MODELLING OF CONCURRENT DISTRIBUTED SYSTEMS USING HIGH-LEVEL NETS: SOME THEORETICAL STUDIES

BY

SASWATI Mukherjee (nee Chattopadhyay)

DISSERTATION SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY (ENGINEERING)

DECEMBER, 1998

JADAVPUR UNIVERSITY
CALCUTTA-700 032
INDIA
MODELLING OF CONCURRENT DISTRIBUTED SYSTEMS USING HIGH-LEVEL NETS: SOME THEORETICAL STUDIES

BY
SASWATI MUKHERJEE
(nee CHATTOPADHYAY)

SYNOPSIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY (ENGINEERING)

DECEMBER 1998

JADAVPUR UNIVERSITY
CALCUTTA 700032
INDIA
Since the publication of the doctoral dissertation of Carl Adam Petri in 1962 [11], the complex world of concurrent distributed systems and its modelling had opened up before the researchers. A distributed system consists of a collection of autonomous computers linked by a computer network and equipped with a set of distributed system software that helps to produce an integrated software facility.

Owing to the complexity of distributed systems, which are often large with many interacting components, modelling before actual implementation of these seems to be the most natural prelude. Besides the complexity, concurrency or parallelism is the other characteristic that such systems exhibit. Hence, as in many other fields of computer science, the development of distributed systems usually progresses by the formulation of abstract models for systems that meet a range of general requirements, followed by the design and implementation of systems that translate those models into physical systems. However the process of studying and analyzing these systems with their inherently complex and interacting features becomes even more difficult since these are particularly hard to spell out in explicit language. Hence the task of modelling such systems poses a serious problem to the research world as well as the users. Although much research work has already been done in the world of distributed systems, it is far from complete, especially in the area of modelling. To cope with the problem of modelling, a variety of static analysis techniques, such as formal semantics, finite automata, control flow graphs, a wide variety of Petri nets etc., have been proposed by researchers. These can be used to help in validating the properties of concurrent distributed systems. However, no fixed guideline exists either for comparison or for selecting the best choice amongst the techniques although, quite undoubtedly, Petri Nets, in its various forms, have been the most extensively researched upon as well as used techniques amongst the existing ones.

The present work proposes a tool, the purpose of which is to provide a complete apparatus that will help the users achieve their goal of effectively analyzing a distributed system without incurring much cost or time. This tool belongs to the vast family of high-level nets, which are successors of Petri nets. High-level nets have been found to be suitable for the modelling of a number of different systems, especially concurrent distributed systems.
In its original form, Petri net is composed of four parts: a set of places, a set of transitions, an input function and an output function, wherein the input function is a mapping from a collection of places to a transition and the output function is a mapping from a transition to a collection of places.

However the original Petri net has undergone a sea change. The earliest hint of change can be traced to as early as 1970 brought in by Holt & Commoner [5]. Among those who have contributed in bringing basic changes in the original definition of Petri net, Ramchandani was one of the earliest who introduced the time element in the Petri nets [12]. His work was further progressed by Merlin when time Petri net [9], [10] was introduced. For some specific application purposes, Ramamoorthy and Ho developed the concept of timed Petri net [13]. Another variation of Petri net, in attaching the time element in the net, has come in the form of timing constraint Petri net [14]. Stochastic Petri nets (SPN-s), which added a new dimension in time-related nets, evolved out of the necessity to incorporate relative probability into the changes in the states of various sub-components of the system to be modeled [4]. Attaching colors to the tokens of Petri net (CP nets) was another revolutionary idea that provided a different dimension to Petri nets [7]. Generalized application-oriented variation of Petri net was presented as Predicate/Transition Nets (PrT-net) [6] at which juncture the term high-level net was evolved. Today, most of the practical applications of Petri nets use either PrT-nets or CP-nets although other kinds of high-level nets have also been proposed for modelling and subsequent analysis of different dynamic systems. There have been various attempts to integrate Petri net as basic formalism of concurrency and algebraic specifications for data abstraction. Some of these are Algebraic High-level nets and Algebraic Nets Schemes (ANSs) [15], OBJSA nets [2], Modular Algebraic Nets [3] etc.

However, these high-level nets, in spite of being subjected to a wide spectrum of diversification in various dimensions, are still not adequate enough in modelling a generalized concurrent distributed system with their simultaneous requirement of resource sharing, mutual exclusion and process synchronization. It may be noted that these requirements are driven by the simultaneous occurrence of interprocess communication and mutual exclusion and are not a function of time as a parameter, nor are they probabilistic in nature. Although some recent studies have concentrated on
algebraic manipulations for analyzing concurrent systems, the fundamental problem from
the modelling aspect has remained mostly unresolved. In another attempt of modelling
concurrent systems [1], the base has been built on Lamport’s Fast Mutual Exclusion
Algorithm [8]. This algorithm deals with concurrent programs and tries to solve the
mutual exclusion problem on machines lacking an atomic “test and set” instruction. Thus
it falls short of the essential generality of application to the wide variety of machines that
allow concurrent processes to perform and be controlled by direct “test and set”
instructions.

In the present research work a new concept of a high-level net is being proposed
which is capable of handling the modelling of distributed systems with their inherent
complexity of interdependence along with mutual exclusion. In the thesis, our work is
organized as follows: at first a net in an abstract form has been proposed which forms the
basis of three new nets developed subsequently. This basic net, namely Abstract Colored
Net or ACN, shows the potential of the nets proposed later. Abstract Colored Net or ACN
is defined as a five tuple.

\[ \text{ACN} = \{ P, T, A, TKN, M \} \]

where \( P \) is a set of places, \( T \) is a set of transitions between places, \( A \) is the set of arcs
connecting places and transitions, \( TKN \) is the set of tokens that are present in the places,
and \( M \) is the set of states. In discussing this net, it has been shown that an abstract
system, which exhibits the characteristics of a distributed system, can be successfully
modeled by ACN and the net has been shown to be capable of modelling the
shortcomings of the system.

Next the first of the three nets is described. This new net, namely Variable
Transition Net or VTN, belongs to the basic family of ACN.

A variable transition net (VTN) is defined as a 5-tuple \((P, T_r, T_f, A, M)\)

where

1) \( P = \{ P_{11}, P_{12}, \ldots, P_{1n}, P_{21}, \ldots, P_{2m} \} \)

where \( P_{1i} \) is a process and \( P_{2j} \) is a resource pool.

2) \( T_r = \{ T_{r1}, T_{r2}, \ldots, T_{rm} \} \) is a set of transitions known as \textit{r-}
transitions.
iii) \( T_f = \{T_{f_1}, T_{f_2}, \ldots, T_{f_n}\} \) is a set of transitions known as f-transitions.

iv) \( A \) is a set of arcs connecting the places with transitions.

v) \( M \) is the set of markings with \( M = \{M_1, M_2, \ldots, M_3\} \).

Tokens in VTN are of two types viz. r-tokens representing the resources and f-tokens representing the flags. Since the specific ordering in which the resources appear in the processes decide the sequence in which these resources are acquired by the processes, token ordering is strictly maintained in VTN and the firing of the transitions, which represent the resource-related activities, is done on the basis of this ordering. Indexing of the tokens have been done with the purpose of making the tokens place-sensitive as well as occurrence-sensitive.

It has been shown in this work that the tokens in any net are capable of carrying a lot of information, which actually can help the static analysis of a system being modeled by the net. Two types of transitions in VTN take care of the different types of activities of the system being modeled. While r-transitions (\( T_r \)-s) may fire in two modes viz., direct mode and reverse mode and these model the resource-related activities of the process places (i.e., the 'HOLD' and 'RELEASE' of tokens), f-transitions (\( T_f \)-s) may also fire in two modes viz., ready mode and remove mode and these model the flag-related activities of the process places (i.e., the 'SET' and 'TEST' of flags). Firing rules for the transitions have been discussed along with examples in detail. The most important criteria of modelling any system, i.e., detection of deadlock in all instances along with the prevention of deadlock, wherever possible, has been addressed partly through the design of the transitions and the firing rules and partly through other algorithms as detailed in the dissertation. Figure 1(a) shows a system whose initial model using VTN is shown in figure 1(b).

However, the one major problem faced by VTN is the well-known problem of state-space explosion. With its complete dependence on the number of transitions and with the transitions increasing with increase in the size of the system, state-space grows exponentially.
Process P
- hold a, b, c
- release b, c
- hold d
- set f₁
- release a, d
- test f₂

Process Q
- |
- test f₁

Process R
- |
- set f₂
- test f₁

Figure 1(a). A system with three processes.

Figure 1(b). A typical VTN.

In an attempt to curb this problem, another net has been developed. This net, namely as Extended Colored Net or ECN, although an improvement of VTN, hoists
many a feature that are completely unique in nature. The structure of ECN helps in that the average time of analyzing the system being modeled by ECN is reduced in comparison to that of VTN. ECN is also a member of the ACN family and it has been devised in such a manner that the net can successfully handle the complex characteristics of distributed systems emphasizing the weaknesses of these system when this tool is used for modelling that system. ECN is defined as a 4-tuple \((P, T, A, M)\) where \(T\) stands for only one type of transitions in contrast to the two types in VTN, the rest of the symbols remaining the same. In ECN, although there exists two types of tokens (viz. r-tokens and f-tokens) the indexing of these tokens have been exploited fully. While in VTN token indexing is present in its primitive form, in ECN tokens are indexed more extensively thereby making these carriers of a greater amount of information. Firing of the transitions occurs according to the new set of firing rules laid down in details in the dissertation. ECN is an easier and more pliable tool than VTN. Deadlock detection has been dealt with in the light of the new set of tokens present in the net. Figure 2 shows the initial model of ECN for the system presented in figure 1 (a).

Figure 2. Initial state of ECN for system of Fig 1(a).
It is expected that ECN along with its capacity of static analysis of the system and
deadlock detection as well as resolution wherever possible, coupled with its token-
indexing capacity will become widely acceptable as the modelling tool of distributed
systems.

The state-space explosion problem as present in VTN is still present in ECN,
though in a curtailed form, thereby leaving room for further improvement. Also, to
represent a system that exhibits inherent concurrency, viz. concurrent distributed system,
a tool is required that is capable of exhibiting the same concept of parallelism. The
present work has been rounded off with a discussion on the concurrency aspect of ECN
within its operational behavior. This work, which forms the third net, show that
concurrency can be incorporated in ECN and the transitions can be made firable
concurrently when the initial state of the model is analyzed for that purpose. This work
on ECN, in which the concurrent firing of transitions has been incorporated, is named as
Concurrent ECN or CECN, which describes, in detail, the characteristics of ECN when
the net is made to fire concurrently. The basic net in this work remains the same as in
ECN. Complete new analyses techniques for deadlock detection and prevention have
made this net more acceptable. New analysis is also required for the net to fire
concurrently. And, quite obviously, new firing rules have been devised by which
concurrent firing is made possible. Thus Concurrent ECN or CECN describes the whole
gamut of possibilities that can be exploited when a powerful tool like ECN is made
capable of showing concurrency, thereby making it a new tool on its own right.

This claim is proved in the application area where the CECN is applied to model a
system and it is shown that the strength of this tool lies not only in its capacity to be able
to model a complex system but in its capability of being molded according to the
necessities of the user and the system being modeled. In this work, a system is modeled
that has two, instead of one type of resource-related instructions besides having process
interdependence (executed by flags). Therefore, to model this system, two types of r-
instructions are needed. Also this system is allowed to exhibit the concept of 'dirty read'
and hence that ides also is accommodated in the model. It has been shown in great detail
that all these modifications can be done while modelling this system, although in the
original tool (i.e., in CECN) such provisions are not present.
Extensive bibliographic references, a curtailed version of which has been attached with this synopsis, show the amount of interest the research world has shown so far on the topic of Petri nets as well as distributed systems.

It is needless to emphasize that the basic goal of any software engineering research on any system, especially in case of distributed systems, is to provide developers with cost-effective techniques for evaluating the system. To this end a new family of high-level net, i.e., ACN has been created and it has been shown that the nets based on this basic structure are actually capable of successfully handling the modelling of concurrent distributed systems even when the complexity of such systems are enormous. Thus on the base of ACN we have built VTN which has graduated to ECN finally culminating in the form of CECN. It is expected that these nets will be used extensively and will contribute positively by providing the developers with the much-hunted tool in the much-researched area of modelling of concurrent distributed systems.
LIST OF PUBLICATIONS
BY
THE AUTHOR IN THE RELEVANT AREA.


REFERENCES


