Verification of ISO in Engineering and Education Using Matrix Structural Analysis and Petri Net

Dr.Abbas M. Al-Bakry 1*, Samaher Hussein Ali 2**

*Department of Computer Science, University of Babylon, Iraq

abbasmoh67@yahoo.com

**Department of Computer Science, University of Babylon, Iraq

Samaher_hussein@yahoo.com

Abstract: This paper presents a methodology for design models verify ISO conditions in engineering and Education. The first model uses matrix structural analysis to calculate the exact values design engineering and this model verify ISO conditions also it is far you estimation principles or ad-hoc principles but depend on the calculate the actual values. The second model uses Petri net have four layers sparsely connected networks. Petri net contain dynamic structures that evolve by growing and pruning of places, Transitions and connections. Petri net (also known as a place/transition net or P/T net [1]) is one of several mathematical representations of discrete distributed systems. As a modeling language, it graphically depicts the structure of a distributed system. In this paper, Petri net merge three levels represent (the conditions of building, staff, students) to design model of computer science department verify ISO while level number four represents the decision level. The models suggest in this paper recognize by capable of extended and generalizing from a small data set of large data vectors as well as from large data sets and small data vectors depend on uses data analysis principle.

Key words: Data analysis, ISO, Petri Net, Matrix Structural analysis.

1. INTRODUCTION

Data analysis is the process of looking at and summarizing data with the intent to extract useful information and develop conclusions[2]. Some people divide data analysis into Exploratory Data Analysis (EDA) and Confirmatory Data Analysis (CDA), where the EDA focuses on discovering new features in the data, and CDA on confirming or falsifying existing hypotheses. In this paper, focuses on CDA as effective tools of building this models .

Section 2 provides a brief review on Petri net. Section 3 characteristics that from the basis of ISO is identified. Section 4 explains the special structure analysis . Section 5 highlight a design of two different models with application to engineering and education. In section 6 concludes of this paper.

2. BASIC PETRI NETS

A Petri net consists of *places*, *transitions*, and *directed arcs*. Arcs run between places and transitions—not between places and places or transitions and transitions. The places from which an arc runs to a transition are called the input places of the transition; the places to which arcs run from a transition are called the output places of the transition[3].

Places may contain any number of tokens. A distribution of tokens over the places of a net is called a *marking*[4]. Transitions act on input tokens by a process known as *firing*. A transition is *enabled* if it can fire, i.e., there are tokens in every input place.

When a transition fires, it consumes the tokens from its input places, performs some processing task, and places a specified number of tokens into each of its output places. It does this atomically, i.e., in one non- interruptible step. Execution of Petri nets is nondeterministic. This means two things[5]:

a. multiple transitions can be enabled at the same time, any one of which can fire
b. none are *required* to fire — they fire at will, between time 0 and infinity, or not at all (i.e. it is totally possible that nothing fires at all).

Since firing is nondeterministic, Petri nets are well suited for modeling the concurrent behavior of distributed systems.

2.1. A formal definition [6][7]

A Petri net is a 5-tuple (S, T, F, M_0, W)

- *S* is a set of *places*.
- *T* is a set of *transitions*.
- *S* and *T* are disjoint, i.e. no object can be both a place and a transition.
- *F* is a set of arcs known as a *flow relation*.

The set F is subject to the constraint that no arc may connect two places or two transitions, or more formally: $F \subseteq (S \times T) \cup (T \times S)$

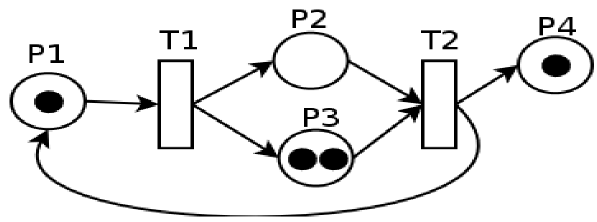
- $M_0 : S \longrightarrow \mathbb{N}$ is an *initial marking*, where for each place $s \in S$ there are $n_s \in \mathbb{N}$ tokens.
- $W : F \longrightarrow \mathbb{N}^+$ is a set of *arc weights*, which assigns to each arc $f \in F$ some $n \in \mathbb{N}^+$ denoting how many tokens are consumed from a place by a transition, or alternatively, how many tokens are produced by a transition and put into each place.

A variety of other formal definitions exist. Some definitions do not have arc weights, but they allow multiple arcs between the same place and transition, which is conceptually equal to one arc with a weight of more than one.

2.2. Basic Mathematical Properties

The state of a Petri net can be represented as an M vector, where the 1st value of the vector is the number of tokens in the 1st place of the net, the 2nd is the number of tokens in the 2nd place, and so on. Such a representation fully describes the state of a Petri net.

A state-transitionlist, $\vec{\sigma} = \langle M_{i_0} t_{i_1} M_{i_1} \ldots t_{i_n} M_{i_n} \rangle$, which can be shortened to simply $\vec{\sigma} = \langle t_{i_1} \ldots t_{i_n} \rangle$ is called a *firing sequence* if each and every transition satisfies the firing criteria (i.e. there are enough tokens in the input for every transition). In this case, the state-transition list of $\langle M_{i_0} M_{i_1} \ldots M_{i_n} \rangle$ is called a *trajectory*, and M_{i_n} is called *reachable* from M_{i_0} through the firing sequence of $\vec{\sigma}$. Mathematically written: $M_{i_0}[\vec{\sigma} > M_{i_n}$. The set of all firing sequences that can be reached from a net N and an initial marking M_0 are noted as $L(N,M_0)$. The state-transition matrix W^- is | S | by | T | large, and represents the number of tokens taken by each transition from each place. Similarly, W^+ represents the number of tokens given by each transition to each place. The sum of the two, $W = W^+ - W^-$ can be used for calculating the above mentioned equation of $M_{i_0}[\vec{\sigma} > M_{i_n}$ which can now be simply written as $M_0 - M_n = W^T \cdot \sigma$, where σ is a vector of how many times each transition fired in the sequence. Note that just because the equation can be satisfied, does not mean that it can actually be carried out - for that there should be enough tokens for each transition to fire, i.e. the satisfiability of the equation is required but not sufficient to say that state M_n can be reached from state M_0[8]. Example:-



Fig(1): Petri Net Example

$$W^+ = \begin{bmatrix} * & t1 & t2 \\ p1 & 0 & 1 \\ p2 & 1 & 0 \\ p3 & 1 & 0 \\ p4 & 0 & 1 \end{bmatrix}$$

$$W^- = \begin{bmatrix} * & t1 & t2 \\ p1 & 1 & 0 \\ p2 & 0 & 1 \\ p3 & 0 & 1 \\ p4 & 0 & 0 \end{bmatrix}$$

$$W = \begin{bmatrix} * & t1 & t2 \\ p1 & -1 & 1 \\ p2 & 1 & -1 \\ p3 & 1 & -1 \\ p4 & 0 & 1 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 & 0 & 2 & 1 \end{bmatrix}$$

2.3. Petri Net Types

There are many extensions to Petri nets. Some of them are completely backwards-compatible (e.g. coloured Petri nets) with the original Petri net, some add properties that cannot be modelled in the original Petri net (e.g. timed Petri nets). If they can be modelled in the original Petri net, they are not real extensions, instead, they are convenient ways of showing the same thing, and can be transformed with mathematical formulas back to the original Petri net, without losing any meaning. Extensions that cannot be transformed are sometimes very powerful, but usually lack the full range of mathematical tools available to analyse normal Petri nets[9].

The term high-level Petri net is used for many Petri net formalisms that extend the basic P/T net formalism. This includes coloured Petri nets, hierarchical Petri nets, and all other extensions sketched in this section.

A short list of possible extensions:

- In a standard Petri net, tokens are indistinguishable. In a Coloured Petri net, every token has a value. In popular tools for coloured Petri nets such as CPN Tools, the values of tokens are typed, and can be tested and manipulated with a functional programming language. A subsidiary of coloured Petri nets are the well-formed Petri nets, where the arc and guard expressions are restricted to make it easier to analyse the net[10].

- Another popular extension of Petri nets is hierarchy: Hierarchy in the form of different views supporting levels of refinement and abstraction were studied by Fehling. Another form of hierarchy is found in so-called object Petri nets or object systems where a Petri net can contain Petri nets as its tokens inducing a hierarchy of nested Petri nets that communicate by synchronisation of transitions on different levels[11].
- A Vector Addition System with States (VASS) can be seen as a generalisation of a Petri net. Consider a finite state automaton where each transition is labelled by a transition from the Petri net. The Petri net is then synchronised with the finite state automaton, i.e., a transition in the automaton is taken at the same time as the corresponding transition in the Petri net. It is only possible to take a transition in the automaton if the corresponding transition in the Petri net is enabled, and it is only possible to fire a transition in the Petri net if there is a transition from the current state in the automaton labelled by it. (The definition of VASS is usually formulated slightly differently.)
- Prioritised Petri nets add priorities to transitions, whereby a transition cannot fire, if a higher-priority transition is enabled (i.e. can fire). Thus, transitions are in priority groups, and e.g. priority group 3 can only fire if all transitions are disabled in groups 1 and 2. Within a priority group, firing is still non-deterministic.
- The non-deterministic property has been a very valuable one, as it lets the user abstract a large number of properties (depending on what the net is used for). In certain cases, however, the need arises to also model the timing, not only the structure of a model. For these cases, timed Petri nets have evolved, where there are transitions that are timed, and possibly transitions which are not timed (if there are, transitions that are not timed have a higher priority than timed ones). A subsidiary of timed Petri nets are the stochastic Petri nets that add nondeterministic time through adjustable randomness of the transitions. The exponential random distribution is usually used to 'time' these nets. In this case, the nets' reachability graph can be used as a Markov chain.

There are many more extensions to Petri nets, however, it is important to keep in mind, that as the complexity of the net increases in terms of extended properties, the harder it is to use standard tools to evaluate certain properties of the net. For this reason, it is a good idea to use the most simple net type possible for a given modeling task[11].

3. International Organization for Standardization (ISO)

ISO is the world's largest developer and publisher of International Standards. ISO is a network of the national standards institutes of 157 countries, one member per country, with a Central Secretariat in Geneva, Switzerland, that coordinates the system.

ISO is a non-governmental organization that forms a bridge between the public and private sectors. On the one hand, many of its member institutes are part of the governmental structure of their countries, or are mandated by their government. On the other hand, other members have their roots uniquely in the private sector, having been set up by national partnerships of industry associations.
Therefore, ISO enables a consensus to be reached on solutions that meet both the requirements of business and the broader needs of society.

4. Matrix Structure Analysis

One of the responsibilities of the structural design engineer is to devise arrangements and proportions of members that can withstand, economically and efficiently, the conditions anticipated during the lifetime of a structure. A central aspect of this function is the calculation of the distribution of forces within the structure and the displaced state of the system. Our objective is to describe modern methods for performing these calculations in the particular case of framed structures. The number of structures that are actually simple frameworks represents only a part of these whose idealization in the form of a framework is acceptable for the purposes of analysis. Building of various types, portions of aerospace and ship structures, and radio telescopes and the like can often be idealized as framework[12].

In design, both serviceability limit states and strength limit states should be considered.

Structures consisting of two-or three-dimensional components-plates, membranes, shells, solids are more complicated in that rarely do exact solutions exist for the applicable partial differential equations. One approach to obtaining practical, numerical solutions is the **finite element method**. The basic concept of the method is that the total structure can be modeled analytically by its. Subdivision into regions(the finite elements) in each of which the behavior by a set of assumed functions representing the stresses or displacements in that region. This permits the problem formulations to be altered to one of the establishment of a system of algebraic equations.

Viewed in this way, structural analysis may be broken down into five parts[13].

- Basic mechanics: the fundamental relationships of stress and strain, compatibility and equilibrium.

- Finite element mechanics: the exact or approximate solution of the differential equations of the element.
- Equation formulation: the establishment of the governing algebraic equations of the system.
- Equation solution: computational methods and algorithms.
- Solution interpretation: the presentation of results in a form useful in design.

This paper deals chiefly with part 3,4 and 5, it is on matrix structural analysis. This is approach to these parts that currently seems to be most suitable for automation of the equation formulation process and for taking advantage of the powerful capabilities of the computer in solving large order systems of equations.

The equations of the matrix (or finite) element approach are of a form so generally applicable that it is possible in the theory to write a single computer program that will solve an almost limitless variety of problems in structural mechanics. Many commercially available general purpose programs attempt to obtain this objective, although usually on a restricted scale. The advantage of general purpose programs is not merely this capability but the unity afforded in the instruction of prospective users, input and output data interpretation procedures, and documentation[14].

The four components in the flowchart of figure 2 are common to virtually all general purpose, finite element analysis programs.

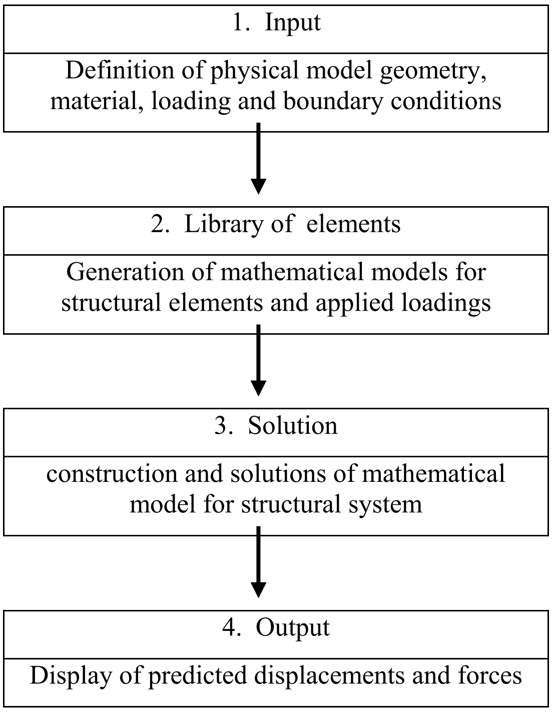


Fig.(2): Structural analysis computational flow

5. The Proposed Models

5.1 Engineering Models

This paper deals with some of the methods for training problems of large size. One way to handle larger structures idealization, that is to disregard, suppress, or approximate the effect of degrees of freedom that in the opinion of the analyst, have only a minor bearing on the result. The many different ways in which this may be done are so dependent upon the individual structure that they cannot be discussed usefully in a general text. Here we presents scheme for reducing the order of the system of equations that have be solved at any one time once the structure has been idealized. This means that generally we will discussing method for reducing the order of the stiffness matrices to be inverted. The flowchart of this model explains in figure 3.

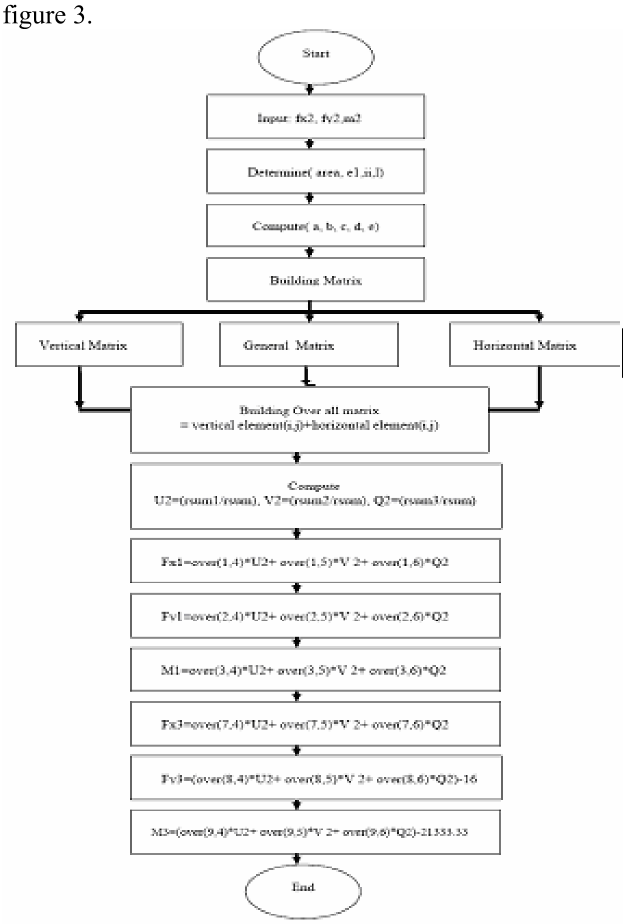


Fig.(3): Flowchart of the Engineering Models

Experiment :-The engineering model uses matrix structural analysis to calculate the exact values and this model verify ISO conditions also it is far you estimation principles or ad-hoc principles but depend on the calculate the actual values. Before this, we need to determine some of parameters relate to This model such as (u1 = 0, v1 = 0, Q1 = 0, u3 = 0, v3 = 0, Q3 = 0, fx2 = 50, fy2 = -16, m2 = 21333.33, area = 6000, e1 = 200, l=8000.). And by experiment we find U2= 0.344, V2= 0.1187, Q2= 0.0005, Fx1= -2.1975, Fy1= -17.805, M1= 6290, Fx3= -51.6, Fy3= -18, M3= -15888.205.

Computation:

ii = 200 * (10 ^ 6) (1)

$$a = \frac{(area * e1)}{(l)} \quad (2)$$

$$b = \frac{(12 * e1 * ii)}{(l ^ 3)} \quad (3)$$

$$c = \frac{(6 * e1 * ii)}{(l^2)} \quad (4)$$

$$d = \frac{(4 * e1 * ii)}{(l)} \quad (5)$$

e = (1 * 10 ^ 7) (6)

Building Matrix:

$$\text{VerticalMatrix} = \begin{bmatrix} b & 0 & -c & -b & 0 & -c \\ 0 & a & 0 & 0 & -a & 0 \\ -c & 0 & d & c & 0 & e \\ -b & 0 & c & b & 0 & -c \\ 0 & -a & 0 & 0 & a & 0 \\ c & 0 & e & c & 0 & d \end{bmatrix}$$

$$\text{Horizontal Matrix} = \begin{bmatrix} a & 0 & 0 & -a & 0 & 0 \\ 0 & b & c & 0 & -b & c \\ 0 & c & d & 0 & -c & e \\ -a & 0 & 0 & a & 0 & 0 \\ 0 & -b & -c & 0 & b & -c \\ 0 & c & e & 0 & -c & d \end{bmatrix}$$

General Matrix=

$$\begin{bmatrix} a\cos^2 + b\sin^2 & a\cos\sin - b\sin\cos & -c\sin & -a\cos^2 + b\sin^2 & -a\cos\sin - b\sin\cos & -c\sin \\ a\cos\sin - b\sin\cos & a\sin^2 + b\cos^2 & c\cos & -a\cos\sin - b\sin\cos & -a\sin^2 + b\cos^2 & c\cos \\ -c\sin & c\cos & d & c\sin & -c\cos & e \\ -a\cos^2 + b\sin^2 & -a\cos\sin - b\sin\cos & c\sin & a\cos^2 + b\sin^2 & a\cos\sin - b\sin\cos & -c\sin \\ -a\cos\sin - b\sin\cos & -a\sin^2 + b\cos^2 & -c\cos & a\cos\sin - b\sin\cos & a\cos^2 + b\sin^2 & -c\cos \\ -c\sin & c\cos & e & -c\sin & -c\cos & d \end{bmatrix}$$

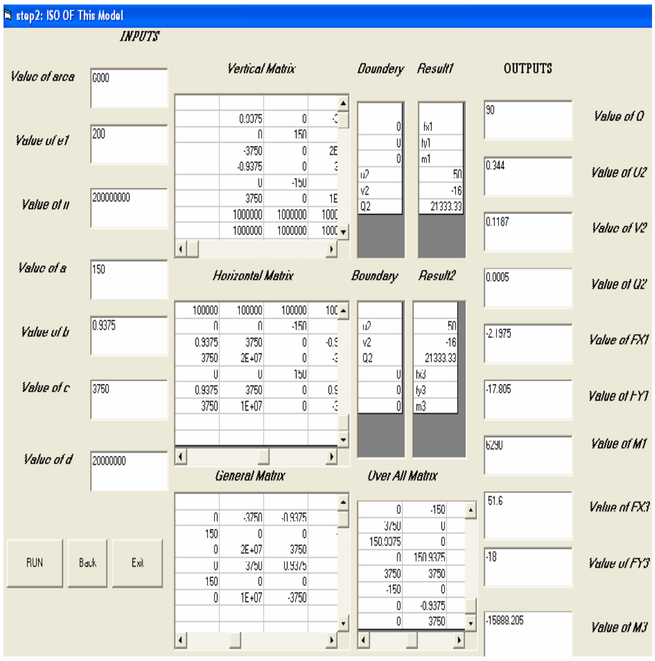


Fig.(4): Interface and result of the Engineering Model

$$\begin{bmatrix} fx1 \\ fy1 \\ m1 \\ fx2 \\ fy2 \\ m2 \\ fx3 \\ fy3 \\ m3 \end{bmatrix} = [overallmat\ rix]* \begin{bmatrix} U1 \\ V1 \\ Q1 \\ U2 \\ V2 \\ Q2 \\ U3 \\ V3 \\ Q3 \end{bmatrix} \quad (7)$$

From this result, we can see the accurate and fast of the proposed model and also we can use this model with different angles as explained in figure 4.

5.2 Education Models

This model deals with some of the properties for Petri net to design dynamic structure of computer science department verify the conditions of ISO. This problem discuss from many respects include building, staff, students. Figure 5 explains Petri net of this model. **The first level** (layer) of net represents the building and in this level test nine conditions includes(number of classrooms, number of Adaptive, number of stoles, number of fan, number of Labs, number of computers in each Labs, number of stoles in these Labs, number of Blackboard and number of offices). The Firing Process is Performed when all these conditions are verification. Each place in that level have three tokens while each to represents one of the condition . **The second level** of net represents the staff and in this level test eight conditions includes (number of teacher has title is professor, Ass. Professor, Lecture, Ass Lecture, Programmer, Ass programmer, number of artistcal and number of services handlers). As the above level, the Firing Process is Performed when all these conditions are verification also Each place in that level have three tokens, each to represents one of the condition while one of these token represents the verification of all condition in level number one . **The third level** of net represents the students and in this level test three conditions includes(number student in each class, number of stages, and number of classes in stage). When condition is verification, put token (has red color) in the place that represent that condition while, when the firing process is performed (all condition at that level is verify) the tokens are hidden from all places in that level and transform to new place in the second level. Figure 5 shows the Petri net of the education model while all the following figures show how you can deal with this model.

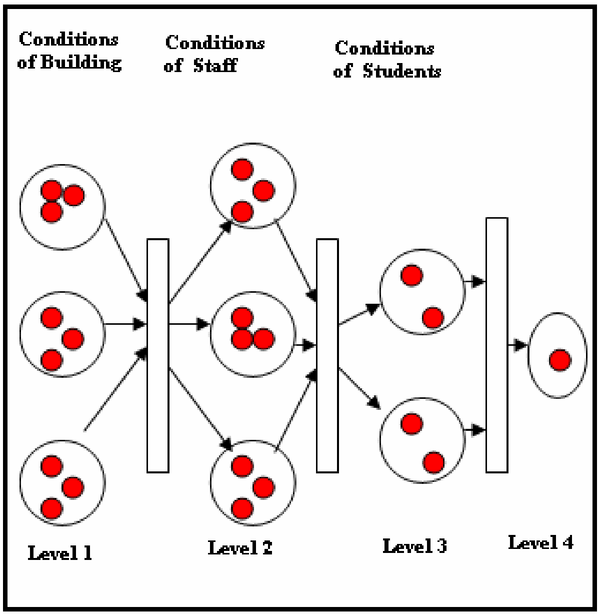


Fig.(5): Petri net of the Education Model

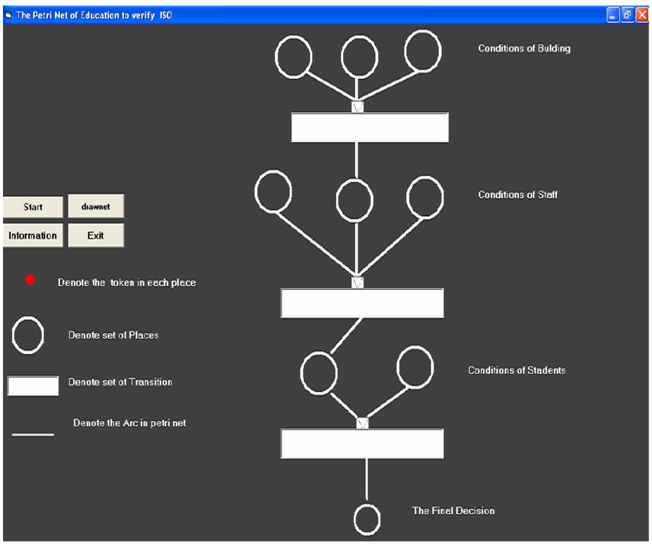


Fig.(6): Interface and Petri net of the Education Model

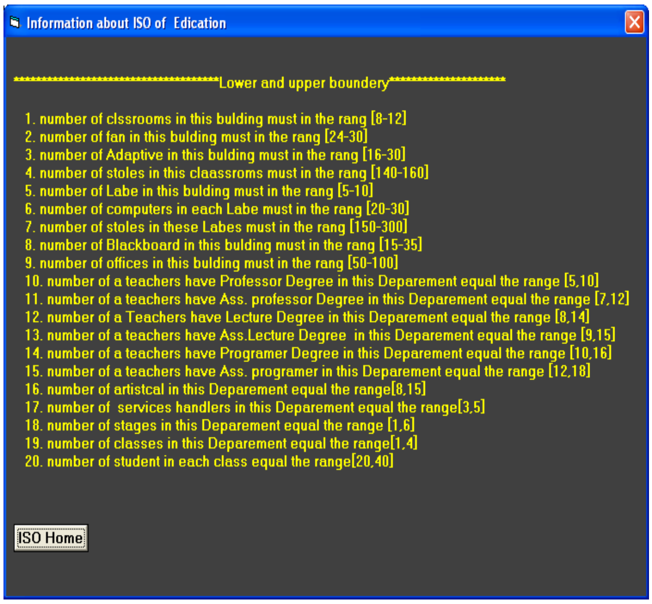


Fig.(7): Information About ISO of Education

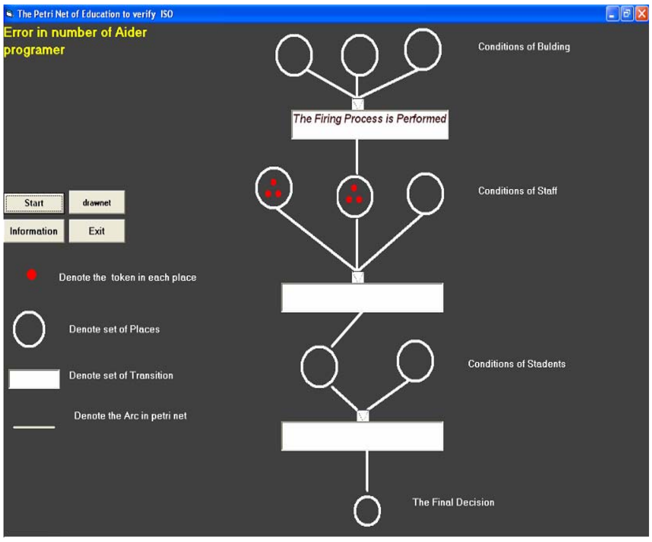


Fig.(8): Interface when occur error in level no two

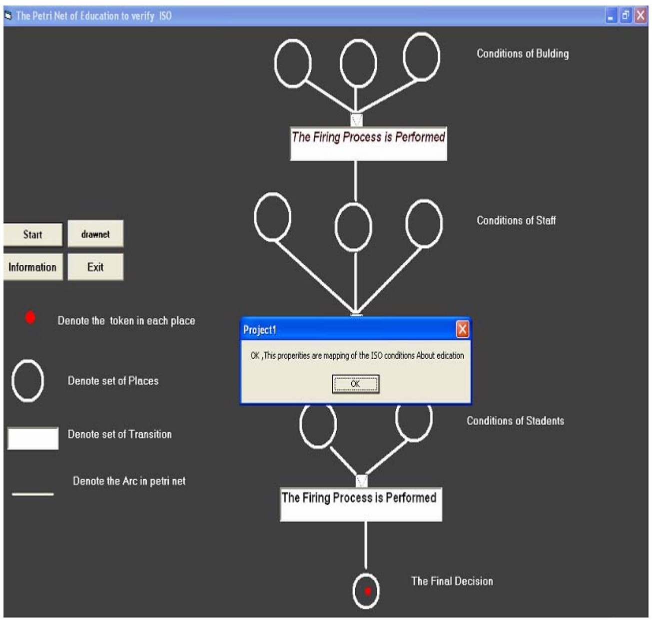


Fig.(9): Interface when the model successful

6. Conclusion

The objective of this paper was to use Matrix Structural Analysis technique and Petri net in engineering and education fields to verify ISO conditions. As the main contribution, we introduced matrix structure analysis prove fast and accrue design model in engineering field. Where This paper deals:-

- Equation formulation: the establishment of the governing algebraic equations of the system.
- Equation solution: computational methods and algorithms.
- Solution interpretation: the presentation of results in a form useful in design.

it is on matrix structural analysis. This is approach to these parts that currently seems to be most suitable for automation of the equation formulation process and for taking advantage of the powerful capabilities of the computer in solving large order systems of equations.

Also, this paper prove Petri net is more suitable tool to dynamic structure and verify them from applying it in the Education models where we uses four levels of net, each level represents the condition of one of the main standardization condition to design education models. As a result, we can say the design models (engineering and education) is clearly understandable and have a good generality.

REFERENCES

[1] Cardoso, Janette; Heloisa Camargo. *Fuzziness in Petri Nets*. Physica-Verlag. ISBN 3-7908-1158-0.

[2] "http://en.wikipedia.org/wiki/Data_analysis".

[3] Котов, Вадим (1984). *Сети Петри (Petri Nets, in Russian)*. Наука, Москва.

[4] Pataricza, András (2004). Formális moódszerek az informatikában (Formal methods in informatics). TYPOTEX Kiadó. *ISBN 963-9548-08-1*.

[5] Peterson, James L. (1977). "Petri Nets". *ACM Computing Surveys* **9** (3): 223–252.

[6] Peterson, James Lyle. *Petri Net Theory and the Modeling of Systems*. Prentice Hall. ISBN 0-13-661983-5.

[7] Reisig, Wolfgang. *A Primer in Petri Net Design*. Springer-Verlag. ISBN 3-540-52044-9.

[8] Riemann, Robert-Christoph. *Modelling of Concurrent Systems: Structural and Semantical Methods in the High Level Petri Net Calculus*. Herbert Utz Verlag. ISBN 3-89675-629-X.

[9] Störrle, Harald. *Models of Software Architecture - Design and Analysis with UML and Petri-Nets*. Books on Demand. ISBN 3-8311-1330-0.

[10] Zhou, Mengchu; Frank Dicesare. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers. ISBN 0-7923-9289-2.

[11] Zhou, Mengchu; Kurapati Venkatesh. *Modeling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach*. World Scientific Publishing. ISBN 981-02-3029-X.

[12] William, Richard (2000). "Matrix Structure Analysis", book, New York.

[13] H.C. Martin(2002). "Introduction to Matrix Method of Structural Analysis"

[14] C.K. Wang (1998)." Intermediate Structure Analysis". Book.