K/CSD/INF-76/3

THE PETRI NET AS A MODELING TOOL

Pamela B. Thomas

Management Systems Department
Computer Sciences Division*
Union Carbide Corporation, Nuclear Division
Oak Ridge, Tennessee

14th Annual Southeast Regional Association for Computing Machinery Conference to be held at Birmingham, Alabama, Thursday, through Saturday, April 22 - 24, 1976.

*Operated by the Union Carbide Corporation for the U. S. Energy Research and Development Corporation

THE PETRI NET AS A MODELING TOOL

Pamela B. Thomas, Union Carbide Corporation/University of Tennessee

ABSTRACT

The Petri net model is presented as a modeling tool for coordination of asynchronous processes. The Petri net is defined and shown to be both flexible in representing concurrency and easy to comprehend. An extended Petri net model is introduced, and its greater flexibility for coordination modeling is demonstrated. The Cigarette Smokers' Problem is modeled with a Petri net, and a Producer - Consumer Problem is solved with the extended Petri net. The net is briefly evaluated with mention of results of an experimental high-level language simulation of the extended Petri net.

INTRODUCTION

Man has great difficulty in comprehending parallel processing, for the human mind normally works in a sequential manner. Sequential operation is quite satisfactory for many computer applications, but imposing such a restraint on software systems results in assigning an artificial time relation to independent concurrent events. Therefore parallelism and asynchronous operation become fundamental concepts for software development, and man must search for aids to overcome his natural resistance to parallel ideas.

One of the most effective tools introduced to aid in understanding any formidible subject is the model, a small copy or representation of an object or concept. Modeling results in an abstraction of the problem into a similar but simpler structure. The use of modeling tools to clarify concurrency in software systems has been vital to the development of sophisticated systems, for it is by modeling that the designer is able to separate the control aspects of the problem from the computational nature and pictorially grasp the parallelism necessary for efficient operation.

In the past decade many models have been introduced for studying parallelism. One of the most prominent of these is the theory of Petri nets. Since studies of Petri nets suggest great versatility and effectiveness in representing many classes of coordination problems, it is of significant value to examine Petri net theory in some detail.

PETRI NET DEFINITIONS

Definition 1 [1]. A <u>Petri net</u> $N=\{T, P, A, B^\circ\}$ is a directed graph consisting of the following:

i) $T=\{t_1,\ldots,t_m\}$ is a finite set of transitions
ii) $P=\{p_1,\ldots,p_n\}$ is a finite set of places
iii) $A=\{a_1,\ldots,a_k\}$ is a finite set of directed arcs of the form $<x,y>$ which either connect a transition to a place or a place to a transition. Each place may have one or more markers in it, or it may be empty. A place is full if it has at least one marker.
iv) $B^\circ=\{<p,n>|p\varepsilon P \text{ and } n\varepsilon N\}$ is the initial marking.

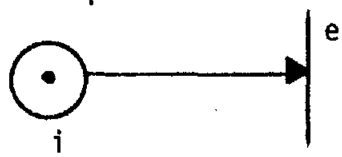
A Petri net N is represented graphically as follows:
i) T and P form the nodes of the graph.
ii) Each place is denoted by a circle and represents a condition.

iii) Each transition is denoted by a bar and represents an event or process.

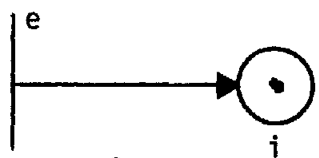
iv) Circles and bars are connected by arcs.

v) A marking is represented by drawing a number of tokens or markers into a place. P_i=number of tokens in place p_i.

For example, a place i with $P_i=1$ connected to a transition e is represented by



and indicates that every occurrence of event e ends one holding of condition i (removes a token from place i). A transition e connected to a place i with $P_i=1$ is represented by



and indicates that every occurrence of event e begins one holding of condition i (deposits a token in place i).

Definition 2 [1]. The set of <u>input places</u> of a transition t_i is the set of all places from which arcs are incident on t_i: $I_i=\{p_j \mid \langle p_j, t_i\rangle \epsilon A\}$.

Definition 3 [1]. The set of <u>output places</u> of a transition t_i is the set of all places onto which arcs are incident from t_i: $O_i=\{p_j \mid \langle t_i, p_j\rangle \epsilon A\}$.

Definition 4 [1]. A transition t_i is enabled if each input place of t_i is full.

Definition 5 [1]. Two transitions t_i and t_j are in <u>conflict</u> if during the simulation the net reaches a marking where both t_i and t_j are enabled and $I_i \cap I_j \neq 0$ (t_i and t_j share an input place).

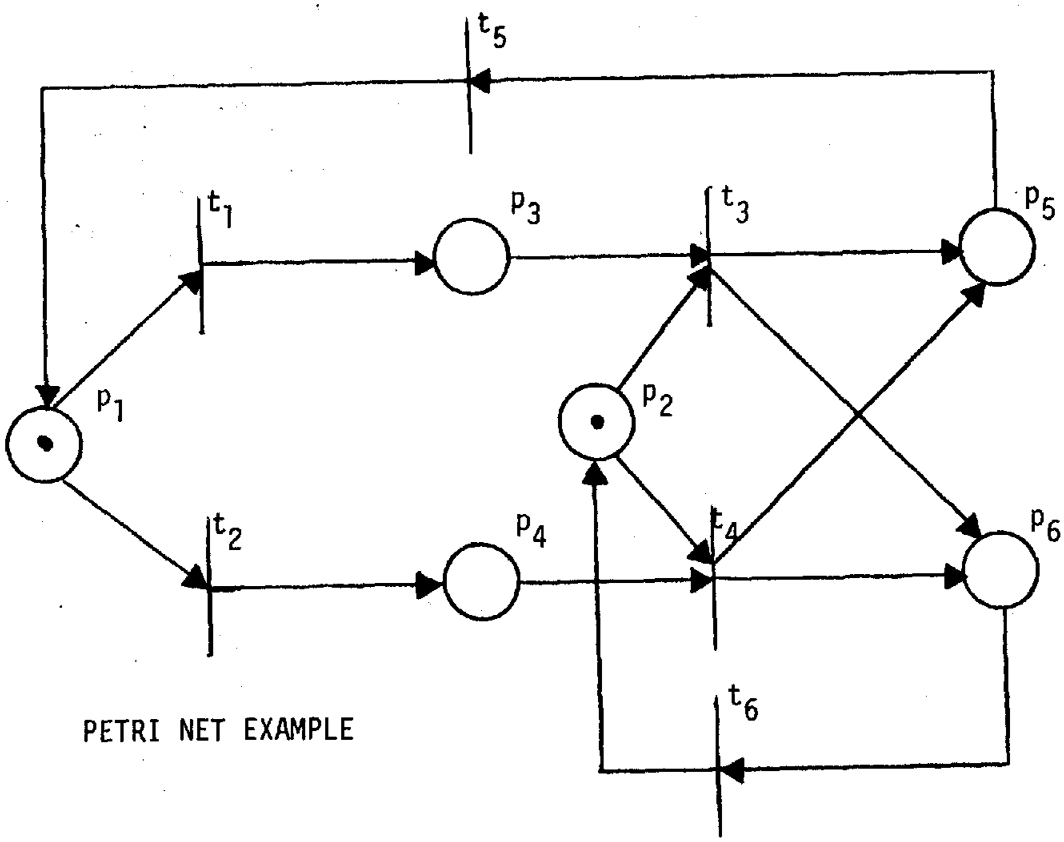
Definition 6 [1]. <u>Simulation rules</u> are as follows: Whenever a transition t_i is enabled, it may at some arbitrary time fire. At such a time it reserves a token in each input place and begins firing. No other transition may claim this reserved token. At the completion of firing (a finite but unknown time) the transition removes the reserved tokens and places one token in each of its output places. If two transitions are in conflict, the decision as to which one will fire is arbitrary and nondeterministic.

Definition 7 [1]. A place p_i in a Petri net is <u>safe</u> with respect to a marking M if no simulation of the net starting from M causes more than one token to be placed in p_i. A marking M is safe if all the places in the net are safe with respect to M.

Definition 8. The <u>reachability set</u> of a Petri net from a marking M is the set of markings derived from all possible firing sequences starting at M.

Definition 9 [1]. A marking of a Petri net is <u>live</u> if for any marking reachable from the given marking, there is a firing sequence that will enable at least one transition of the net.

Example: Consider the Petri net N={T, P, A, B°} on the following page [2]:



PETRI NET EXAMPLE

$T=\{t_1,\ t_2,\ t_3,\ t_4,\ t_5,\ t_6\}$
$P=\{p_1,\ p_2,\ p_3,\ p_4,\ p_5,\ p_6\}$
$A=\{<p_1,t_1>,<p_1,t_2>,<p_3,t_3>,<p_4,t_4>,$
$<p_2,t_3>,<p_2,t_4>,<p_5,t_5>,<p_6,t_6>,$
$<t_1,p_3>,<t_2,p_4>,<t_3,p_5>,<t_3,p_6>,$
$<t_4,p_5>,<t_4,p_6>,<t_5,p_1>,<t_6,p_2>\}$
$B°=\{<p_1,1>,<p_2,1>\}$

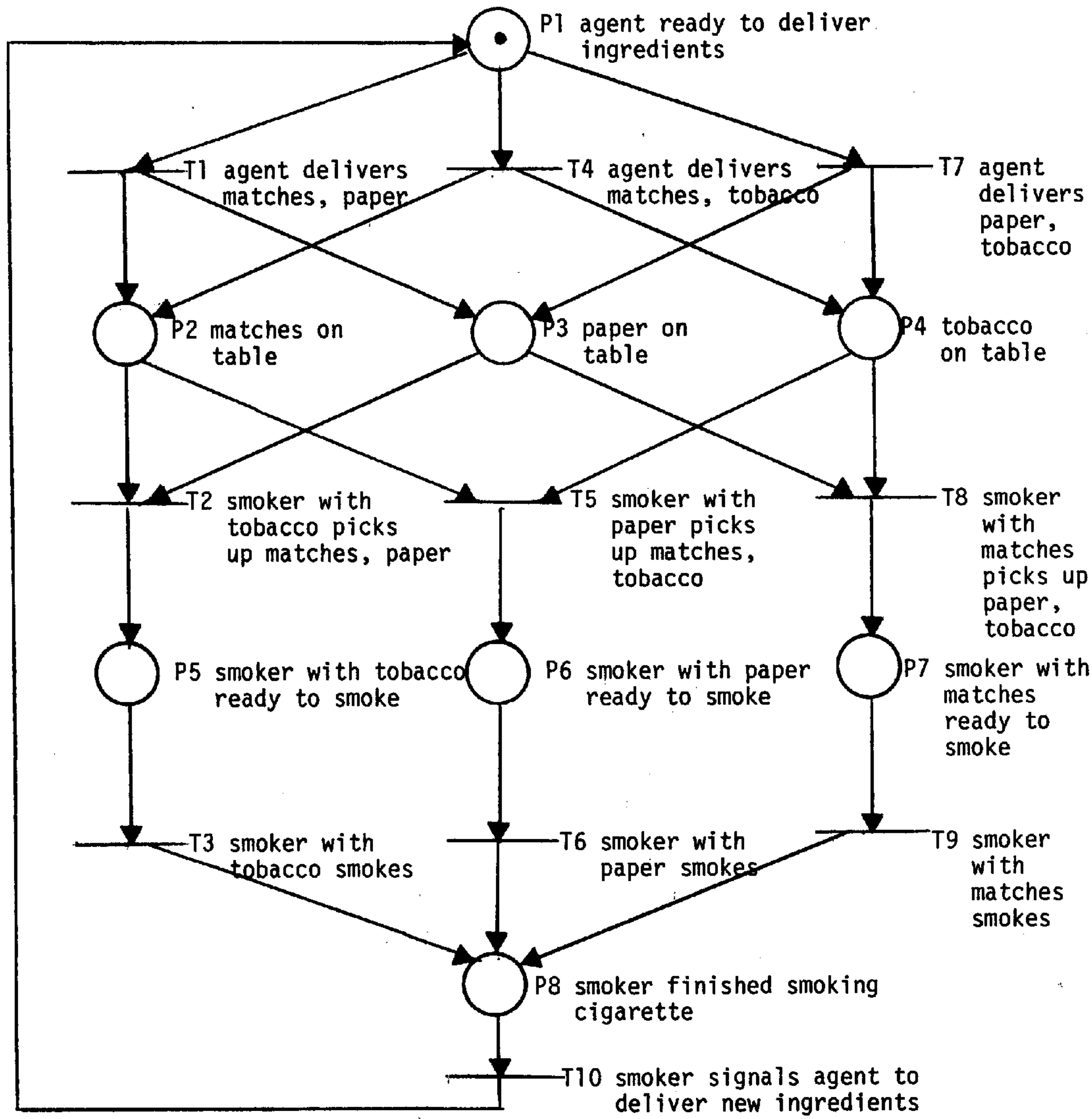
The net indicates that t_1 and t_2 are enabled initially and that these two transitions are in conflict. The net is live and safe with respect to B°. A possible firing sequence is as follows: 1) t_1 chooses to fire 2) t_3 is enabled 3) t_3 fires 4) t_5 and t_6 are enabled 5) t_5 chooses to fire 6) t_1 and t_2 are enabled 7) t_2 chooses to fire 8) t_6 must then fire before the net can proceed: when t_6 fires t_4 is enabled 9) t_4 fires 10) t_5 and t_6 are enabled and the simulation continues.

PETRI NET CAPABILITIES

The Petri net model is a powerful modeling tool, both because of its capability to represent most coordination problems and because it is easy to use and understand. Petri nets have graphical appeal which help overcome the human reluctance to accept concurrent concepts, and the movement of tokens is a natural way to retain a history of the system to influence present behavior. A very difficult activity can often be quickly and easily translated to Petri representation. Consider for example the following problem:

The Cigarette Smokers' Problem [3] Three smokers are sitting at a table. One of them has tobacco, another has cigarette papers, and the third one has matches--each one has a different ingredient required to make and smoke a cigarette, but he may not give any ingredient to another. On the table in front of them two of the three ingredients will be placed, and the smoker who has the necessary third ingredient should pick up the ingredients from the table, make a cigarette, and smoke it. Since a new set of ingredients will not be placed on the table until this action is completed, the other smokers who cannot make and smoke a cigarette with the ingredients on the table must not interfere with the smoker who can. In addition, the process which supplies the ingredients cannot be changed, and no conditional statements can be used.

The desired coordination can be accomplished handily with the below Petri net:



THE CIGARETTE SMOKERS' PROBLEM

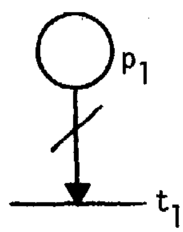
This problem has received considerable attention and is very difficult to model because of a potential deadlock situation. A process may pick up one of the new ingredients placed on the table when the second ingredient placed on the table is not the second ingredient he requires. Since a new set of ingredients will not be placed on the table until the smoker completes smoking a cigarette, each of two smokers could be waiting on a second ingredient indefinitely. It is significant that Petri nets can model the process so easily since Dijkstra's P and V operations [4] can model the desired coordination only with a complex scheme demanding arrays of semaphores [5].

EXTENSIONS TO THE PETRI NET MODEL

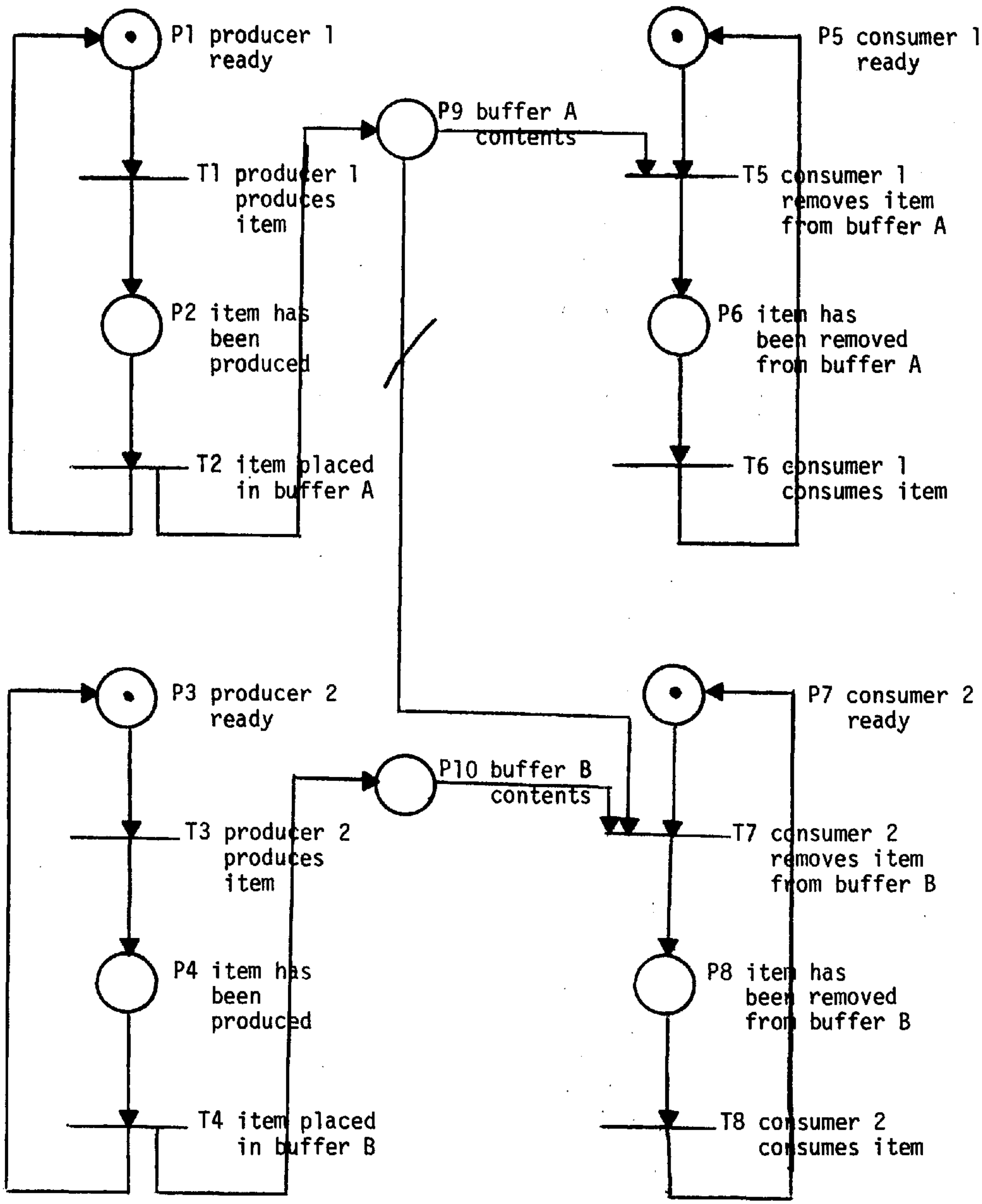
Petri nets cannot model all coordination problems. Consider the following problem proposed by Kosaraju [3]:

<u>The Producer - Consumer Problem</u>. There are four processes: Producer 1, Producer 2, Consumer 1, and Consumer 2. Producers 1 and 2 produce items and place them in unbounded buffers A and B respectively. These four asynchronous processes must interact in some manner so that a consumer does not try to consume when its buffer is empty. In addition, Consumers 1 and 2 share the channel through which they access their own buffers; Consumer 1 is given priority so that Consumer 2 can consume only if Consumer 1's buffer A is empty.

Kosaraju has shown that the Producer - Consumer Problem cannot be modeled with an ordinary Petri net [3]. To increase the power of the ordinary Petri net, several researchers have proposed extensions to the model. One of the most versatile of these augmentations has been offered by Agerwala [1]. He adds to the model another arc between places and transitions:



This arc represents NOT input logic. More formally, in the example above t_1 is enabled if and only if P_1=0. With this extension, Kosaraju's problem can be represented with little difficulty, as shown on the following page.



THE PRODUCER - CONSUMER PROBLEM

CONCLUSION

Examination of the Petri model suggests that it surpasses most well-known concurrent representations in aiding man's understanding of the nature of concurrent processing. A simulation of one of the extended Petri models has proved effective in net analysis for deadlock detection and behavior trends in asynchronous systems and thus for determining the correctness and efficiency of proposed systems.

Although research has not conclusively determined its significance in parallel computation, the Petri net model and its extensions appear to be the basis for a complete model for representing all but possibly the most complex coordination problems. The Petri net concept is powerful, easy to understand, and easy to use; certainly the model will have impact on concurrent software development accuracy and efficiency in the future.

REFERENCES

1. T. Agerwala, Comments on the Capabilities, Limitations, and "Correctness" of Petri Nets, Hopkins Computer Research Reports No. 26, The John Hopkins University, July 1973.

2. J. B. Dennis, Concurrency in Software Systems, Advanced Course in Software Engineering, Lecture Notes in Economics and Mathematical Systems, Ed. M. Bechmann et al. June 1972, pp. 111-127.

3. S. R. Kosaraju, Limitations of Dijkstra's Semaphore Primitives and Petri Nets, Hopkins Computer Research Reports No. 25, Research Program in Computer Systems Architecture, The John Hopkins University, May 1973.

4. E. W. Dijkstra, The Structure of the THE Multiprogramming System, Communications of the ACM, vol. 15, no. 3, March 1972, pp. 171-176.

5. D. L. Parnas, On a Solution to the Cigarette Smokers' Problem (without Conditional Statements), Tech. Report, Carnegie-Mellon University, July 1972.