process cycle, work flow and stages of processing.

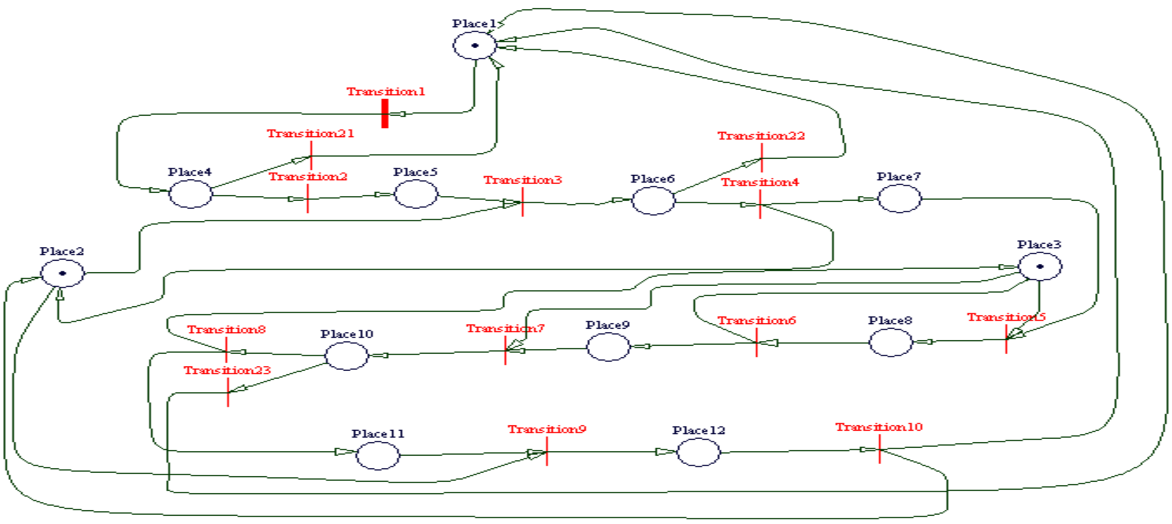

Figure 14: Petri Nets model for Citizen centric eGovernment system

8. CONCLUSIONS

Customer centric eGovernment system is very complex. The customer requests are growing rapidly and their expectations for quick responses are also increasing. The multiple requests have to be operated under different rules and procedures prior to delivery of responses or services to customer. This model can help to visualize the dynamic nature of Customer centric eGovernment system using discrete event driven system. The execution cycle of end to end processing can be modeled with the combination of events and conditions suitably in a predefined flow. The flow could be sequential or concurrent or combination of various net configurations as discussed in previous sections. It allows you to configure the required network topology for processing. The constructed model can be simulated to exhibit system behaviour for each case without construction. It can bring out non-firable transitions, if system is not represented properly for certain cases. This situations can be represented correctly and redo the simulations until all the transitions are firable and reachable from root to end leaf on various conditions.

The Petri Nets model can be used to configure concurrent and distributed system very effectively. The designers and system analysts can use this model for building necessary information system with the desired technology platforms. This model does not put any restriction on the technology to be used. It is technology independent model. Hence, this model cannot become obsolete, since it addresses characteristics and behaviour of the system.

The PN model has been used successfully for Citizen centric eGovernment systems and found to be very useful in capturing the process cycle and execution flow of the systems. It helps to understand how system works for every function by explicitly describing the path of execution. The model has also helped to identify paths, which are hanging or not closed properly. Also, change requests have been considered in the model to run feasibility studies prior to recommending for implementation.

9. ACKNOWLEDGEMENTS

This study was carried out as part of research work under computer science department, Dr. MGR Educational and research Institute, Chennai, India. The authors thank Chairman, ISRO for his continuous support and encouragement for this activity. The authors are also grateful to Scientific Secretary, ISRO for his guidance and support. The authors acknowledge the valuable support provided by ISRO teams and BMP teams for carrying out various eGovernance projects. Also, the valuable assistance and support provided by all the technical staff directly or indirectly are acknowledged.

10. REFERENCES

[1] Rachel Silcock (2001), What is e-government?, Hansard Society for parliamentary Government, Parliamentary Affairs, 54, 88-101.

[2] Alfred Tat-Kei Ho (2002), Reinventing Local Governments and the E-Government Initiative, Public Administration Review, July/August, Vol. 62, No.4, pp 434-444.

[3] Darrel West (2004), E-Government and the transformation of Service Delivery and Citizen Attitudes, Public Administration Review, January/February, Vol.64, No.1, pp 15-27.

[4] Jensen, K (1985) An Introduction to High-Level Petri Nets, Int. Symp. On Circuits and systems, proceedings, Kyoto, Japan, Vol. 2, New York, IEEE, pp. 723-726.

[5] Porter, M.E.,(1990), The Competitive Advantage of nations, London: The macmillan press Ltd.

[6] Christos G. Cassandras, Stéphane Lafortune (2008), 'Introduction to Discrete Event Systems', Second Edition, e-ISBN-13: 978-0-387-68612-7, Springer

[7] Andrea BOBBIO(1990), 'SYSTEM MODELLING WITH PETRI NETS', reprinted from: System reliability Assessment, Kluwer p.c., pp 102-143

[8] James L. Peterson(1981), ' PETRI NET THEORY AND THE MODELING OF SYSTEMS', Prentice-Hall, INC., Englewood Cliffs, N.J. 07632

[9] Tadao Murata (1989), Petri Nets: Properties, Analysis and Applications', Proceedings of the IEEE, Vol. 77, No.4, April, pp 541-580

[10] Ajmone Marsan, M., M.G.Balbo and G. Conte. (1986), 'Performance Models of Multiprocessor Systems', Massachusetts: The MIT Press.

[11] Merlin P and D.Farber (1976), recoverability of communication protocols – implication of a theoretical study, IEEE Transactions on Communications, pp. 1036-1043

[12] Venkatesh K, M.C. Zhou and R.Caudill (1994), 'Comparing ladder logic diagrams and Petri Nets for sequence controller design through a discrete manufacturing system', IEEE Trans. On Industrial Electronics 41(6), pp. 611-619

[13] Wang J (2006), 'Charging information collection modeling and analysis of GPRS networks', IEEE Trans. On Systems, Man and Cybernetics, Part C 36(6).

[14] Y. Narahari, K.Suryanarayan and N.V. Subba Reddy(1989), 'DISCRETE EVENT SIMULATION OF DISTRIBUTED SYSTEMS USING STOCHASTIC PETRI NETS', CH2766-4/89/0000-0622, IEEE, pp. 31.4.1-31.4.4

[15] Jiacun Wang, 'Petri Nets for Dynamic Event-Driven System Modeling', Department of Software Engineering, Monmouth University, West Long Branch, NJ 07764

[16] Wang Ge, Ren nan (2008), 'Modeling and research of the E-government system based on Petri net', International conference on Computer Science and Software Engineering, pp 1086-1089, DOI 10.1109/CSSE.2008.1469, IEEE Computer Society

[17] Vijay Gehlot, Carmen Nigro (2010), 'AN INTRODUCTION TO SYSTEMS MODELING AND SIMULATION WITH COLORED PETRI NETS', Proceedings of the 2010 Winter Simulation Conference, 978-1-4244-9864-2/10, IEEE

[18] Jose R. Celaya, Alan A. Desrochers and Robert J. Graves(2009), ‘Modeling and Analysis of Multi-agent Systems using Petri Nets’, Journal computers, vol.4, No.10, October

[19] Pan Xinglong, He Guo (2011), ‘Research on the Application of Discrete Event System based on Controlled Petri Net’, Third International Conference on Measuring Technology and Mechatronics Automation, DOI 10.1109/ICMTMA.2011.594, IEEE Computer Society

[20] Massimo Mecella, Francesco Parisi Presicce and Barbara Pernici (2002), ‘Modeling E-service Orchestration through Petri Nets’, LNCS 2444, pp. 38-47, Springer-Verlag, Berlin, Heidelberg, 2002

[21] Dragan Gasevic, ‘P3-Petri net tool’, www.sfu.ca/~dgasevic/projects/P3net/, the GoodOldAI research group.