Model's quality assurance using MSA and Petri Net

Abbas M. Al-Bakry
College of Computer Technology, University of Babylon, Iraq

Abstract:

This work presents a methodology for designing models in engineering and education fields for verifying the quality assurance conditions. The first model uses matrix structural analysis (MSA) to calculate the expected values for the model. The matrix structural analysis (MSA) uses to design model in engineering field, this helps to expect the values which is the nearest to the real ones. The MSA verify the quality assurance conditions and estimate principles or ad-hoc principles depending on the calculating the actual values. Petri nets uses in another example (Education field). Petri Nets provide a very important feature which is the dynamic property, so the model designing with Petri nets becomes look like as in real.

Key words: Data analysis, Quality management, Petri Net, Matrix Structural analysis.

الخلاصة

يعرض البحث طريقة لتصميم نماذج في حقلي الهندسة والتعليم لاختيار شروط تأكيد ألجوده. تم استخدام تحليل هيكل المصفوفة في النموذج الأول لغرض احتساب القيم المتوقعة للنموذج , إن استخدام تحليل هيكل المصفوفة لنموذج هندسي لغرض تأكيد جوده الموديل سهل في تخمين القيم الحقيقية والمفترض استخدامها واقعيا. من الناحية الأخرى تم استخدام مخططات ألبتري في الحقل التعليمي لنفس المهمة وهي تأكيد جودة الموديل وان من ابرز صفات مخططات ألبتري هي الصفة الديناميكية والتي تشعرك بأن النموذج يعمل وكأنه في الواقع.

Introduction

Data analysis is the process of looking at and summarizing data with the intent to extract useful information and develop conclusions[WIKI-EN]. Data analysis can be divided 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 1 provides a brief review on Petri net. section 2 explains the special structure analysis . Section 3 highlight a design of two different models with application to engineering and education. In section 4 discussion and concludes the paper.

1. **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[KOTO84].

Places may contain any number of tokens. A distribution of tokens over the places of a net is called a *marking*[PATA04]. 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 [PETE77]:

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.

1.1. A formal definition [PETE77[REIS92]

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

- *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:

$$F \subseteq (S \times T) \cup (T \times S)$$

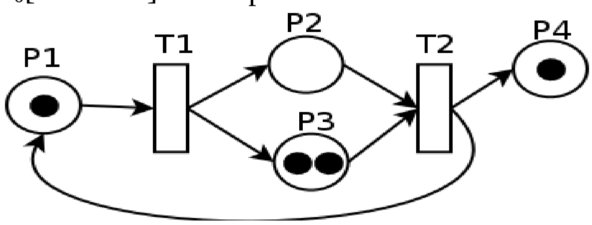
- $M_0 : S \longrightarrow \mathbb{N}$ is an *initial marking*, where for each place ,$s \in S$ there are $n_s \in \mathbb{N}$ tokens.
- $W : F \longrightarrow \mathbb{N}^+$ is a set of *arc weights*, which assigns to each arc $f \in F$ some $n \in \mathbb{N}^+$ denoting how many tokens are consumed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

1.2. 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, $\vec{\sigma} = \langle M_{i_0} t_{i_1} M_{i_1} \dots t_{i_n} M_{i_n} \rangle$, which can be shortened to simply $\vec{\sigma} = \langle t_{i_1} \dots t_{i_n} \rangle$ 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 $\langle M_{i_0} M_{i_1} \dots M_{i_n} \rangle$ 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[RIEM99]. Example:-



Fig(1): Petri Net Example

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

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

1.3. 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[STOR00].

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[ZHOU09].
- 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[ZHOU98].
- 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[ZHOU98]**.**

2. Matrix Structure Analysis(Msa).

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 [WILL00].

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 [MART02].

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 [JOHN06].

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

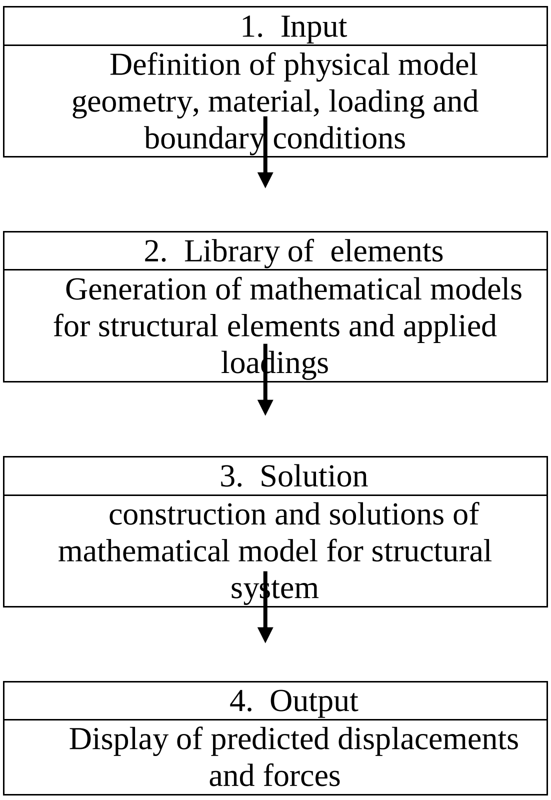


Fig. (2): Structural analysis computation flow

3. The Proposed Models

3.1 The First Example in engineering field

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.

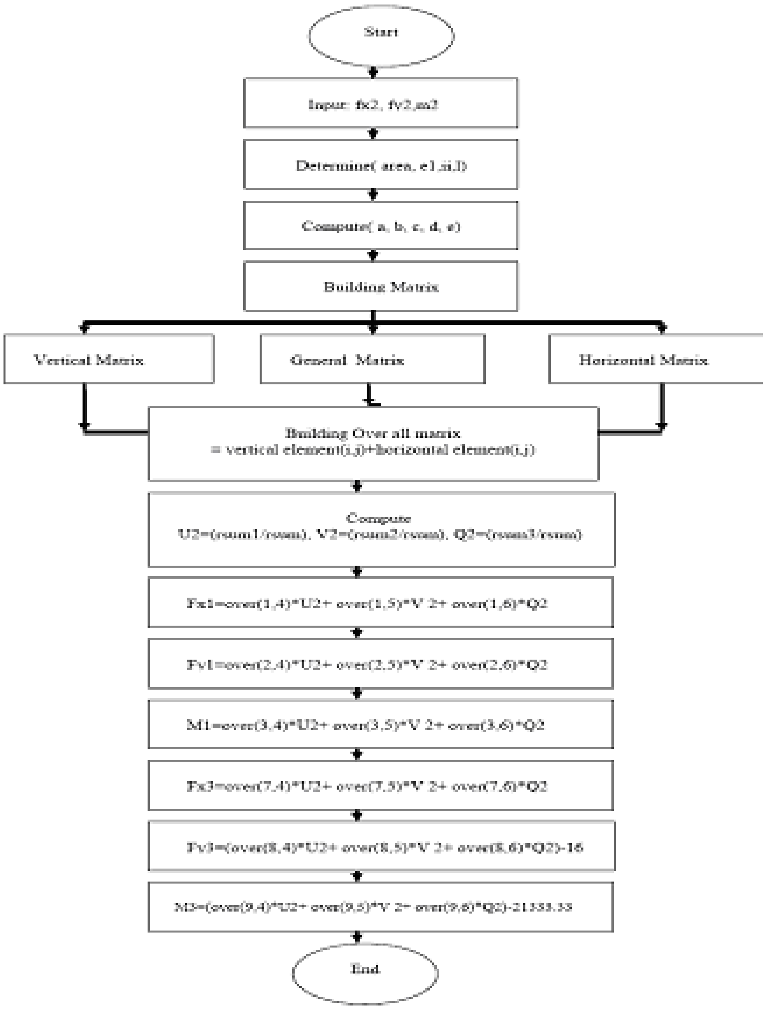


Fig.(3): Flowchart of the Engineering Models

If we beginning with the following values:
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.

The Computation:

$$ii = 200 * (10 \wedge 6) \qquad \text{------------} (1)$$

$$a = \frac{(area * e1)}{(l)} \qquad \text{-------------}(2)$$

$$b = \frac{(12 * e1 * ii)}{(l \wedge 3)} \qquad \text{------------} \ (3)$$

$$c = \frac{(6 * e1 * ii)}{(l \wedge 2)} \qquad \text{-------------}(4)$$

$$d = \frac{(4 * e1 * ii)}{(l)} \qquad \text{-------------}(5)$$

$$e = (1 * 10 \wedge 7) \qquad \text{-------------}(6)$$

The Matrix can be builds as follow:

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

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

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

Example:- Find Fx1,Fy1,M1,Fx3,Fy3,M3. Depend on the above information**.**

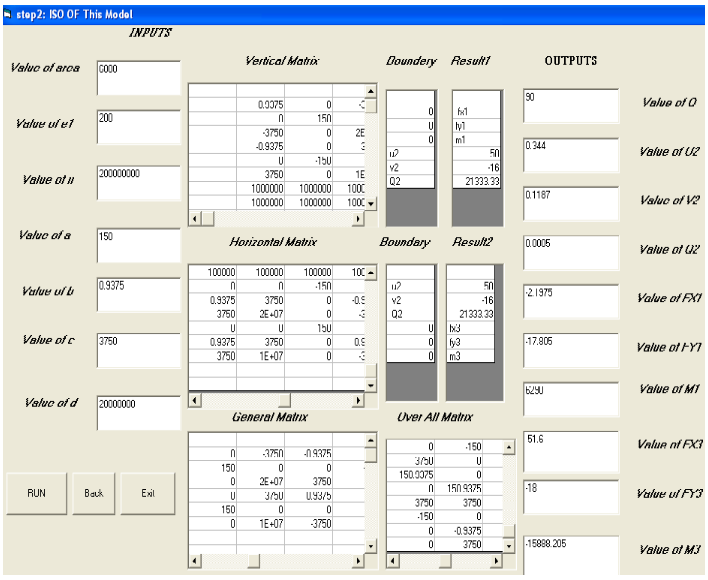


Fig.(4): The resulted values of the Engineering Model

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

By Appling this model, 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. From this result, we can see the accurate and fast of the proposed model and also we can use this model with different angles.

4.2 The Second Example in Education field

This model deals with quality assurance in one department of higher education, so we need to check the interesting thinks such that(Staff availability, classroom suitability, student classifications, labs. suitability, ...ect.).

When we start to design the model we need to check the quality assurance requirements for this department, if this requirements available which can be represented by conditions (C1,C2,..Cn). If all conditions achieved the transition *requirement check* fire and token in requirement place transfer to matched requirements place.

At the second model stage all achieved conditions can matched with the actual limitations if it match the *matched* transition fire and the token transfer to the environment requirements place. If it is not matched we need to refine feed back.

The matched achieved when the *matched* transition fire and the token transferred to Accepted Quality place and the process finished. When matched not achieved we need a refine feed back.

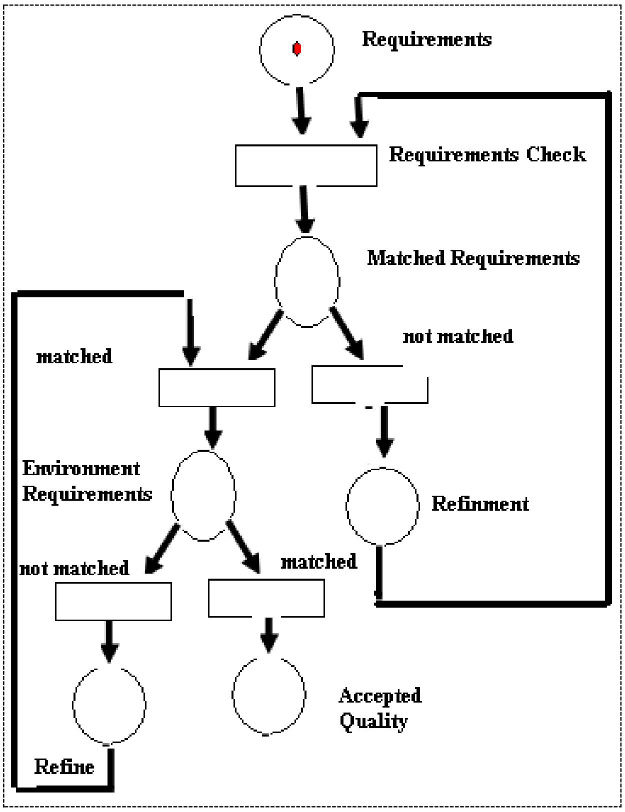


Fig.(5): Petri net of the Education Model

5. Conclusions

The main benefits of use Matrix Structural Analysis (MSA) technique and Petri net for modeling problems is to get the best model quality based on quality assurance conditions. As the main contribution, we introduced MSA to improve the designed model to get fast and accrue results:

- 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.

In MSA approach the parts currently seems to be most suitable for automating the equation formulizing process and for taking advantage of the powerful capabilities of the computer to solve a large order of systems of equations.

Petri nets provide a dynamic structure for the model this facility make the resulted model like in real world. Petri net used to test the quality assurance in education field example. As a result, using MSA and Petri Net for designing models in (engineering and education examples) gives a clear, understandable, and very good result which is very nearest to the real.

References

CARD99, Cardoso, Janette; Heloisa Camargo (1999). *Fuzziness in Petri Nets*. Physica-Verlag. ISBN 3-7908-1158-0,

JOHN06, John W. van (2006)." Intermediate Structure Analysis". PhD thesis.

KOTO84, Котов, Вадим (1984). *Сети Петри (Petri Nets, in Russian)*. Наука, Москва.

MART02, H.C. Martin(2002). "Introduction to Matrix Method of Structural Analysis", 2nd Ed., Wiley McGraw-Hill.

PATA04, Pataricza, András (2004). Formális moódszerek az informatikában (Formal methods in informatics). TYPOTEX Kiadó. *ISBN 963-9548-08-1*.

PETE05, Peterson, James Lyle (2005). *Petri Net Theory and the Modeling of Systems*. Prentice Hall. ISBN 0-13-661983-5.

PETE77, Peterson, James L (1977). "Petri Nets". *ACM Computing Surveys* **9** (3): 223–252.

REIS92, Reisig, Wolfgang (1992). *A Primer in Petri Net Design*. Springer-Verlag. ISBN 3-540-52044-9.

RIEM99, Riemann, Robert-Christoph (1999). *Modelling of Concurrent Systems: Structural and Semantical Methods in the High Level Petri Net Calculus*. Herbert Utz Verlag. ISBN 3-89675-629-X.

STOR00, Störrle, Harald (2000). *Models of Software Architecture - Design and Analysis with UML and Petri-Nets*. Books on Demand. ISBN 3-8311-1330-0.

WIKI-EN, "http://en.wikipedia.org/wiki/Data_analysis",Using Multivariate Statistics, Fifth Edition. Boston: Pearson Education, Inc.

WILL00, William, Richard (2000). "Matrix Structure Analysis", With MASTAN2, 2nd Edition. by **William** McGuire.

ZHOU09, Zhou, Mengchu; Frank Dicesare (2009). *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers. ISBN 0-7923-9289-2.

ZHOU98, Zhou, Mengchu; Kurapati Venkatesh (1998). *Modeling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach*. World Scientific Publishing. ISBN 981-02-3029-X.