Cooperative Navigation in Multimedia Systems

Maxime Wack, Nathanael Cottin, Rachid Bouyekhf

maxime.wack@utbm.fr
University of Technology Belfort-Montbeliard (UTBM)
SeT Laboratory
90010 Belfort Cedex, France

Abstract. The emergence of the New Technologies of Information and Communication (NTIC), and the development of new tools open some perspectives for multimedia application design. In this paper we propose a graphical model of cooperative navigation of the multimedia applications. The model is based on the distinction between public and private areas. We use Petri nets to model several patterns which allow to build a complete navigation process. An example is worked out to illustrate the proposed approach.

1 Introduction

Multimedia applications, e.g. video conferencing and video-on-demand, are of great importance in industry, academic research and standardization. Large scale deployment of multimedia applications will impose very high navigation constraints on the overall system components. Multimedia applications requirements range from high data rates, due to the voluminous nature of multimedia data, to severe temporal constraints, due to the continuous nature of requests.

Multimedia application designers usually do not make use of navigation rules which allow them to build well-configured multimedia applications. This fact may lead to design complex navigation models and therefore inadapted structures from users point of view. Hence, several authors have investigated this problem by means of different approaches [1,2,3]. Most of them focus on individual navigation, whereas few works deal with cooperative navigation, which is our mean concern.

A central issue for the design of multimedia application is the definition of the synchronizations points when presenting the different components of the application. The capacity of defining this structure implies the availability of an adequate synchronization model. Therefore, powerful techniques are needed for rigorously analyzing the possible kinds of behavior of the proposed Multimedia models to assure that they exhibit all and at least the best properties intended. Especially for these synchronization problems, Petri nets offer a suitable mathematical background with an already powerful theory which is still under development.

In order to be applicable to the design of large and complex multimedia applications, the model should be able to satisfy some important requirements:

1. the capacity of describing hierarchical document structures, as well as hierarchical levels of synchronization
2. the capacity of representing user interactions based synchronizations
3. the availability of verification techniques for detecting potential inconsistencies in large multimedia documents
4. and finally, the simplicity and intuitive nature of the modeling concepts provided to the users.

An important property of the Petri net approach is its extreme generality. It helps developers in a general way in reasoning about the behavior of Multimedia systems. Because of its generality, the Petri net framework can be used with Multimedia systems expressed in a wide variety of specification.

The objectives of this research are twofold: To introduce the concepts of the public and private areas within multimedia applications; to use Petri net for modeling the interactions between the users and such areas.

2 Cooperative navigation

Cooperative navigation provides tools and mechanisms to help the user during his navigation. We define *public* and *private* areas depending on the user's degree of liberty when he wants to access resources :

- A *public* area is a set of non-constrained resources which may be accessed concurrently by many users
- A *private* area defines rules to access shared and critical resources. The navigation is constrained to distribute these resources among users. This area type may be extended to describe *meeting* areas.

This classification is based on a graph representation of web sites where :

- Nodes represent local resources, which can either be static or dynamically generated during the navigation process. Similarly to navigation areas, we distinguish *public* and *private* nodes
- Links between resources are assimilated to edges. There may be *synchronous* or *asynchronous* links. Synchronous links are used by private areas to manipulate groups of users whereas asynchronous links refer to public areas.

3 Basic Petri nets definitions

In this section, we recall the basic Petri net terminology and notation mostly taken from [4]. A Petri net is a five-tuple $N = (P, T, I, O, M_0)$ where

- $P = \{p_1, p_2, \cdots, p_m\}$ is a finite set of places (represented with circles),
- $T = \{t_1, t_2, \cdots, t_n\}$ is a finite set of transitions (represented with line segments),
- $I : P \times T \rightarrow \mathbb{N}$ is an input function such that $I(p_i, t_j)$ is the weight of the arc directed from place p_i to transition t_j,
- $O : P \times T \rightarrow \mathbb{N}$ is an output function such that $O(p_i, t_j)$ is the weight of the arc directed from transition t_j to place p_i,
- M_0 is an initial marking that associates zero or more tokens to each place.

Furthermore, $n \geq 0$, $m \geq 0$, $n + m \geq 1$ and $P \cap T = \varnothing$.

The state of a Petri net is defined by the number of tokens in each place and is represented by a vector $M = [M(p_1), ..., M(p_m)]^t$, called the marking vector of the Petri net, where $M(p_i)$ is the number of tokens in place p_i. A transition $t_j \in T$ is said to be enabled if and only if $M(p_i) \leq I(p_i, t_j), \forall p_i \in P$. An enabled transition may fire. When transition t_j fires, $I(p_i, t_j)$ tokens are remouved from each input place p_i of t_j and $O(p_i, t_j)$ tokens are added to each output place p_i of t_j.

The dynamic behavior of the modeled system is described by the transitions firing mechanism. If the transition t_j is fired then the marking $M_0(p_i)$ results in new marking $M(p_i)$ such that

$$M(p_i) = M_0(p_i) + O(p_i, t_j) - I(p_i, t_j), \tag{1}$$

Let $R(M_0)$ denote the set of all marking that are reachable from M_0. The incidence matrix C of a PN is an $m \times n$ matrix of integers defined as $C = O - I$ where $c_{ij} = O(p_i, t_j) - I(p_i, t_j)$. The marking $M \in R(M_0)$ is reached when a firing sequence x is executed from M_0 and satisfies the state equation

$$M = M_0 + Cx, \tag{2}$$

where $x : T \to \mathbb{N}$ is the firing vector sequence given by $x = [x_1, ..., x_n]^t$ where x_j is the number of times that t_j is fired.

4 Navigation graph

The graph representing the navigation process under our notations is called a *navigation graph* (NG). This graph is defined as follows: $NG = \{N, P_r \cup P_u\}$ where N is a Petri net and $P_r = \{P_{r1}, P_{r2}, \ldots, P_{rn_1}\} \subset P$ is the set of private places and $P_u = \{P_{u1}, P_{u2}, \ldots, P_{un_2}\} \subset P$ is the set of public places with $n_1 + n_2 = n$ and $P_r \cap P_u = \emptyset$

A given navigation graph NG can be composed of public or private areas. An area A_i is defined by the maximum number of linked places of the same type. By linked places we mean that for every pair of distinct places p_i and p_j in the area there exists a path from p_i to p_j. Besides, $A_i \cap A_j = \emptyset, \forall i \neq j$.

Roughly speaking, private areas describe policies used to access critical resources, whereas public areas define non-constrained individual navigation. In the following, precise definitions of both area types are presented.

4.1 Public areas

Navigation in public areas is left to the user's discretion. The navigation process is not constrained and links between resources are typically asynchronous. When a user enters to a public area, the whole area is duplicated so that the user may navigate independantly from other users. Navigation steps remain confidential to each user. Such areas are adapted to individual navigation.

4.2 Private areas

Private areas make use of synchronous links to allow user groups constitution. In particular, external links must make use of synchronization mechanisms to avoid

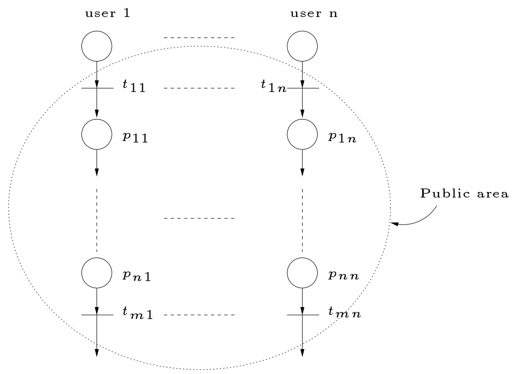

Fig. 1. PN model for public area

resource shortage and manage access priorities and utilization delais. Token-based semaphores principle may be applied to such links, figure 2.

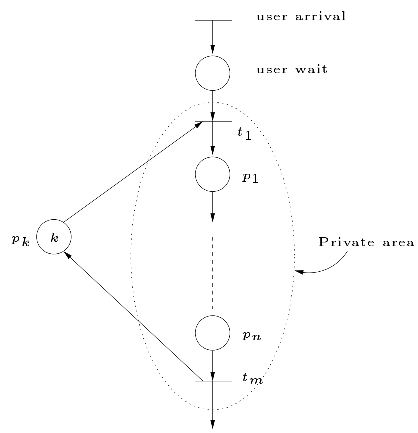

Fig. 2. PN model for semaphore

Broadly speaking, private area can be caracterized by event-based synchronization, Non inter-bloking rules and Lifetime rule.

Event-based synchronization Event-based synchronization aims to allow private area to take appropriate actions in response to a notification event. The event can be either internal or external to the area. For example, when several users access to shared data, e.g. camera, and at the same time the later break down, they are redirected to an error place. Thus, the corresponding fired transition is an external event-based synchronization. An internal event can be merely the arrival of the president in the conference.

The event-based synchronization is used to create a sequential access order between two or more private areas. This implies the notion of *enabled* and *disabled* private areas. Figure 3 presents an example for this sequential access. Indeed, the private area A_2 is disabled until a user leaves the private area A_1. A_1 provides the user with the necessary knowledge to be able to manipulate A_2's resources. When users leaves A_1 the transition t_{21} is fired, in this case the private area is said enabled.

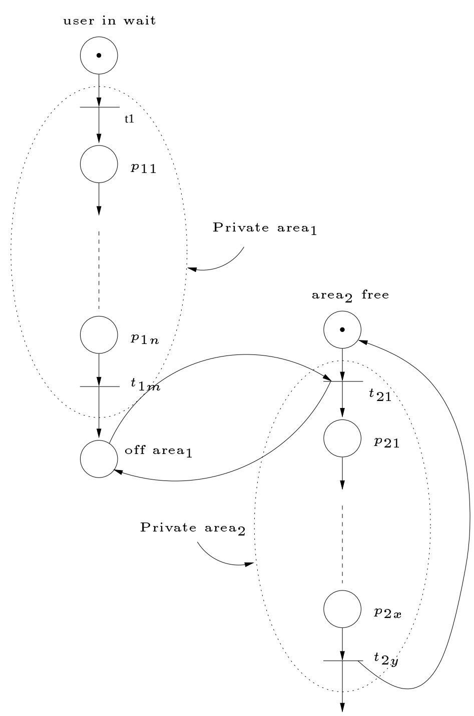

Fig. 3. PN model for sequential access

Non inter-bloking rules We define a set of access priorities for each user. A priority stands for a combination of a user and a private area. The prior-right users will be able to access a private area first. They are considered as priviledged users. It may happen that some users having similar acces priorities whish to enter into a private area when there is only one entry token left, figure 4. In this case, a user will be randomly elected to gain access rights to the private area.

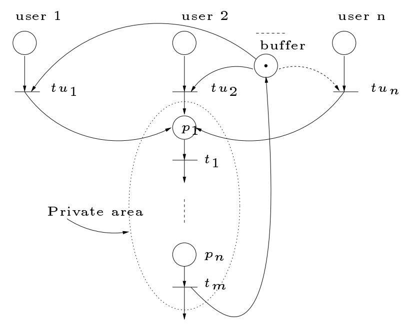

Fig. 4. PN model for mutual exclusion

Lifetime rule Every private area allows users to keep an entry token during a maximum amount of time, called a session lifetime. Once the time spent by a user exceeds the lifetime allowed, the user work stops and he has to leave the private area. This can be modeled by possibility to leave the current private area independently from the place we are.

5 Example

Figure 5 shows the mutual exclusion problem. Our example consists of two users who want to access to a given private area. This area allows to manipulate a robot in order to execute a specific task.

The resource utilisation protocol consists of the three following steps:

1. the users request the robot;
2. one of them (acording to his priority) accesses to the private area and munipulates the robot;
3. the other users wait until the private area is available.

The following code fragment shows the form of a our mutual exclusion policy.

```
PROCESS user(i)
      WHILE (running ())
```

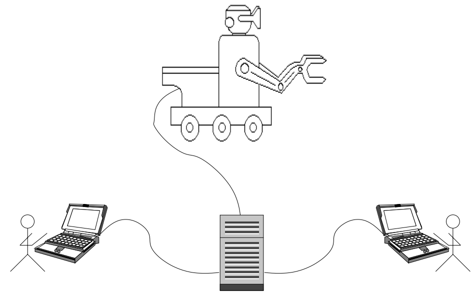

Fig. 5. Example of private area resource access

```
SEND (private area, REQUEST (priority))
RECEIVE (private area, ACCESS)
Enter the private area
Perform some computation on the private area
SEND (private area, LEAVE)
```

```
PROCESS private area
     requesters={}
     WHILE (running ())
          RECEIVE (user(i), REQUEST (priority)) for some i
          APPEND (requesters, {user(i), priority})
          SELECT the first user in requesters list such that his priority
          is maximum
          SEND (user(j), ACCESS)
          RECEIVE (user(j), LEAVE)
          REMOVE (requesters, {user(j), priority})
```

A Petri Net implementing this algorithm is shown in Figure 6, where the places and transitions correspond to the following states and actions:

p_1 (p_3) User 1 (2) requests to access the private area,
p_2 Private area free,
p_4 (p_5) User 1 (2) waiting for response to request,
p_7 Private area in use,
p_6 (p_8) User 1 (2) enters the private area and can perform some computation,
p_9 (p_{10}) User 1 (2) has left the private area,

t_1 **(**t_2**)** User 1 (2) sends a request

t_3 **(**t_4**)** User 1 (2) receives an access permission

t_7 **(**t_8**)** User 1 (2) leaves the private area,

t_5 **(**t_6**)** User 1 (2) prepares another request.

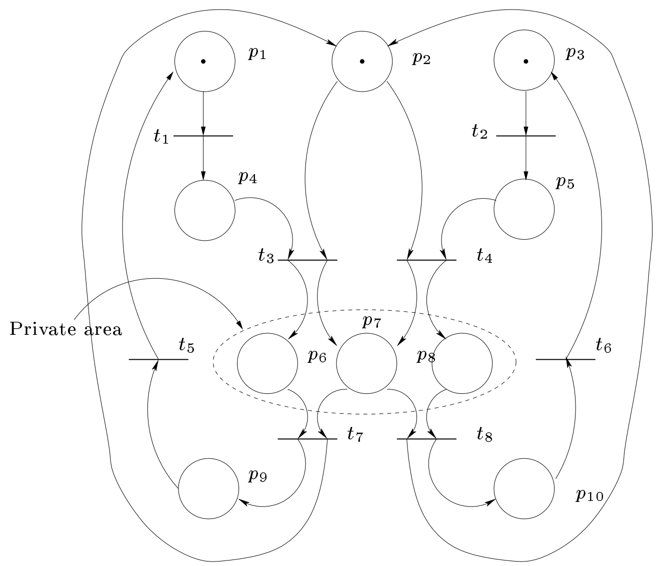

Fig. 6. Petri Net of the example

Remark 1. It is important to note that:

1. each private area must be repeated infinitely so that the area is accessible as soon as it is freed. For our example this property is characterized by the two transitions t_7 and t_8 which marked the place P_2.
2. The place P_2 indicates the number of available resource within the private area which corresponds in our example to the number of robots.
3. the model must be bounded to guarantee that there will be no overflows in the shared resources, i.e., the robot. The following section gives an answer of this question

6 Structural analysis

Boundedness is important structural property that a PN must verify. Indeed, quite often in a Petri net, place are used to model resources. By checking that the net is bounded, it is guaranteed that there will be no overflows in the resources. If it is unbounded, then we may have a modeling error.

Structural properties are independent of the initial marking of the Petri net [4]. They are characterized in terms of the topological structure of the net. A Petri net is said to be k-bounded or simply bounded if the number of token in each place does not exceed a finite number k for any marking reachable from M_0, i.e. $M(p_i) \leq k$. It is said to be structurally bounded if it is bounded for any finite initial marking.

There are three main methods of analysis which allow behavioral and structural properties to be verified. The first method is the coverability tree method introduced by Karp and Miller [5]. It involves the enumeration of all reachable markings of the net. However, this method is limited to small nets and may become unusable when the analysis of large system is involved. The second method uses reduction techniques to facilitate the analysis of large systems. The reduction does not provide equivalent Petri nets but enables certain properties to preserved [6]. This method is powerful but, in the case where the Petri net does not present the required properties, it is difficult to find out and correct the design mistakes. The third method, based on linear algebra, is particularly interesting because the state space explosion problem which is a salient feature of concurrent systems is avoided.

Structural analysis method based on linear algebra can be applied to general Petri nets. However, the most satisfactory results are obtained when the scope is limited to subclasses of Petri nets such as marked graph, state machines [7]. It is basically this fact together with the general structure of our system which motivate us to propose a complete and unified approach for the study of boundedness property of general Petri nets. This approach is based on a modification of the classical incidence matrix which results in a square matrix. The modified incidence matrix eigenvalues are computed and used to prove boundedness of a general Petri nets. The following lemma is a characterization of boundedness property [4].

Lemma 1. *A Petri net is bounded if there exists $x > 0$ such that$C^t x \leq 0$.*

First of all, we define a matrix A to be in Z if and only if all off-diagonal elements of A are nonpositive [8].

Now, consider the matrix $U = [u_{ij}]_{m\times n}$ defined by

$$u_{ij} = \begin{cases} 1 & \text{if } p_i \in {}^\circ t_j \text{ or } t_j^\circ \ , \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where ${}^\circ t$ and t_j° are the sets of input and output places of transition t_j respectively.

Let $R = U^t C$. Then R is $n \times n$ square matrix. Define the matrix B in the following manner

$$b_{ij} = \begin{cases} |r_{ij}| & \text{if } i \neq j \ , \\ r_{ij} & \text{if } i = j. \end{cases}$$

The results given in the Appendix are used to prove the structural properties of a Petri net. We begin with the following result which is a characterization of boundedness of Petri nets.

Theorem 1. *If all real eigenvalues of $-B$ are positive then N is structurally bounded.*

Proof. To prove theorem 1 we must show that $-R \in \mathbb{P}$. Indeed, suppose $-R \notin \mathbb{P}$ then property (c) of $\mathbb{P}$ matrix is violated (Definition A_1 in Appendix). Hence, there exists a diagonal matrix Σ with diagonal elements $+1$ or -1 such that $\Sigma R \Sigma$ is in Alternative I. But from the hypothesis all real eigenvalues of $-B$ are positive, then $-B \in \mathbb{P}$ (Definition A_1) and hence Alternative II holds for $-B$ in view of remark A_1 in the appendix, i.e. there exists $\lambda > 0$ such that $-\lambda^t B > 0$. Let $\Delta = \text{diag}(B) = \text{diag}(R)$. Then $-\lambda^t B > 0$ implies

$$-\lambda^t B = -\lambda^t \Delta - \lambda^t (B - \Delta) > 0$$

it follows

$$-\lambda^t \Delta > \lambda^t (B - \Delta)$$

hence

$$-\lambda^t \Delta + \lambda^t \Sigma (R - \Delta) \Sigma > \lambda^t (B - \Delta) + \lambda^t \Sigma (R - \Delta) \Sigma \tag{4}$$

Note that $\Sigma\Delta\Sigma = \Delta$. Therefore (4) yields

$$\lambda^t \Sigma R \Sigma > \lambda^t (B + \Sigma R \Sigma)$$

Or $(B + \Sigma R\Sigma)_{ij} = b_{ij} + \sigma_i \sigma_j r_{ij}$, where σ_i is a diagonal entry of Σ. Hence, if $\sigma_i\sigma_j = 1$ then $(B + \Sigma R\Sigma)_{ij} = b_{ij} + r_{ij} = |r_{ij}| + r_{ij} \geq 0$ and if $\sigma_i\sigma_j = -1$, then $(B + \Sigma R\Sigma)_{ij} = b_{ij} - r_{ij} = |r_{ij}| - r_{ij} \geq 0$. Therefore, for all $\sigma_i\sigma_j = \pm 1$ we have $(B + \Sigma R\Sigma)_{ij} \geq 0$ and hence $(B + \Sigma R\Sigma) \geq 0$. Therefor $\lambda^t \Sigma R\Sigma > 0$ which implies that $\Sigma R \Sigma$ is in Alternative II, which is contradiction. Therefore, $-R \in \mathbb{P}$ and hence Alternative II holds for $-R$, i.e. there exists $y > 0$ such that $-R^t y > 0$. Since $R = U^t C$, then $-C^t U y > 0$. Let $x = Uy > 0$, then there exists $x > 0$ such that $C^t x < 0$. Consequently, N is structurally bounded.

Theorem 1 requires the positiveness of all eigenvalues of $-B$ for verifying the boundedness of Petri net. From practical and theoretical point of view, it is quite important and useful to investigate the question of when the Petri net preserves the desired property when only the nonnegativeness of eigenvalues of $-B$ is required. The following result gives an answer to this question.

A matrix A is said to be reducible if there exists a non-empty set $F \subset K$, $F \neq K$, such that $a_{ij} = 0$ for $i \in F$ and $j \in K - F$, where a_{ij} is the value of A at the point (i, j). A matrix A is irreducible if it is not reducible. According to [9], a matrix A is irreducible if and only if the associated directed graph is strongly connected. An unweighted Petri net can be drawn as a directed graph where arcs correspond to places and nodes to transitions. A directed graph is said to be strongly connected if for every pair of distinct nodes i and j, there exists a directed path from i to j as well as on from j to i.

The following theorem is concerned with structural boundedness for strongly connected Petri nets.

Theorem 2. *If $-B$ is irreducible and all real eigenvalues of $-B$ are nonnegative, then N is structurally bounded.*

Proof. First note that $-B$ is in Z. Hence, according to Theorem A_2, if $-B$ is irreducible and all real eigenvalues of $-B$ are nonnegative, then there exists $y > 0$ such that $-B^t y \geq 0$. Since $R^t \leq B$, it follows $R^t y \leq 0$. But $R^t = C^t U$, then $C^t U y \leq 0$. Let $x = Uy > 0$, then there exists $x > 0$ such that $C^t x \leq 0$. Hence, N is structurally bounded.

For our model, the modified matrix $-B$ is as follow,

$$B = \begin{pmatrix} -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -2 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & -2 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

The eigenvalues of the modified matrix $-B$ are:

$$\lambda_1 = \lambda_2 = 0; \qquad \lambda_3 = \lambda_4 = 0.4502 \pm 0.6855i$$
$$\lambda_5 = \lambda_6 = 1.1847 \pm 0.8873i; \qquad \lambda_7 = 3.3422$$
$$\lambda_8 = 1.3266$$

The matrix B is irreducible. All condition of theorem 2 are fulfilled, we may conclude that the petri net is structurally bounded.

7 Conclusion

In this work we introduced the concept of both private and public areas for cooperative navigation. Based on Petri net modeling, we provide several patterns and rules to help design multimedia applications. Integrating some of the presenting patterns we can easily build a complete navigation process.

the complexity of the existing methods such as the coverability tree approach and the reduction techniques for deciding the properties for a Petri net before it is implemented makes their application prohibitive when a large systems are concerned. We have shown in this paper the computation simplicity offered by the eigenvalues method in handling such tasks. The method is powerful in the sense that the computation of eigenvalues may be done by the well-known MATLAB.

8 Appendix

Definition A_1: [8], $\mathbb{P}$ denotes the class of matrices $A \in \mathbb{R}^{n\times n}$ which satisfy one of the following equivalent conditions.

(a) Every real eigenvalue of A, as well as of each principal submatrix of A is positive.

(b) For each vector $x \neq 0$, there exists an index k such that $x_k y_k > 0$ where $y = Ax$.

(c) For each signature matrix S(here S is diagonal with diagonal entries ± 1), there exists an $x > 0$ such that

$$SASx > 0$$

We state a Lemma of alternatives due to J. von Neumann and O. Morgenstern [10], which we make use of throughout the paper.

Lemma A_1: For any matrix A (not necessarily square) there exists

$$\text{either } \ x \geq 0 \ \text{ such that } \ Ax \leq 0 \qquad \text{(Alternative I)}$$

$$\text{or } \ y > 0 \ \text{ such that } \ A^t y > 0 \qquad \text{(Alternative II)}$$

Remark A_1: From the Lemma of alternative just stated, it is clear that a necessary condition for a matrix A to be in $\mathbb{P}$ is that Alternative II hold for A since Alternative I violates property (b) of $\mathbb{P}$ matrices

The folowing theorems are due to Fildler and Ptak [8]. Let's denote by Z the class of all real square matrices whose off-diagonal elements are nonpositive.

Theorem A_1: For any matrix A, the following statements on $\hat{A}$ are equivalent:

1. The real part of each eigenvalue of $\hat{A}$ is positive.
2. All real eigenvalues of $\hat{A}$ is positive.
3. $\hat{A}^{-1}$ exists and $\hat{A}^{-1} \geq 0$

Theorem A_2: Let $A \in Z$ be irreducible, and let all real eigenvalues of A be nonnegative. Then there exists a vector $y > 0$ such that $A^t y \geq 0$

References

1. F. Garzotto, P. Paolini and D. Schwabe, " HDM-A model for the design of hypertext applications" Proc of hypertext'91, ACM Press, pp.313, 1991.

2. D. Lange, "An object-oriented design method for hypermedia information systems" Proc. 27th Hawaii international conference on system sciences, 1994.
3. D. Schwabe and G. Rossi, " the object-oriented hypermedia model", ACM, vol. 38, pp. 74-86, 1995.
4. Murata, T., 1989. Petri nets: Properties, analysis and applications. *Proceeding IEEE*, pp. 541-580
5. Karp, R., and Miller, R., 1969, Parallel program shemata. J. Computer Systems Science, **3**, 147-195.
6. Berthelot, G., and Terrat, R., 1982, Petri nets theory for correctness of protocols. IEEE Transaction on Communication, **30COM**, 2497-2505. 1982.
7. Amer-yahia, C., Zerhouni, N., and El moudni, A., 1999, Some subclass of Petri nets and the analysis of their structural properties: A new approach. IEEE Transaction on Systems. Man and Cybernetics, **29A**, 164-172.
8. Fieldler, M., and Ptak, V.,1962, On matrices with nonpositive off-diagonal elements and positive principal minors. Czech. Math. J, **12**, 382-400.
9. Lienbeer, D. J. N.,1982, The application of generalized diagonal dominance to linear system stability theory. International Journal Control, **36**, 185-212.
10. Kuhn, H. W., and Tucker, A. W., 1956, *Linear Enequalities and Related Systems* (Princeton university press).