Isomorphisms Between Petri Nets and Dataflow Graphs

KRISHNA M. KAVI, SENIOR MEMBER, IEEE, BILLY P. BUCKLES, SENIOR MEMBER, IEEE, AND U. NARAYAN BHAT

Abstract—**Dataflow graphs are a generalized model of computation. Uninterpreted dataflow graphs with nondeterminism resolved via probabilities are shown to be isomorphic to a class of Petri nets known as free choice nets. Petri net analysis methods are readily available in the literature and this result makes those methods accessible to dataflow research. Nevertheless, combinatorial explosion can render Petri net analysis inoperative. Using a previously known technique for decomposing free choice nets into smaller components, it is demonstrated that, in principle, it is possible to determine aspects of the overall behavior from the particular behavior of components.**

Index Terms—**Dataflow graphs, free choice nets, isomorphism, performance analysis, timed Petri nets.**

I. Introduction

INCREASING interest in dataflow architectures derives in part from the quest for large improvements in performance through parallelism. This interest has given impetus to the development of new representation methods and languages for parallel algorithms. Our interest is in the dataflow graph and its potential to represent any computational structure including computer architectures. The inherent ability of these graphs to represent the natural parallelism in high performance architectures has been noted by others [4], [9], [19].

The chief advantages of dataflow graphs as a computational schema are their compactness and amenability to direct interpretation. That is, the translation from the conceived system to a dataflow graph is straightforward and, once accomplished, it is equally straightforward to determine by inspection which aspects of the system are represented [7], [8]. Unfortunately, the analysis techniques for dataflow graphs are not yet well developed.

It may be possible to develop analytical methods for dataflow graph models independent of Petri nets. However, the homomorphism presented in this paper immediately makes available the vast amount of theory developed with Petri nets to the analysis of dataflow models. Certain abstract properties of Petri nets such as liveness and boundedness have immediate relevance in any general computational schema including dataflow graphs. Other properties such as comparative firing frequencies assume relevance with respect to the semantics of the system being modeled. Thus it is clearly of benefit to establish the correspondence between dataflow graphs and Petri nets in order to combine the representational ease of one with the analysis power of the second.

Manuscript received April 30, 1985; revised July 31, 1986. This work was supported in part by NASA-Ames Research Center under Contract NAG 2-273.

K. M. Kavi is with the Department of Computer Science Engineering, University of Texas at Arlington, Arlington, TX 76019.

B. P. Buckles is with the Department of Computer Science, Tulane University, New Orleans, LA 70118.

U. N. Bhat is with the Department of Statistics, Southern Methodist University, Dallas, TX 75275.

IEEE Log Number 8716571.

Dataflow graphs can be used to model parallel processors [4], [7], [8], [19]. Performance analysis of computer architectures represented as dataflow graphs via Petri nets (more precisely, timed Petri nets) is a goal of this work. The dataflow graphs considered are known as uninterpreted, i.e., the semantics of the data tokens are removed. The nondeterminism introduced is represented by the assignment of probability mass functions to decision points. For those graphs representable in Petri net form, properties such as those mentioned above can be analyzed. In addition, properties dealing with time can be evaluated.

II. Dataflow Graphs and Petri Nets

Formalized treatment of Petri nets is common in the literature [2], [10], [14], [15], [17] and will be dealt with briefly.

Definition 1: A Petri net is a quintuple

$$PN = \langle P, T, D, MP_0, MP_t \rangle$$

where

$P = \{p_1, p_2, \cdots, p_n\}$, a set of places.
$T = \{t_1, t_2, \cdots, t_m\}$, a set of transitions.
$D \subseteq \{P \times T\} \cup \{T \times P\}$, a set of directed arcs.
MP_0 is a given initial marking.
MP_t is a set of terminal markings.

Here we are chiefly interested in extensions to the basic model that incorporate concepts of time.

Timing information has been incorporated in three ways. Sifakis and others [3], [18] associated a nonnegative constant b, with each place having the semantics that an arriving token was "unavailable" until it had been in the place for a time interval of length b. The two other methods attach timing information directly to transitions. One may associate with a transition a nonnegative constant (timed Petri nets [10], [17], [20]) or a probability distribution (stochastic Petri nets [1], [5], [11], [12]). The first case is equivalent to assigning time values to places [18]. In either case, the principal problem to be resolved

0098-5589/87/1000-1127$01.00 © 1987 IEEE

is when to begin the firing epoch—upon arrival of the first token or the instant a transition is enabled. One need also consider whether a second or subsequent epoch can begin while one is in progress.

A second problem to resolve is firing conflicts. Those models that depend on fixed firing time generally assign a probability over the marking space from the current to next marking [20]. Stochastic Petri net models generally use the firing rate (based on random firing times) to determine the next marking from the current one [1], [11], [12]. A difficulty arises if one allows some transitions to have zero firing time. The probability that such transitions will fire once enabled approaches one. The solution is to augment the firing rates with transition probabilities as is done in timed Petri nets. Several investigators have noted the direct correspondence between Petri nets with timing information and Markov processes [11], [12], [20]. In this work, timed Petri nets are employed.

Definition 2: A timed Petri net is the pair

$$\text{TPN} = < \zeta, f >$$

where

ζ is a *PN*.
f: $T \rightarrow \{R^+ \cup \{0\}\}$, a firing time function.

In addition to analyzing the time properties of nets, a goal of this research is the determination of the overall behavior of a system by the inspection of properties of components. Hack [6] first demonstrated necessary and sufficient conditions for liveness and safeness of a subclass of Petri nets important to this work. Ramchandani [16] achieved related results for general nets in the more formal context of solutions to Diophantine equations derived from the connectivity of the net. Solutions to the equations results in subnets (more precisely, T-subnets or P-subnets) whose structure is that of a marked graph or state machine under some circumstances. Ramamoorthy and Ho [17] developed techniques for cycle time computations for such subnets and Magott [10] transformed the method to a solution of a linear program. We have extended this work by showing how the mean time between events of a net composed of marked graph components can be obtained. Coolahan and Roussopoulos [3] have also developed statistical measures of transition firing frequencies and these are adaptable to our model. Datta and Ghosh [2] developed a labeling method that guarantees liveness for nets with transitions of in-degree (and out-degree) at most two (2).

A formal treatment of dataflow graphs has been lacking in the literature due to the purpose that other investigators have used them. Due to the nature of our study and the need to demonstrate homomorphic structures between the dataflow and Petri net models, a formal definition has been developed [8]. The following reviews those results.

Definition 3: A dataflow graph is a labeled bipartite graph where the two types of nodes are called actors and links.

$$\text{DFG} = < A \cup L, E, S, T, g >$$

where

$A = \{a_1, a_2, \cdots, a_n\}$, a set of actors.
$L = \{l_1, l_2, \cdots, l_m\}$, a set of links.
$E \subseteq (A \times L) \cup (L \times A)$, a set of edges such that $(a_i, l_k) \in E \wedge (a_j, l_k) \in E => a_i = a_j$ and $(l_i, a_k) \in E \wedge (l_j, a_k) \in E => l_i = l_j$.
$S = \{l \in L \mid (a, l) \notin E$ for any $a \in A\}$, a nonempty set of links called the starting set.
$T = \{l \in L \mid (l, a) \notin E$ for any $a \in A\}$, a nonempty set of links called the terminating set.
$g = A \rightarrow \{R^+ \cup \{0\}\}$, a firing time function.

Fig. 1 illustrates an uninterpreted dataflow graph. (By "uninterpreted" it is meant that specific semantics are not given to data tokens or actors.) While links are implicitly present in traditional DFG's, they are not always explicitly displayed as is done here. Note that each link has at most one input and at most one output. Meeting this restriction may require the introduction of dummy actors (e.g., to duplicate an input token on several output links).

Let $I(a)$, $a \in A(I(l), l \in L)$ and $O(a)$, $a \in A(O(l), l \in L)$ denote the sets of input and output links (actors) of actor a (link l), respectively. $|I(a)|$ and $|O(a)|$ must be nonzero for each actor while $|I(l)|$ and $|O(l)|$ are at most one. The notation is directly extended to the places and transitions of Petri nets. However, there are no cardinality constraints on the sets denoted.

A marking of a dataflow graph denotes the presence of absence of tokens in links. A marking is a function $M: L \rightarrow \{0, 1, \cdots, k\}$. When M (or MP for Petri nets) is used it means the vector

$$< M(l_1), M(l_2), \cdots, M(l_m) >$$

A marking is distinguished as an initial marking (terminal marking) if $M(l) \neq 0 \rightarrow l \in S (M(l) \neq 0 \rightarrow l \in T)$.

Associated with each actor are an input and output firing set denoting which links enable the actor and which receive tokens when the actor fires. These sets are denoted F_1 and F_2, respectively.

$$F_1(a, M) \subseteq I(a)$$

$$F_2(a, M) \subseteq O(a)$$

Dataflow graphs exhibit special arcs called control arcs whose purpose is to affect the flow of data at decision points. These do not exist in uninterpreted dataflow graphs used here but a probability mass function over the powerset of $O(a)$ serves the same purpose.

An actor a is enabled in marking M if $M(l) \neq 0$ for each $l \in F_1(a, M)$. The firing of an enabled actor a results in a new marking indicated by $M \xrightarrow{a} M'$.

$$M' = M - < I(a) > + < O(a) >$$

This can be generalized to a firing sequence σ denoted $M \xrightarrow{\sigma} M_p$ where

$$M \xrightarrow{a1} M_1 \xrightarrow{a2} M_2 \xrightarrow{a3} \cdots \xrightarrow{ap} M_p.$$

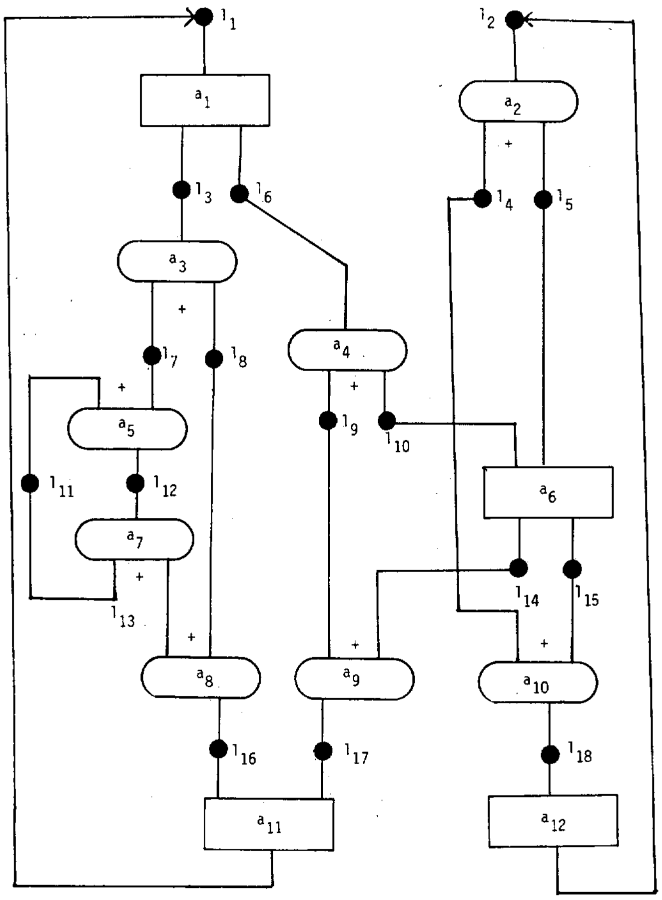
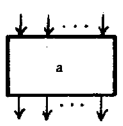


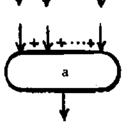
Fig. 1. An uninterpreted dataflow graph.

The forward marking class $\vec{M}$ of a marking M is the set of markings which can be derived (or reached) from M via some firing sequence $\vec{M} = \{M' \mid M \xrightarrow{\sigma} M'\}$.

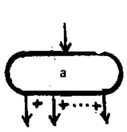
It is simple to extend the semantics given above by assigning a nonnegative real value to each actor representing the time it takes to fire. In the diagrams that follow, the conventions below have been adopted.



an actor for which $F_1(a,M) = I(a)$ and $F_2(a,M) = O(a)$; such an actor is said to have conjunctive input and distributive output



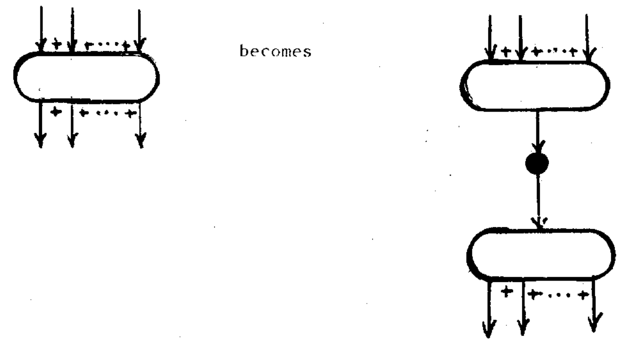
an actor for which $F_1(a,M) \subset I(a)$ and $F_2(a,M) = O(a)$; such an actor is said to have disjunctive input; if it is always the case that $|F_1(a,m)| = 1$, the actor is said to be unit disjunctive



an actor for which $F_1(a,M) = I(a)$ and $F_2(a,M) \subset O(a)$; such an actor is said to have selective output; if it is always the case that $|F_2(a,m)| = 1$, the actor is said to be unit selective

Links are represented by solid circles. For the second type of actor (the disjunctive actor) the enabling input link is chosen nondeterministically.

While it is not permitted for an actor to be simultaneously disjunctive and selective, the restriction is not severe. The uninterpreted nature of the data tokens allows such actors to be separated into two actors.



It is not necessary to enumerate each analogous term for Petri nets such as $\vec{MP}$ and $MP \xrightarrow{t} MP'$. Yet it should be noted that the standard firing set semantics are

$$FP_1(t, MP) = I(t)$$

$$FP_2(t, MP) = O(t).$$

The effect of firing an enabled transition t is

$$MP' = MP - \langle I(t) \rangle + \langle O(t) \rangle.$$

III. Graph to Net Transformations

Transformation of graphs representing asynchronous processes to Petri nets occurs frequently in the literature. (See [15] for examples.) Therefore, we will treat the topic informally. Let DFG be an arbitrary uninterpreted dataflow graph.

Algorithm A1

1) Let $MP_0 \leftarrow M_0$ and $MP_t \leftarrow M_t$.

2) For each $l_i \in L$ in DFG, create $p_i \in P$ in the TPN

3) For each conjunctive actor $a_i \in A$ in DFG, create a transition $t_i \in T$ in TPN such that if $l_j \in O(a_i)$ then $p_j \in O(t)$, if $l_j \in I(a)$ then $p_j \in I(t)$

4) for each disjunctive actor $a_i \in A$, perform the transformation shown in Fig. 2(a). Create a unique transition for each $l_j \in I(a_i)$. If $l_j \in O(a_i)$ then $p_j \in O(t_i^{(h)})$ for each $t_i^{(h)}$.

5) For each selective actor $a_i \in A$, perform the transformation shown in Fig. 2(b). Create a unique transition for each $l_j \in O(a_i)$. If $l_j \in I(a_i)$ then $p_j \in I(t_i^{(h)})$ for each $t_i^{(h)}$.

6) If $g(a_i) = \alpha$, then let $f(t_i) = \alpha$. If more than one transition was derived from a_i, the time associated with each is α.

Let the symbol $\sim$ mean "derived from." It may be used with individual components ($t \sim a$) or entire graphs (TPN $\sim$ DFG). Algorithm A1 is reversible. Thus it is possible to reconstruct the dataflow graph from the Petri net. In this context, we can use $t \sim a$ and $a \sim t$ ($p \sim l$ and $l \sim p$) interchangeably. Fig. 3 shows a Petri net de-

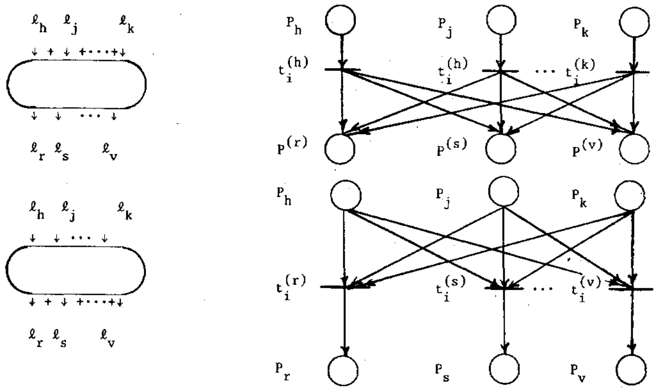


Fig. 2. Graph to net transformations. (a) Disjunctive actors. (b) Selective actors.

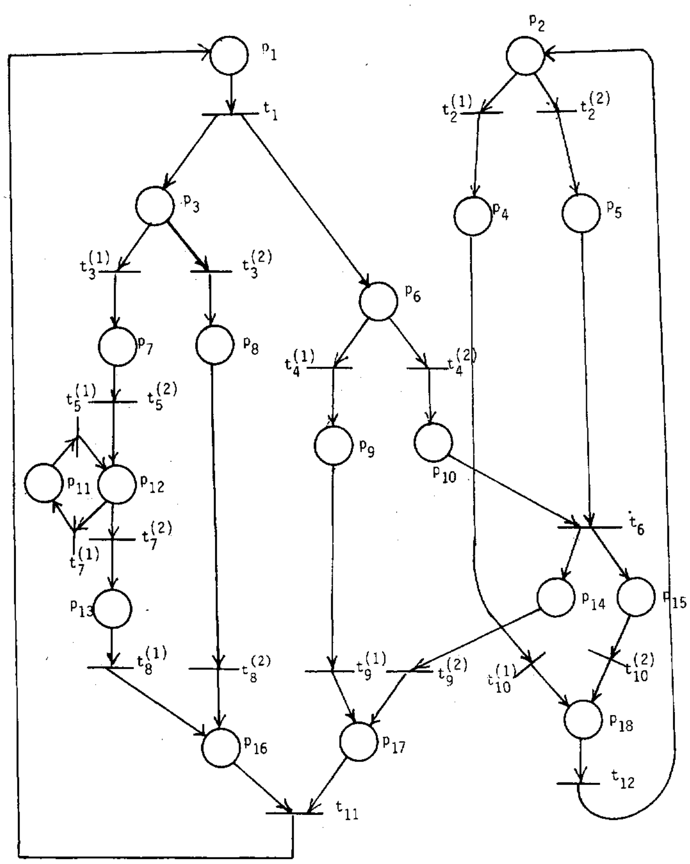


Fig. 3. Illustration of Algorithm A1.

rived from the DFG of Fig. 1 using the transformation above. Let an arbitrary M be represented by the vector $< m_1, m_2, \cdots, m_n >$ where each m_i is an integer. An arbitrary MP can be represented similarly. Then when we say $M = MP$ we mean simple vector equality.

Definition 4: A unit disjunctive DFG is one for which $F_1(a, m) \in I(a)$ for all disjunctive actors. A unit selective DFG is one for which $F_2(a, m) \in O(a)$ for all selective actors.

Theorem 1 (Isomorphism): Given a unit disjunctive and unit selective DFG (i.e., $|F_1(a, m)| = 1$ and $|F_2(a, m)| = 1$), if TPN $\sim$ DFG, $M = MP$, and $M \xrightarrow{\sigma} M'$ then there exists σ' such that $MP \xrightarrow{\sigma'} MP'$ and $M' = MP'$. Conversely if $MP \xrightarrow{\sigma'} MP'$ there exists σ such that $M \xrightarrow{\sigma} M'$, $M = MP$, and $M' = MP'$.

Proof: We prove the first part by demonstrating how to select $t_1, t_2, \cdots, t_k$ in σ' that correspond to $a_1, a_2, \cdots, a_k$ in σ. Let a_1 be the first actor in the sequence $M \xrightarrow{a_1} M_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} M'$. $M_1 = M - < I(a_1) > + < O(a_1) >$. If a_1 is a conjunctive actor, select $t_1 \sim a_1$. If a_1 is disjunctive select $t_1^{(r)} \sim a_1$ such that $l_r \in F_1(a_1, M)$ is the link that enabled a_1. If a_1 is selective select $t_1^{(r)} \sim a_1$ such that $l_r \in F_2(a_1, m)$ is the link upon which the token is produced. If $MP \xrightarrow{t} MP_1$ then clearly $< I(a_1) > = < I(t_1) >$ and $< O(a_1) > = < O(t_1) >$. Thus $M = MP_1$. Choosing $t_2, t_3, \cdots, t_k$ in the same manner produces $MP' = M'$. The converse is proved similarly. Q.E.D.

The above theorem demonstrates that properties of a particular dataflow graph can be discovered by examining the derived Petri net. Before describing these methods, however, we introduce an additional term and its relation to the transformation described.

Definition 5: A free choice net is a Petri net for which each arc from a place to a transition is a unique output of the place or the unique input of the transition. More formally,

$$(\forall (p, t) \in D)\big(O(p) = \{t\} \text{ or } I(t) = \{p\}\big).$$

Theorem 2: If $|I(a)| = 1$ for each selective actor of DFG and TPN $\sim$ DFG then TPN is a timed free choice net.

Proof: Recall that by definition a DFG link has a single input and a single output actor. Thus, $|O(l)| = 1$ for all $l \in L$. If $(l, a) \in E$, $a \in A$ is conjunctive, $p \sim l$, then $|O(p)| = 1$ since for conjunctive actors there is a one-to-one correspondence between $(l, a) \in E$ and $(p, t) \in D$. If $(l_i, a) \in E$, $a \in A$ is disjunctive, $p_i \sim l_i$, then $t^{(i)} \sim a$ is created by step 4 of Algorithm A1 such that $O(p_i) = \{t^{(i)}\}$. If $(l_i, a) \in E$, $a \in A$ is selective then $t^{(j)} \sim a$ is created by step 5 of Algorithm A1 such that $I(t^{(j)}) = \{p_i \mid p_i \sim l_i, l_i \in I(a)\}$. Clearly $|I(t^{(j)})| = 1$ if $|I(a)| = 1$. Q.E.D.

The unit disjunctive/selective criterion assures a DFG can be converted to a Petri net without combinatorial explosion, $|T| \leq |E|$. If selective actors have a single input, then the DFG is isomorphic to a timed free choice net. This is significant primarily because a considerable body of theory exists for the analysis of free choice nets.

Corollary 1: If TPN $\sim$ DFG then

$$(\forall t_i, t_j \in T)\big(I(t_i) \cap I(t_j) \neq \Phi => I(t_i) = I(t_j)\big).$$

Proof: Note that in DFG, $I(a_i)$ iff $a_i \equiv a_j$ because each link has but one output. (It is impossible for distinct actors to share a link.) Therefore, if two transitions share a place they must be derived from the same actor. If a_i is an actor with conjunctive input, there is only one t_i such that $t_i \sim a_i$ and it shares no input. If a_i is an actor with disjunctive input then a unique transition is created for each element of $I(a_i)$. Again $I(t_i) \cap I(t_j) \neq \Phi$ implies $t_i \equiv t_j$. For a selective actor, a_i, for every $l_j \in I(a_i)$ then

$p_j \in I(t_i^{(h)})$ for every h such that $t_i^{(h)} \sim a_i$. Thus, $I(t_i^{(h)}) = I(t_i^{(k)})$. Q.E.D.

The corollary is well known for free choice nets. Here, however, its proof is based directly on the relationship with dataflow graphs. Its importance is the partitioning of the transitions into blocks $B_1, B_2, \cdots, B_h$ such that if any transition in block B_i is enabled, all are. Further, the firing of a transition in block B_i cannot disable a transition in block B_h if $i \neq h$. This will make it possible to resolve conflicts with probability mass functions over transitions rather than (the ordinary practice) over markings.

A state machine is a net or subnet for which $|I(t)| \leq 1$ and $|O(t)| \leq 1$ for every $t \in T$. A marked graph is a net or subnet for which $|I(p)| \leq 1$ and $|O(p)| \leq 1$ for every $p \in P$. The incidence matrix $C = [c_{ij}]$ of a Petri net is an $n \times m$ matrix where

$$c_{ij} = \begin{cases} -1 & \text{if } (p_i, t_j) \in D \\ 1 & \text{if } (t_j, p_i) \in D \\ 0 & \text{if neither } (p_i, t_j) \in D \text{ nor} \\ & (t_j, p_i) \in D \text{ or both are in } D. \end{cases}$$

For example, the incidence matrix for the free choice net of Fig. 3 is as follows:

$$C = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1
\end{bmatrix}$$

IV. Probabilistic Timing Analysis

An objective of Petri net analysis is to determine overall behavior by decomposing it into components which can be analyzed individually.

Definition 6: A subnet of a Petri net TPN $= \langle P, T, D, MP_0, MP_t, f \rangle$ is another Petri net TPN$' = \langle P', T', D', MP_0', MP_t', f' \rangle$ such that $P' \subseteq P$, $T' \subseteq T$, $D' = D \cap ((P' \times T') \cup (T' \times P'))$, $MP_0(p) \in MP_0'$ if $p \in P'$, $MP_t(p) \in MP_t'$ if $p \in P'$, and $f'(t) = f(t)$ if $t \in T'$ and undefined otherwise.

Let $I(\cdot)'$ and $O(\cdot)'$ be the input/output sets of TPN$'$. TPN$'$ is said to be a T-subnet if for every $t \in T'$, $I(t) = I(t)'$ and $O(t) = O(t)'$. It is said to be a P-subnet if for every $p \in P'$, $I(p) = I(p)'$ and $O(p) = O(p)'$. A net or subnet is said to be strongly connected if for every p_i, $p_j \in P$ there is a directed path from p_i to p_j.

A Petri net is said to be live if for every marking in $\vec{MP}_0$ there are firing sequences that enable each transition. A Petri net is bounded if there is a finite number of tokens in each place given any (perhaps infinite) firing sequence. A Petri net is safe if the number of tokens in any place never exceeds one.

Hack [6] gave necessary and sufficient conditions for the liveness and safeness of a marked free choice net. If $MP_0 = \langle 1 \;\; 1 \;\; 0 \;\; 0 \;\; \cdots \;\; 0 \rangle$, Fig. 3 satisfies the conditions. It can be shown [16] that such nets can be decomposed into strongly connected components by finding the simple nonnegative solutions to the system of equations

$$C \cdot Y = 0.$$

A solution is simple if it cannot be additively obtained from other solutions. For Fig. 3,

$$Y_1 = <1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1>$$

$$Y_2 = <0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1>$$

$$Y_3 = <1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0>$$

$$Y_4 = <1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1>$$

$$Y_5 = <1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0>$$

$$Y_6 = <0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0>.$$

Each Y-vector designates a set of transitions. For instance, Y_1 designates $\{t_1, t_2^{(2)}, t_3^{(2)}, t_4^{(2)}, t_6, t_8^{(2)}, t_9^{(2)}, t_{10}^{(2)}, t_{11}, t_{12}\}$ and Y_2 designates $\{t_2^{(1)}, t_{10}^{(1)}, t_{12}\}$. Each transition set together with *all* directly connected places constitutes a strongly connected a T-subnet. For free choice nets, each component is a marked graph.

For components that are timed marked graphs, Ramamoorthy and Ho [17] first showed how to determine the overall cycle time by enumerating the circuits. The time required for each circuit is the sum of the transition times divided by the number of tokens in the circuit. Let K_{ij} be the set of places and transitions in the jth elementary circuit of the ith component.

$$\tau_{ij} = \sum_{t_k \in K_{ij}} f(t_k) \Big/ \sum_{p_k \in K_{ij}} MP(p_k)$$

Assuming r circuits and a firing epoch begins as soon as it is enabled, the overall cycle time for the component is $\bar{\tau}_i = \max(\tau_{i1}, \tau_{i2}, \cdots, \tau_{ir})$.

For example, let $MP(p_1) = MP(p_2) = 1$ and $MP(p_i) = 0$ if $i > 2$ for the component, Y_1, of Fig. 3. The elementary circuits are

$$K_{11} = \{p_1 t_1 p_3 t_3^{(2)} p_8 t_8^{(2)} p_{16} t_{11}\}$$

$$K_{12} = \{p_1 t_1 p_6 t_4^{(2)} p_{10} t_6 p_{14} t_9^{(2)} p_{17} t_{11}\}$$

$$K_{13} = \{p_2 t_2^{(2)} p_5 t_6 p_{15} t_{10}^{(2)} p_{18} t_{12}\}.$$

For concreteness, let $f(t_i^{(j)}) = f(t_i) = \lceil i/3 \rceil$. The overall cycle time $\bar{\tau}_1 = \max(\tau_{11}, \tau_{12}, \tau_{13}) = 12$. For components, $Y_1, Y_2, \cdots$, and Y_6, the cycle times are 9, 10, 14, 14, and 5 (if $MP(12) = 1$), respectively. Magott [10] generalized the technique by showing that the solution could be expressed as a linear program. These methods assume a firing epoch begins as soon as a transition is enabled. When modeling real systems, this is not an unreasonble assumption.

As shown in Corollary 1, free choice net transitions that potentially conflict have the same input place. Because the T-subset components produced as above are strongly connected marked graphs, a place has but one output arc. It follows that a given component is reached from the initial marking by a single transition whose firing is randomly chosen from a conflict set.

With this in mind, we are prepared to deal with the probability mass functions over the output links of selective actors. If the actors are unit selective then $\Pr(F_2(a, M)) > 0$ for each $l \in F_2(a, M)$ and is zero for nonsingleton elements of the powerset of $O(a)$. If $t \sim a$ and $p \sim l \in Q(a)$ then let $\Pr(t) = \Pr(F_2(a, M))$ for l. This contrasts to the normal method of assigning probabilities to markings in the reachability graph but is equivalent due to the partitioning of the transitions described above.

As an illustration of the employment of the foregoing, define $\mathrm{MTTE}(t_j)$ to be the mean time to the event of the beginning of the firing epoch for t_j. Assume t_j has s input places with s independent loop-free paths $\sigma_1, \sigma_2, \cdots, \sigma_s$ from MP_0. The time to traverse σ_i will also be denoted by σ_i. (Dependent paths can be dealt with but serve no purpose at present.) t_j cannot fire until the last token appears at an input. That is, $\mathrm{MTTE}(t_j) = \max(\sigma_1, \sigma_2, \cdots, \sigma_s)$ and each σ_i a constant. Now assume that adjacent to path σ_i there are loops that may be traversed an indeterminate number of times. The path time, also denoted by σ_i, is a random variable and

$$\mathrm{MTTE}(t_j) = E[\max(\sigma_i, \sigma_1, \cdots, \sigma_s)]$$

for which the lower bound is

$$\downarrow\mathrm{MTTE}(t_j) = \max\big(E[\sigma_1], E[\sigma_2], \cdots, E[\sigma_s]\big).$$

(It is well known that $E[\max(X_1, X_2, \cdots, X_n)] \geq \max(E[X_1], E[X_2], \cdots, E[X_n])$ [13].)

For clarity, t will be defined with superscripts and subscripts. Let $t_k^{(m)}$ *adj* σ_i mean that $t_k^{(m)} \notin \sigma_i$ while an equivalently subscripted transition is $(t_k^{(h)} \in \sigma_i, h \neq m)$. By extension, Y_i *adj* σ_i means $t_k^{(m)} \in Y_i$ and $t_k^{(m)}$ *adj* σ_i. Let

$$\lambda_i = \prod_{t_k^{(m)} \in Y_i} \Pr(t_k^{(m)})$$

if Y_i *adj* σ_j and is undefined otherwise.

$$E[\sigma_j] = \sum_{t_k \in \sigma_j} f(t_k) + \sum_{t_k^{(m)} \in \sigma_j} [f(t_k^{(m)}) + A(t_k^{(m)})].$$

The latter term, $A(\cdot)$, represents the expected amount of time within components adjacent to σ_j at transition $t_k^{(m)}$. Let $\lambda_1, \lambda_2, \cdots, \lambda_r$ be the probabilities of the r components adjacent at $t_k^{(m)}$. Let $\lambda_{r+1} = \Pr(t_k^{(m)})$. (Note that $\Sigma_{i=1}^{r+1} \lambda_i = 1$.) $\bar{\tau}_i'$ stands for the expected cycle time for component Y_i. That is, $\bar{\tau}_i'$ differs from $\bar{\tau}_i$ in that it takes into account components adjacent to circuits in Y_i.

$$\tau_{ij}' = E[K_{ij}] \Big/ \sum_{p_k \in K_{ij}} MP(p_k)$$

TABLE I
CIRCUITS

K_{ij}	Sequence
K_{21}	$\{p_2\ t_2^{(1)}\ p_4\ t_{10}^{(1)}\ p_{18}\ t_{12}\}$
K_{31}	$\{p_1\ t_1\ p_3\ t_3^{(2)}\ p_8\ t_8^{(2)}\ p_{16}\ t_{11}\}$
K_{332}	$\{p_1\ t_1\ p_6\ t_4^{(1)}\ p_9\ t_9^{(1)}\ p_{17}\ t_{11}\}$
K_{51}	$\{p_1\ t_1\ p_3\ t_3^{(1)}\ p_7\ t_5^{(2)}\ p_{12}\ t_7^{(2)}\ p_{13}\ t_8^{(1)}\ p_{16}\ t_{11}\}$
K_{52}	$\{p_1\ t_1\ p_6\ t_4^{(1)}\ p_9\ t_9^{(1)}\ p_{17}\ t_{11}\}$
K_{61}	$\{p_{12}\ t_7^{(1)}\ p_{11}\ t_5^{(1)}\}$

TABLE II
INTERMEDIATE MTTE CALCULATIONS

Path	τ'_{ij}	G	$\bar{\tau}'_i$	λ_i
K_{61}	5	5	$\bar{\tau}'_6=5$	1/7
K_{52}	10	10	--	--
K_{51}	14.97	14.97	$\bar{\tau}'_5=14.97$	1/12
K_{32}	10	10	--	--
K_{31}	9	9	$\bar{\tau}'_3$	1/6
K_{21}	9	9	$\bar{\tau}'_2$	1/2
σ_1	--	5.88	--	--
σ_2	--	19	--	--

If, to simplify the subscript scheme, $Y_1, Y_2, \cdots, Y_r$ are adjacent to $t_k^{(m)}$,

$$A(t_k^{(m)}) = \lambda_{r+1} \sum_{m_0=0}^{\infty} \sum_{m_1=0}^{m_0} \cdots \sum_{m_{r-1}}^{m_0 - m_1 - \cdots - m_{r-2}} \cdot \frac{m_0!}{m_1!m_2! \cdots m_{r-1}!(m_0 - m_1 - \cdots - m_{r-1})!}$$
$$\{[m_1\bar{\tau}'_1 + m_2\bar{\tau}'_2 + \cdots + m_{r-1}\bar{\tau}'_{r-1} + (m_0 - m_1 - \cdots - m_{r-1})\bar{\tau}'_r]\, \lambda_1^{m_1}\lambda_2^{m_2} \cdots \lambda_{r-1}^{m_{r-1}}\lambda_r^{(m_0 - m_1 - \cdots - m_{r-1})}\}$$

which can be simplified to:

$$A(t_k^{(m)}) = \frac{1}{\lambda_{r+1}} \sum_{i=1}^{r} \lambda_i \bar{\tau}'_i.$$

Each path σ_i defines a hierarchy over the set of components. For example, in Fig. 3, the path $\{p_1 t_1 p_6 t_4^{(2)} p_{10}\}$ has Y_3 and Y_5 immediately adjacent (at $t_4^{(2)}$) while Y_6 is adjacent to one of the circuits in Y_5. Thus, one must solve Y_6 before Y_5 can be solved.

To illustrate, or Fig. 3 let $\Pr(t_k^{(1)}) = 1/k$ and $\Pr(t_k^{(2)}) = 1 - 1/k$. For $\downarrow$MTTE(t_6),

$$\sigma_1 = \{p_1\ t_1\ p_6\ t_4^{(2)}\ p_{10}\}$$
$$\sigma_2 = \{p_2\ t_2^{(2)}\ p_5\}.$$

Table I contains the cycles for Y_2, Y_3, Y_5, and Y_6 which are the only components needed. Table II contains the relevant intermediate calculations.

$$\downarrow\text{MTTE}(t_6) = \max[5.88, 19] = 19$$

Given the isomorphism theorem and the equivalence of timing between actors and transitions, it can be concluded that $\downarrow$MTTE(a_j) = $\downarrow$MTTE(t_j). If the transition is superscripted then $\downarrow\text{MTTE}(a_j) = \min_k \{\downarrow\text{MTTE}(t_j^{(k)})\}$. The mean time to event is but one measure possible. From it, other measures such as mean time between events can be derived and correlated to components in the real system.

V. SUMMARY AND CONCLUSIONS

Dataflow graphs are useful representations for abstract computations, generally rendering models that are easily related to the real system being modeled. Petri nets, while less powerful computationally, have been studied intensively, giving rise to a large body of analytic methods. Here, we have shown that a significant class of dataflow graphs can be effectively transformed to Petri nets. The applications of this class have been primarily in modeling concurrency in computer systems. Thus, the isomorphism between the two computational models allows considerable analytic capability to be employed.

Within the applications context, timed transitions and probabilistic resolution of nondeterminism are introduced. Using these extensions, a measure (a bound for Mean Time To Event) was illustrated. The principle behind the derivation was the determination of overall behavior by examination of the contained components. This principle is currently being exploited to determine other properties of the net.

REFERENCES

[1] B. Bayaert, G. Florin, P. Lone, and S. Natkin, "Evaluation of computer systems dependability," in *Proc. Symp. Fault Tolerant Comput.*, 1981, pp. 79–81.

[2] A. Datta and S. Ghosh, "Synthesis of a class of deadlock-free Petri nets," *J. ACM*, vol. 31, no. 3, pp. 486–506, July 1984.

[3] J. E. Coolahan, Jr. and N. Roussopoulos, "Timing requirements for time-driven systems using augmented Petri nets," *IEEE Trans. Software Eng.*, vol. SE-9, pp. 603–616, Sept. 1983.

[4] J. L. Gaudiot and M. D. Ercegovac, "Performance analysis of dataflow computers with variable resolution actors," in *Proc. of 4th Int. Conf. Distributed Comput. Syst.*, San Francisco, CA, May 1984, pp. 2–9.

[5] R. Geist, K. S. Trivedi, J. B. Dugan, and M. Smotherman, "Design of the hybrid automated reliability of predictor," in *Proc. IEEE/AIAA Digital Avionics Systems Conf.*, Oct. 1983, pp. 16.5.1–16.5.8.

[6] M. H. T. Hack, "Analysis of production schemata by Petri nets," Massachusetts Inst. Technol., Rep. 94, Project MAC, 1972, 119 pp.

[7] K. M. Kavi, "Data flow modeling techniques," in *Proc. IASTED Int. Conf. Simulation Modeling*, Nov. 1983, pp. 1–4.

[8] K. M. Kavi, B. P. Buckles, and U. N. Bhat, "A formal definition of dataflow graph models," *IEEE Trans. Comput.*, vol. C-35, no. 11, pp. 940–948, Nov. 1986.

[9] W. P. Karplus and A. Makoui, "The role of data flow methods in continuous systems simulation," in *Proc. Summer Simulation Conf.*, Denver, CO, 1982.

[10] J. Magott, "Performance evaluation of concurrent systems using Petri nets," *Inform. Processing Lett.*, vol. 18, no. 1, pp. 7–13, Jan. 1984.

[11] M. K. Molloy, "On the integrating delay and throughput measures in distributed processing models," Ph.D. dissertation, Univ. California, Los Angeles, 1981.

[12] ——, "Performance analysis using stochastic Petri nets," *IEEE Trans. Comput.*, vol. C-51, pp. 913-917, Sept. 1982.
[13] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*, 3rd ed. New York: McGraw-Hill, 1974.
[14] J. L. Peterson, "Petri nets," *ACM Comput. Surveys*, vol. 9, pp. 233-352, Sept. 1977.
[15] ——, *Petri Net Theory and the Modeling of Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1981.
[16] C. Ramchandani, "Analysis of asynchronous systems by Petri nets," Massachusetts Inst. Technol., Tech Rep. 120, Project MAC, 1974.
[17] C. V. Ramamoorthy and G. S. Ho, "Performance of asynchronous concurrent systems using Petri nets," *IEEE Trans. Software Eng.*, vol. SE-6, pp. 440-449, Sept. 1980.
[18] J. Sifakis, "Performance evaluation of systems using nets," in *Net Theory and Applications*, W. Brower, Ed. New York: Springer-Verlag, 1980, pp. 307-319.
[19] V. P. Srini and J. F. Asenjo, "Analysis of Cray 1S architecture," in *Proc. 10th Int. Symp. Comput. Architecture*, June 1983, pp. 194-206.
[20] W. M. Zuberek, "Timed Petri-nets and preliminary performance evaluation," in *Proc. 7th Annu. Symp. Comput. Architecture*, 1980, pp. 88-96.

Krishna M. Kavi (S'76-M'81-SM'86) received the Master's and Ph.D. degrees in computer science from Southern Methodist University, Dallas, TX, in 1977 and 1980, respectively.

He is an Associate Professor in the Department of Computer Science Engineering at the University of Texas, Arlington. Prior to joining UTA, he was an Assistant Professor at the University of Southwestern Louisiana, Lafayette. He consulted for Recognition Equipment, Inc. of Dallas, Motorola of Phoenix, and Communications Enterprises, Inc. of Dallas. His research interests include dataflow modeling techniques, computer systems architecture, distributed operating systems, performance and reliability modeling of computer systems, and software engineering. He has published widely in the area of computer architecture and dataflow models.

Dr. Kavi is a member of the IEEE Computer Society, the Association for Computing Machinery, Sigma Xi, and Upsilon Pi Epsilon.

Billy P. Buckles (SM'82) received the M.S. degree in operations research, the M.S. degree in computer science, and the Ph.D. degree in operations research from the University of Alabama, Huntsville.

Presently, he is an Associate Professor of Computer Science at Tulane University. Previously, he was a member of the faculty at the University of Texas, Arlington. Prior to his appointment at UTA, he was a Technical Staff member at Computer Science Corporation, Science Applications Inc., and General Research Corporation. He has served as Principal Investigator on various NSF funded projects as well as those supported by industrial research labs. He has over a dozen journal publications. Currently, his interests are Petri net modeling as it relates to parallel computing and uncertainty representation in databases using fuzzy set theory.

Dr. Buckles is a member of the IEEE Computer Society, the Association for Computing Machinery, the International Fuzzy Systems Association, and Sigma Xi.

U. Narayan Bhat received the B.A. degree in mathematics from Madras University, India, in 1953, the M.A. degree in statistics from Karnatak University, India, in 1958, and the Ph.D degree in statistics from the University of Western Australia, Perth, in 1965.

He has held faculty positions at Karnatak University (1958-1961), the University of Western Australia (1965), Michigan State University, East Lansing (1965-1966), Case Western Reserve University, Cleveland, OH (1966-1969), and Southern Methodist University, Dallas, TX, from 1969 to the present, where he is a Professor of Statistics and Operations Research. He has also held various administrative positions at SMU between 1976 and 1982, the last being the Vice Provost and the Dean for Graduate Studies. His research interests are in the area of applied probability with particular reference to stochastic modeling of queueing, reliability, and computer systems. He is the author or coauthor of 3 books and more than 50 research articles.

Dr. Bhat is a member of several professional societies including the American Statistical Association, Operations Research Society of America, and the Institute of Management Sciences.