Stochastic Activity Networks: Formal Definitions and Concepts *

William H. Sanders[1] and John F. Meyer[2]

[1] Department of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main St., Urbana, IL 61801, USA.
whs@crhc.uiuc.edu
[2] Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109, USA.
jfm@eecs.umich.edu

Abstract. Stochastic activity networks have been used since the mid-1980s for performance, dependability, and performability evaluation. They have been used as a modeling formalism in three modeling tools (METASAN, *UltraSAN*, and Möbius), and have been used to evaluate a wide range of systems. This chapter provides the formal definitions and basic concepts associated with SANs, explaining their behavior and their execution policy precisely.

1 Introduction

The development of model-based methods for evaluating computer systems and networks has as long a history as the systems themselves. When applied properly, these techniques can provide valuable insights into nonfunctional properties of a system, such as its performance, dependability, or performability. One approach in this regard has been the development of stochastic extensions to Petri nets. These extensions permit the representation of timeliness (real-time constraints) as well as parallelism in a stochastic setting. As models for performability evaluation [1], they also permit the representation of fault tolerance and degradable performance. Use of these nets was facilitated by the early recognition (see [2] and [3, 4], for example) that, with an appropriate definition, their behavior could be represented as discrete-state Markov processes. Motivated by this representational power and solution capability, researchers sought to define particular

* This material is based upon work supported, in part, by the National Science Foundation under Grant No. 9975019 and by the Motorola Center for High-Availability System Validation at the University of Illinois (under the umbrella of the Motorola Communications Center). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or of Motorola.

variants of stochastic Petri nets well-suited to particular application needs or solution methods (DSPNs, GSPNs, SANs, and SRNs, for example).

One stochastic extension of these nets, known as "stochastic activity networks," was defined with the express purpose of facilitating unified performance/dependability (performability) evaluation as well as more traditional performance and dependability evaluation. In the time since their introduction, they have served as the basis for three modeling tools (METASAN [5], *UltraSAN* [6], and Möbius [7, 8]), and have been used to evaluate a wide variety of systems. (See `www.crhc.uiuc.edu/PERFORM` for a partial list of references and information on how to get these tools.)

In order to be effective and generally applicable, a modeling scheme should have a formal basis that describes its primitives and behavior in an unambiguous way. A scheme must also be general enough to allow for easy representation of realistic systems, and formal enough to permit derivation of useful results. This chapter provides the formal definitions and concepts for stochastic activity networks (SANs), a variant of stochastic Petri nets. We first precisely define activity networks. Activity networks are the non-probabilistic model on which SANs are built, just as in a similar fashion, (un-timed) Petri nets provide the foundation for stochastic Petri nets. We then describe the execution of a SAN as a sequence of markings, activity completions, and case selections. With activity networks as a base, we then define stochastic activity networks, and describe precisely when a SAN's behavior is fully quantified in a probabilistic sense. When it is, we say that a SAN is *well-specified*. Finally, we provide basic algorithms for determining when a SAN is well-specified, using the structure of the net to do this efficiently.

While stochastic activity networks have been used since the mid-1980s, their formal definition appears only in dissertation form, and is not generally available. We hope that the definitions and concepts in this chapter will be more accessible, and aid other researchers who are developing and applying formal methods for stochastic evaluation of computer systems and networks.

2 Activity Networks

The desire to represent system characteristics of parallelism and timeliness, as well as fault tolerance and degradable performance, precipitated the development of general network-level performability models known as *stochastic activity networks* [9, 10]. Stochastic activity networks are probabilistic extensions of "activity networks"; the nature of the extension is similar to the extension that constructs stochastic Petri nets from (classical) Petri nets.

2.1 Definitions

Informally (as in [10]), activity networks are generalized Petri nets with the following primitives:

- *activities*, which are of two kinds: *timed* activities and *instantaneous* activities. Each activity has a non-zero integral number of *cases* [1].
- *places*, as in Petri nets.
- *input gates,* each of which has a finite set of inputs and one output. Associated with each input gate are an n-ary computable predicate and an n-ary computable partial function over the set of natural numbers which are called the *enabling predicate* and the *input function*, respectively. The input function is defined for all values for which the enabling predicate is true.
- *output gates,* each of which has a finite set of outputs and one input. Associated with each output gate is an n-ary computable function on the set of natural numbers, called the *output function.*

Timed activities represent the activities of the modeled system whose durations impact the system's ability to perform. Instantaneous activities, on the other hand, represent system activities that, relative to the performance variable in question, are completed in a negligible amount of time. Cases associated with activities permit the realization of two types of spatial uncertainty. Uncertainty about which activities are enabled in a certain state is realized by cases associated with intervening instantaneous activities. Uncertainty about the next state assumed upon completion of a timed activity is realized by cases associated with that activity. Gates are introduced to permit greater flexibility in defining enabling and completion rules.

Before formally defining an activity network, it helps to define several related concepts in a more precise manner. Let P denote the set of all places of the network. If S is a set of places ($S \subseteq P$), a *marking of* S is a mapping $\mu : S \rightarrow I\!N$. Similarly, the set of *possible markings of* S is the set of functions $M_S = \{\mu \mid \mu : S \rightarrow I\!N\}$. With these definitions in mind, an *input gate* is defined to be a triple, (G, e, f), where $G \subseteq P$ is the set of *input places* associated with the gate, $e : M_G \rightarrow \{0, 1\}$ is the *enabling predicate* of the gate, and $f : M_G \rightarrow M_G$ is the *input function* of the gate. Similarly, an *output gate* is a pair, (G, f), where $G \subseteq P$ is the set of *output places* associated with the gate and $f : M_G \rightarrow M_G$ is an *output function* of the gate. One can then formally define an activity network in terms of allowable interconnections between these model primitives.

Definition 1 *An* activity network *(AN) is an eight-tuple*

$$AN = (P, A, I, O, \gamma, \tau, \iota, o)$$

[1] The term *case*, as used here, should not be confused with the notion of *cases* of elementary net systems [11]. Here the term *case* is used to denote a possible action that may be taken upon the completion of an event.

where P is some finite set of places, A is a finite set of activities, I is a finite set of input gates, and O is a finite set of output gates. Furthermore, $\gamma : A \rightarrow \mathbb{N}^+$ specifies the number of cases for each activity, and $\tau : A \rightarrow \{Timed, Instantaneous\}$ specifies the type of each activity. The net structure is specified via the functions ι and o. $\iota : I \rightarrow A$ maps input gates to activities, while $o : O \rightarrow \{(a, c) | a \in A$ and $c \in \{1, 2, \ldots, \gamma(a)\}\}$ maps output gates to cases of activities.

Several implications of this definition are immediately apparent. First, each input or output gate is connected to a single activity. In addition, each input of an input gate or output of an output gate is connected to a unique place. In contrast to the definition in [10], different output gates and input gates of an activity may be connected to identical places, as has been done in practice. Ambiguity in the execution of the net is avoided by requiring that the marking obtained upon completion of each activity not depend on 1) the order of application of the input gate functions, or 2) the order of application of the output gate functions.

The following definitions aid in the discussion that follows.

Definition 2 *If $AN = (P, A, I, O, \gamma, \tau, \iota, o)$ is an activity network and $S, G \subseteq P$ then*

1. *a mapping $\mu : P \rightarrow \mathbb{N}$ is a* marking of the network*,*
2. *for $S \subseteq P$, $\mu_S : S \rightarrow \mathbb{N}$ is the restriction of μ to places of S (i.e. $\mu_S(p) = \mu(p), \forall p \in S$),*
3. *an input gate $g = (G, e, f)$* holds *in a marking μ if $e(\mu_G) = 1$,*
4. *an activity a is* enabled *in a marking μ if g holds for all $g \in \iota^{-1}(a)$,*
5. *a marking μ is* stable *if no instantaneous activities are enabled in μ,*
6. *the* input places of an activity *a consist of the set $IP(a) = \{p \mid \exists\ (G, e, f) \in \iota^{-1}(a)$ such that $p \in G\}$, and*
7. *the* output places of an activity *a consist of the set $OP(a) = \{p \mid$ for some $c = 1, 2, \ldots, \gamma(a),\ \exists\ (G, f) \in o^{-1}(a, c)$ such that $p \in G\}$.*

The marking of a network can alternatively be represented as a vector, in which each component of the vector is the number of tokens in a particular place. The correspondence of components of the vector to markings of places is done via some designated total ordering of P. For example, for a set of places $\{p_1, p_2, \ldots, p_n\} \subseteq P$ and marking vector $(n_1, n_2, \ldots, n_n)$, $\mu(p_1) = n_1, \mu(p_2) = n_2, \ldots, \mu(p_n) = n_n$, if $p_1 < p_2 < \ldots < p_n$. The functional notation for markings is more convenient for the development of theory, while the vector notation is useful for examples.

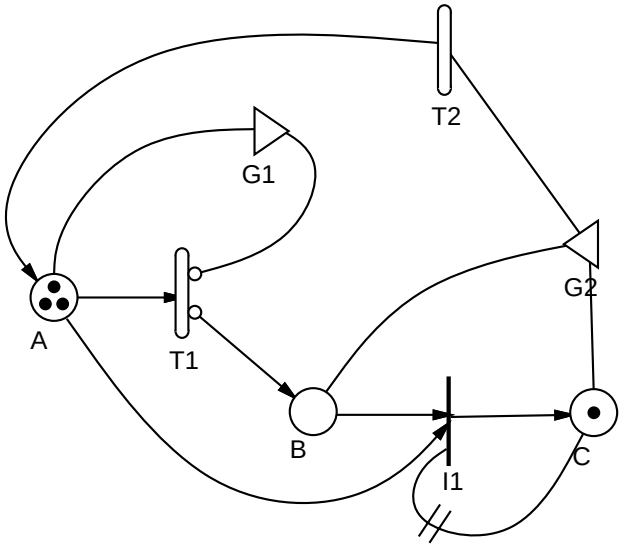
2.2 Graphical Representation

To aid in the modeling process, a graphical representation for activity networks is typically employed. In fact, for all but the smallest networks, specification via the tuple formulation presented in the definition is extremely cumbersome. Not only is the graphical representation more compact, but it also provides greater insight into the behavior of the network. For example, let i and j be natural numbers, and consider the following activity network:

$$
\begin{aligned}
P &= \{A, B, C\}, \text{ where } A < B < C \\
A &= \{T1, T2, I1\} \\
I &= \{GA1, GA2, GB, GC, G2\} \\
O &= \{AG, BG, CG, G1\} \\
\gamma &= \{(T1, 2),\ (T2, 1),\ (I1, 1)\} \\
\tau &= \{(T1, Timed),\ (T2, Timed),\ (I1, Instantaneous)\} \\
GA1 &= (\{A\},\ \{(i, 1) \mid i > 0\} \cup \{(0, 0)\},\ \{(i, i-1) \mid i > 0\} \cup \{(0, 0)\}) \\
GA2 &= (\{A\},\ \{(i, 1) \mid i > 0\} \cup \{(0, 0)\},\ \{(i, i-1) \mid i > 0\} \cup \{(0, 0)\}) \\
GB &= (\{B\},\ \{(i, 1) \mid i > 0\} \cup \{(0, 0)\},\ \{(i, i-1) \mid i > 0\} \cup \{(0, 0)\}) \\
GC &= (\{C\},\ \{(i, 0) \mid i > 0\} \cup \{(0, 1)\},\ \{(i, i) \mid \forall i\}) \\
G2 &= (\{B, C\},\ \{((i, j), 1) \mid i > 0 \text{ or } j > 0\} \cup \\
&\qquad \{(0, 0), 0)\},\ \{((i, j), (0, 0)) \mid \forall i, j\}) \\
AG &= (\{A\},\ \{(i, i+1) \mid \forall i\}) \\
BG &= (\{B\},\ \{(i, i+1) \mid \forall i\}) \\
CG &= (\{C\},\ \{(i, i+1) \mid \forall i\}) \\
G1 &= (\{A, B\},\ \{((i, j), (i+2, j)) \mid i < 5 \text{ and } j < 5\} \cup \\
&\qquad \{((i, j), (i-1, j)) \mid j > 5 \text{ and } i = 0\} \cup \\
&\qquad \{((i, j), (i, j)) \mid j > 5 \text{ and } i \neq 0\}) \\
\iota &= \{(GA1, T1),\ (GA2, I1),\ (GB, I1),\ (GC, I1),\ (G2, T2)\} \\
o &= \{(AG, (T2, 1)),\ (BG, (T1, 2)),\ (CG, (I1, 1)),\ (G1, (T1, 1))\}
\end{aligned}
$$

Figure 1 depicts the graphical representation of this network.

One can immediately see the utility of a graphical representation. Here places are represented by circles (A, B, and C), as in Petri nets. Timed activities ($T1$ and $T2$) are represented as hollow ovals. Instantaneous activities ($I1$) are represented by solid bars. Cases associated with an activity are represented by small circles on one side of the activity (as on $T1$). An activity with only one case is represented with no circles on the output side (as on $T2$).



Gate	Predicate	Function
G1	-	if (MARK(A) < 5 and MARK(B) < 5) then MARK(A) = MARK(A) + 2; else if (MARK(A) > 0) then MARK(A) = MARK(A) - 1;
G2	MARK(B)>0 or MARK(C)>0	MARK(B) = 0; MARK(C) = 0;

Fig. 1. Graphical Activity Network Representation

Gates are represented by triangles. $G2$ is an example of an input gate with 2 inputs. $G1$ is an example of an output gate with 2 outputs. Enabling predicates and functions for gates are typically given in tabular form. Three types of commonly used gates are given default (non-triangle) representations for ease of interpretation and to illustrate their similarity to classical Petri net primitives. In particular, an input gate with one input, enabling predicate $\{(i,1) \mid i > 0\} \cup \{(0,0)\}$, and function $\{(i,i-1) \mid i > 0\} \cup \{(0,0)\}$ (e.g., $GA1$) is represented as a directed line from its input to its output. Similarly, an output gate with one output and output function $\{(i,i+1) \mid \forall i\}$ (e.g., AG) is shown as a directed line from its input to its output. Finally, an input gate with one input, enabling predicate $\{(i,0) \mid i > 0\} \cup \{(0,1)\}$, and function $\{(i,i) \mid \forall i\}$ (e.g., GC) is shown as a directed line from its input to its output crossed by two short parallel lines. This type of input gate corresponds to an inhibitor arc in extended Petri nets. These shorthand notations for gates help the viewer understand the behavior of a network from its graphical representation.

2.3 Activity Network Behavior

The behavior of an activity network is a characterization of the possible completions of activities, selection of cases, and changes in markings. Specifically,

Definition 3 *An activity a* may complete *in a marking μ if*

1. *a is enabled in μ, and*
2. *if a is timed, no instantaneous activities are enabled in μ.*

This imposes two explicit priority classes on activities. We can now define the result of the completion of an activity and selection of a possible case. This is made easier by expanding the domain and range of the gate functions to the complete network marking. Specifically, for an activity network with places P, a gate (of the network) with set of places G, and a function f, define the function $\tilde{f} : M_P \to M_P$ where if $\tilde{f}(\mu) = \mu'$ then

$$\mu'(p) = \begin{cases} f(\mu_G)(p) \text{ if } p \in G \\ \mu(p) \qquad \text{otherwise.} \end{cases}$$

Using this notion, we can define an activity completion and case selection.

Definition 4 *Given an activity network $AN = (P, A, I, O, \gamma, \tau, \iota, o)$ with activity a that may complete in μ, the* completion of activity a and choice of case c in μ yields

$$\mu' = \tilde{f}_{O_m}(\cdots \tilde{f}_{O_1}(\tilde{f}_{I_n}(\cdots \tilde{f}_{I_1}(\mu) \cdots)) \cdots)$$

where $\iota^{-1}(a) = \{I_1, \ldots, I_n\}$ and $o^{-1}(a, c) = \{O_1, \ldots, O_m\}$.

While the gates in the two sets are numbered, there is no implied ordering of their application within a set, since the SAN definition does not specify an ordering among input gates or output gates. Output gate functions are applied after input gate functions, however. The notation $\mu \xrightarrow{a,c} \mu'$ is used to indicate that the completion of a and choice of c *yields* μ'. Furthermore, we say that marking μ' is *immediately reachable* from a marking μ if $\mu \xrightarrow{a,c} \mu'$ for some activity a and case c.

The set of reachable markings from a given marking can be defined directly in terms of the reflexive, transitive closure of the yields relation, which we denote as $\xrightarrow{*}$. Using this notation, the set of reachable markings from some marking can be defined as follows.

Definition 5 *The set of* reachable markings *of an activity network AN in a marking μ_0 is the set of markings $R(AN, \mu_0)$ where*

$$R(AN, \mu_0) = \{\mu \mid \mu_0 \xrightarrow{*} \mu\}.$$

Sets of "stable reachable markings" and "unstable reachable markings" of an activity network can then be defined in terms of its reachable markings. Specifically, the set of *stable reachable markings* of an activity network AN in an initial marking μ_0 is the set $SR(AN, \mu_0) \subseteq R(AN, \mu_0)$ of reachable markings of AN from μ_0 that are stable. Similarly, the set of *unstable reachable markings*, denoted $UR(AN, \mu_0)$, is the set of markings reachable from μ_0 that are not stable.

The behavior of an activity network can be described in terms of successive applications of the yields relation. Each application of the yields relation represents the completion of one of the one or more activities that may complete in the marking. Note that, unlike elementary net systems [11], the yields relation is defined only for single activities and that the concurrent completion of more than one activity is not considered. Each step in the evolution of the network is called a *configuration*, which, formally, is a marking-activity-case triple $< \mu, a, c >$ where a is some activity with case c that may complete in μ. A *completion of a configuration* occurs when the activity associated with the configuration completes. The behavior of a network can thus be described in terms of possible sequences of configurations, more formally called *paths*.

Definition 6 *A* path *of an activity network, AN, with marking μ_0 is a sequence of configurations $\langle \mu_1, a_1, c_1 \rangle, \langle \mu_2, a_2, c_2 \rangle, \ldots, \langle \mu_n, a_n, c_n \rangle$ such that,*

1. $\mu_1 \in R(AN, \mu_0)$,
2. *for each pair of configurations $\langle \mu_i, a_i, c_i \rangle$, $\langle \mu_{i+1}, a_{i+1}, c_{i+1} \rangle$ $(1 \leq i < n)$, $\mu_i \overset{a_i, c_i}{\rightarrow} \mu_{i+1}$, and*
3. $\mu_n \overset{a_n, c_n}{\rightarrow} \mu'$ *for some marking μ'.*

Definition of several additional terms will aid in the discussion that follows. In particular, the *initial marking of a path* is the marking of the first configuration in the path. The *resulting marking of a path* is the marking that is reached upon completion of the last configuration in the path. A path is said to be *from μ to μ'* if μ is the initial marking of the path and μ' is the resulting marking of the path.

3 Stochastic Activity Networks

Activity networks are interesting in their own right, and several of their properties have been studied [12]. However, for the purpose of this chapter, they serve as a non-probabilistic base for a stochastic extension, called *stochastic activity networks*, that is used for performability evaluation. When they are used in this manner, care must be taken to insure that the probabilistic behavior of the stochastic extension is completely specified. Specifically, since we want to be able to ask questions regarding possible sequences of timed activity completions and intervening stable markings, we require that a stable marking eventually be reached after any sequence of consecutive instantaneous activity completions. Identification of situations in which this may occur is aided by the introduction of the notion of a *step*.

Definition 7 *Let AN be an activity network and s be a path of AN with initial marking μ_0. Then s is a step if:*

1. *the initial marking of s is stable, and*

2. the markings of all other configurations of s are unstable.

Note that the resulting marking of the step is not required to be stable. The set of markings that can be reached by completion of different steps from a single initial marking provides insight into the behavior of an activity network. To see this, let

$$S(\mu) = \{s \mid s \text{ is a step with initial marking } \mu\}$$

where μ is a stable reachable marking of the AN in question. Now, since there is only a finite number of steps of a given length from any marking μ, the cardinality of $S(\mu)$ is $\aleph_0$ if and only if the length of steps in $S(\mu)$ increases without bound. Or, equivalently, since all of the activities except the first in a step are instantaneous, $|S(\mu)| = \aleph_0$ if and only if an unbounded number of instantaneous activities can complete without resulting in a stable marking. This leads us to the following definition of a "stabilizing" activity network. Formally,

Definition 8 *An activity network AN in a marking μ_0 is* stabilizing *if, for every $\mu \in SR(AN, \mu_0)$, the set $S(\mu)$ is finite.*

The following example illustrates the concept of stabilizing and non-stabilizing activity networks in a marking. Consider the activity network of Figure 2. If we denote its marking as a vector using the usual lexicographic ordering of place

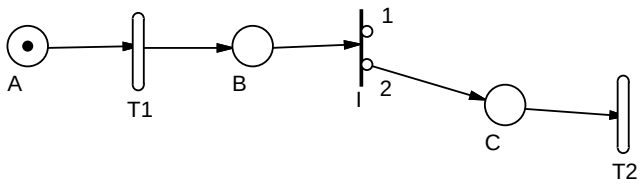


Fig. 2. A Stabilizing Activity Network

names, then the set of steps associated with marking 100, i.e., the set $S(100)$, is

$$\{\langle 100, T1, 1\rangle\langle 010, I1, 1\rangle,\ \langle 100, T1, 1\rangle\langle 010, I1, 2\rangle\}.$$

Similarly, $S(001) = \{\langle 001, T2, 1\rangle\}$. These two markings are the only stable markings reachable from the pictured initial marking. Since both $S(100)$ and $S(001)$ are finite, the activity network is stabilizing. Now consider the activity network of Figure 3. For this network, $S(100)$ is the countably infinite set

$$\left\{ \begin{array}{l} \langle 100, T1, 1\rangle\langle 010, I1, 2\rangle, \\ \langle 100, T1, 1\rangle\langle 010, I1, 1\rangle, \\ \langle 100, T1, 1\rangle\langle 010, I1, 1\rangle\langle 010, I1, 1\rangle, \\ \langle 100, T1, 1\rangle\langle 010, I1, 1\rangle\langle 010, I1, 1\rangle\langle 010, I1, 1\rangle, \\ \vdots \end{array} \right\}.$$

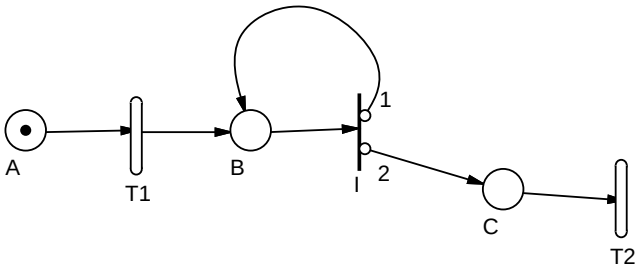


Fig. 3. An Activity Network that is Not Stabilizing

Thus the activity network of Figure 3 is not stabilizing.

Generally, it is not decidable whether an activity network in a marking μ_0 is stabilizing. To see this, recall that it can be shown (see [13], for example) that extended Petri nets (Petri nets with inhibitor arcs) are equivalent, computationally, to Turing machines. The proof of this fact is by construction. Specifically, it can be shown that any register machine can be converted into an equivalent extended Petri net. For this reason, the languages generated by the net can be taken to be the set of sequences of transitions that lead to a reachable marking. Given this equivalence, it is evident that activity networks are equivalent to Turing machines, since every extended Petri net is an activity network (transitions map to activities, places to places, and arcs to gates). In the context of an activity network, the language generated can be taken to be the set of steps with initial marking μ_0 (i.e., $S(\mu_0)$). Thus the class of languages generated by the set of possible activity networks is coextensive with the class of recursively enumerable sets. Since, generally, it is not possible to decide whether a recursively enumerable set is finite [14], we have the following theorem.

Theorem 1 *It is not decidable whether an activity network in a marking μ_0 is stabilizing.*

There are, however, sufficient conditions by which the stabilizing property can be established, based on the structural properties and configuration of the instantaneous activities in the network. Identification of conditions is aided by the introduction of two properties of instantaneous activities. Specifically,

Definition 9 *An instantaneous activity is* self-disabling *if, given any reachable marking μ, it can only complete a finite number of times before any other activities complete.*

This definition allows us to identify activities that can only complete a bounded number of times before the markings of their input places change because of other activities. While this may be a difficult condition to check generally, it is easy to identify several frequently used activity-gate pairs that are self-disabling. For example, an activity with disjoint sets of input and output places and only

default input gates (denoted by directed arcs) is self-disabling. In order to identify those activities that have no potentially unstabilizing interactions with other activities, we introduce the notion of a *cycle-free* instantaneous activity.

Definition 10 *An instantaneous activity I_1 is* cycle-free *if there does not exist a sequence of instantaneous activities $I_1, I_2, \ldots, I_n$ such that*

$$\begin{aligned} &OP(I_1) \cap IP(I_2) \neq \emptyset \ \ \wedge \\ &OP(I_2) \cap IP(I_3) \neq \emptyset \ \ \wedge \\ &\quad \vdots \\ &OP(I_n) \cap IP(I_1) \neq \emptyset. \end{aligned}$$

Informally, then, an instantaneous activity is cycle-free if there does not exist a sequence of instantaneous activities that, through their completions, can affect the original activity's input places. Note that this is a purely structural condition and that even if an activity is not cycle-free, the enabling predicates of the activities in the cycle may be such that the cycle of completions can never occur. Furthermore, because the number of activities in a network is finite by definition, an algorithm exists to determine whether an activity is cycle-free. These two concepts provide us with criteria to identify sufficient conditions for an activity network to be stabilizing. Intuitively, if every instantaneous activity is cycle-free and self-disabling, then no instantaneous activity can complete an unbounded number of times before the completion of a timed activity intervenes. More formally, we have the following theorem.

Theorem 2 *An activity network AN in a marking μ is stabilizing if every instantaneous activity of the network is cycle-free and self-disabling.*

Proof:

The proof is by contradiction. Suppose there exists an activity network AN with activities A that is not stabilizing in a marking μ, but in which every instantaneous activity of the network is cycle-free and self-disabling. Then, by definition, since AN is not stabilizing there exists an activity that can complete an unbounded number of times without resulting in a stable marking. A self-disabling activity can only complete an unbounded number of times without reaching a stable marking if another instantaneous activity changes the marking of one of its input places. But every instantaneous activity in A is cycle-free, so that can not occur. Thus an activity network in a marking μ is stabilizing if every instantaneous activity in the network is cycle-free and self-disabling. □

For an example of an activity network that satisfies the conditions of the above theorem, see Figure 4. First, note that all the instantaneous activities in the network are self-disabling, since they all have default input gates and disjoint input and output places. Secondly, note that all instantaneous activities are cycle-free. Thus by Theorem 2 this activity network is stabilizing. The stability of an activity network is an important necessary condition to insure that the

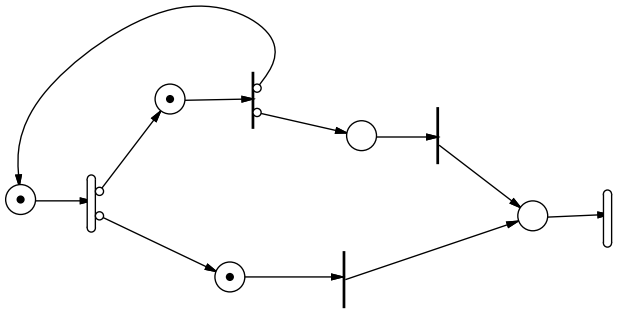


Fig. 4. A Second Stabilizing Activity Network

probabilistic behavior of its stochastic extension is completely specified. Before the second necessary condition can be discussed, the definition of a stochastic activity network must be given.

3.1 Definition of a Stochastic Activity Network

Given an activity network that is stabilizing in some specified initial marking, a stochastic activity network is formed by adjoining functions C, F, and G, where C specifies the probability distribution of case selections, F represents the probability distribution functions of activity delay times, and G describes the sets of "reactivation markings" for each possible marking. Formally,

Definition 11 *A* stochastic activity network *(SAN) is a five-tuple*

$$SAN = (AN, \mu_0, C, F, G)$$

where:

1. $AN = (P, A, I, O, \gamma, \tau, \iota, o)$ *is an activity network.*
2. $\mu_0 \in M_P$ *is the* initial marking *and is a stable marking in which AN is stabilizing.*
3. C *is the* case distribution assignment*, an assignment of functions to activities such that for any activity* a*,* $C_a : M_{IP(a) \cup OP(a)} \times \{1, \ldots, \gamma(a)\} \rightarrow [0, 1]$*. Furthermore, for any activity* a *and marking* $\mu \in M_{IP(a) \cup OP(a)}$ *in which* a *is enabled,* $C_a(\mu, \cdot)$ *is a probability distribution called the* case distribution of a in μ*.*
4. F *is the* activity time distribution function assignment*, an assignment of continuous functions to timed activities such that for any timed activity* a*,* $F_a : M_P \times \mathbb{R} \rightarrow [0, 1]$*. Furthermore, for any stable marking* $\mu \in M_P$ *and*

timed activity a that is enabled in μ, $F_a(\mu, \cdot)$ is a continuous probability distribution function called the activity time distribution function of a in μ*; $F_a(\mu, \tau) = 0$ if $\tau \leq 0$.*

5. *G is the* reactivation function assignment*, an assignment of functions to timed activities such that for any timed activity a, $G_a : M_P \rightarrow \wp(M_P)$, where $\wp(M_P)$ denotes the power set of M_P. Furthermore, for any stable marking $\mu \in M_P$ and timed activity a that is enabled in μ, $G_a(\mu, \cdot)$ is a set of markings called the* reactivation markings *of a in μ.*

Recall that an activity is enabled if all of its input gates hold. While this concept is sufficient to describe the behavior of an activity network, several additional terms are needed to describe the behavior of a stochastic activity network. Figure 5 aids in the description of these terms. In particular, since timed activities represent operations in a modeled system, events must be defined to denote the start and finish of these operations. The start of an operation is signaled by an *activation* of an activity. An activation of an activity will occur if 1) the activity becomes enabled (illustrated by the first time-line) or 2) the activity completes and is still enabled (illustrated by the second time-line). Some time after an activity is activated it will either *complete* (illustrated by the first time-line) or be *aborted* (illustrated by the third time-line). The activity will complete if it remains enabled throughout its activity time (which will be defined momentarily); otherwise it is aborted.

The activity time distribution function specifies (probabilistically) the *activity time* of an activity, i.e., the time between its activation and *completion*. Any continuous distribution (e.g., exponential or normal) is a legal activity time distribution, although the choice of distribution will affect the applicability of many solution methods. Both the distribution type and its parameters can depend on the global marking of the network at the activation time of the activity. Activity times are assumed to be mutually independent random variables.

Two other functions are associated with an activity network to form a SAN. In particular, the case distribution specifies (probabilistically) which case is to be chosen upon the completion of an activity. These probabilities can depend on the markings of the input and output places of the activity at its completion time. A reactivation function is also associated with each timed activity. This function specifies, for each marking, a set of *reactivation markings.* Given that an activity is activated in a specific marking, it is reactivated (i.e., activated again) whenever any marking in the set of reactivation markings for the activation marking is reached before the activity completes (as illustrated by the fourth time-line). Probabilistically, the reactivation of an activity is exactly the same as an activation; a new activity time distribution is selected based on the current marking. This provides a mechanism for restarting activities that have been activated, either with the same or a different distribution. This decision is made on a per-activity basis (based on the reactivation function) and, hence, is not a net-wide execution policy.

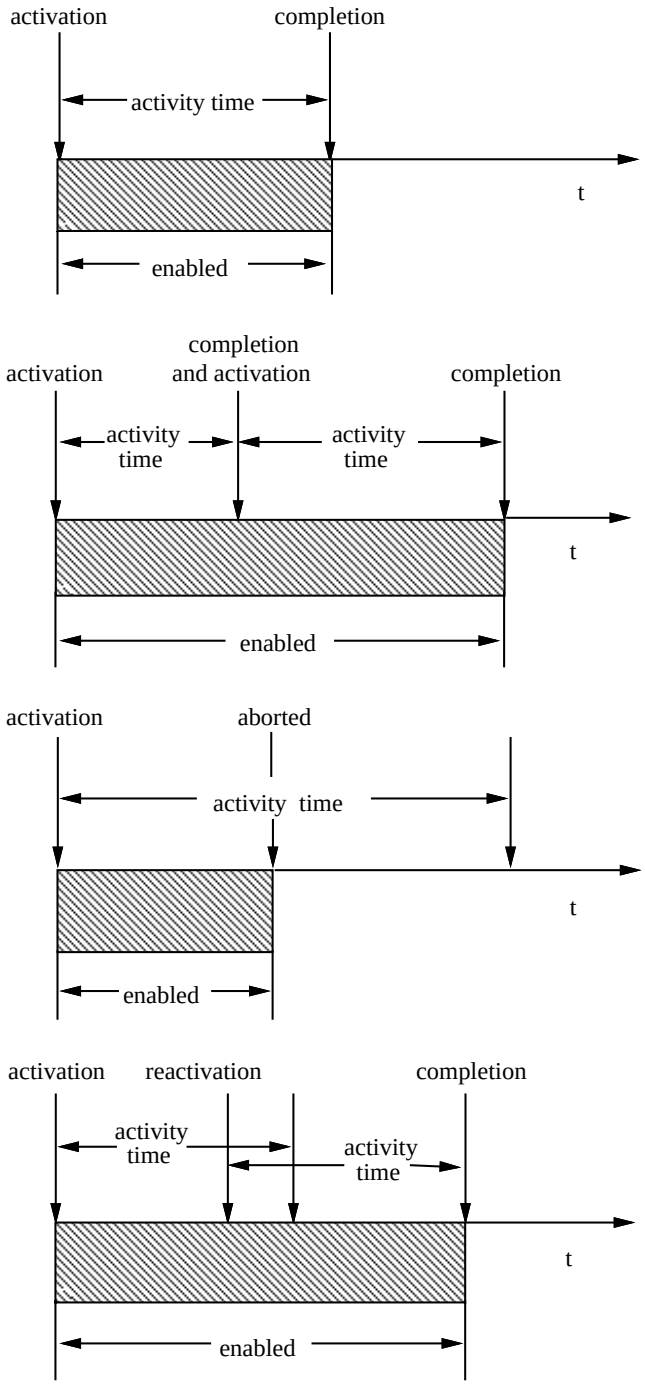


Fig. 5. Terms Related to the Execution of a Timed Activity

The definition of a stochastic activity network presented here differs from that presented earlier in [9] in three respects. First, probabilities associated with cases are required to depend only on input and output places of the associated activity. This requirement permits the development of more efficient algorithms to test whether the probabilistic behavior of a stochastic activity network is completely specified. Since any place in a network can be made to be an input or output place of any activity, this is not a restrictive requirement.

Second, the new definition requires that the probability distribution function associated with each timed activity in a possible marking be continuous. This requirement ensures that two timed activities do not complete at the same time, since the behavior when this occurs is not specified. This requirement is not overly strict for stochastic activity networks that are to be solved analytically, since the solution method usually places stricter constraints on the distributions. In the case of SANs that are to be solved via simulation, the ambiguity can be avoided by assigning an ordering to timed activities that may complete at the same time.

Third, the current definition does not require that the underlying activity network be "well-behaved" [9] in its initial marking. An activity network is said to be *well-behaved* in an initial marking μ if

1. No infinite sequence of instantaneous activities can complete in any marking reachable from marking μ, and
2. If a marking reachable from μ has more than one enabled instantaneous activity, then, from that marking, all possible sequences of reachable markings result in the same stable marking.

The requirement that the underlying activity network be well-behaved is more strict than the requirement that the underlying AN be stabilizing, and does lead to stochastic activity networks whose probabilistic behavior is completely specified. However, delaying the introduction of conditions of this type until after the stochastic extension is defined allows us to identify a larger class of networks whose probabilistic behavior is completely specified.

3.2 Stochastic Activity Network Behavior

Before discussing in detail how activity time distributions, case distributions, and reactivation functions determine an activity network's behavior, it helps to describe, informally, how a network executes in time. In particular, one can think of an execution of a SAN as a sequence of configurations, where for each configuration $\langle \mu, a, c \rangle$ the SAN was in marking μ, activity a completed, and case c was chosen. In any marking μ, the activity that completes is selected from the set of *active* activities in μ, i.e., the set of those activities that have been activated but have not yet completed or aborted. After each activity completion

and case selection, the set of activities that are active is altered as follows. If the marking reached (as specified by the yields relation) is stable, then

1. the activity that completed is removed from the set of active activities,
2. activities that are no longer enabled in the reached marking are removed from the set of active activities,
3. activities for which the reached marking is a reactivation marking are removed from the set of active activities, and
4. activities that are enabled but are not in the set of active activities are placed in it (including those that were reactivated).

In contrast, if the marking reached is not stable, then timed activities (other than the one that just completed, if it is timed) are not added or deleted from the set. Instead,

1. the activity that completed is removed,
2. instantaneous activities that are no longer enabled in the reached marking are removed, and
3. instantaneous activities that are enabled but not in the set are added.

The choice of the activity to complete from the set of active activities is determined by the activity time distribution function of each activity in the set and the relative priority of timed and instantaneous activities (as specified by the definition of "may complete"). If there are one or more instantaneous activities in the set, one of them is chosen (non-deterministically) to complete. If there are none, the timed activity with the earliest completion time is selected (stochastically) based on the activity times of the set of active activities. Recall that the activity time distribution function defines the time between an activity's activation and completion. After the activity to complete is selected, a case of the activity is chosen based on its case distribution in the current marking, and a new marking is reached. The set of active activities is "initialized" at the start of an execution by adding to the set all those activities that are enabled in the initial marking. Note that, because choices between active instantaneous activities are made non-deterministically but not probabilistically, there may be networks for which these choices lead to behaviors that are not probabilistically specified. We now investigate conditions under which probabilistic behavior of the network is completely specified.

3.3 Well-Specified Stochastic Activity Networks

This section provides the basic concepts and ideas that define when a stochastic activity network's behavior is completely probabilistically specified. Since the time this material first appeared [15], further work has been done to develop

algorithms that are tailored to specific reward variables [16], and are more efficient [17]. In addition, [18] presents a similar concept in the context of stochastic reward nets. These works present newer and more efficient algorithms to determine whether a net is well-specified; in contrast, this section focuses on the concept itself and how the structure of a SAN can be used to reduce the complexity of determining whether a SAN is well-specified.

Two types of nondeterminism can occur in stochastic activity networks: 1) uncertainty as to which activity will complete among the active activities, and 2) uncertainty as to which case of the activity that complete will be chosen. To aid in the discussion that follows, we will refer to a choice of the first type as an *activity choice* and a choice of the second type as a *case choice*.

In stochastic activity networks, case choices are quantified by the assignment of a case distribution to each activity. Activity choices are quantified partially by the assignment of an activity time distribution to each timed activity. However, the activity time distribution does not completely quantify this type of non-determinism, since the behavior is not defined if two activities have the same completion time. That would never occur for two timed activities, since all activity time distributions are continuous. It will occur, however, whenever two or more instantaneous activities are enabled, since instantaneous activities complete in zero time. In this situation, the choice of which activity will complete next is non-deterministic and not quantified by either the activity time distribution function assignment or the case distribution assignment.

Since we are interested in possible sequences of timed activity completions and reached stable markings, we would like the probability distribution on the choice of the next stable marking to be the same regardless of any non-probabilistically quantified activity choices that have been made. To investigate this more formally, we introduce the notion of a *stable step*.

Definition 12 *Let $S = (AN, \mu_0, C, F, G)$ be a stochastic activity network and s be a step of AN with initial marking μ_0. Then, s is a* stable step *if the resulting marking of s is stable.*

A stable step can be thought of as a "jump" in the execution of a stochastic activity network that takes the network from one stable marking to another via the completion of a timed activity and zero or more instantaneous activities. There may be several steps with the same first marking and activity, but a different final marking. Accordingly, we define the "set of next stable markings" for a stable marking upon completion of an activity a as follows.

Definition 13 *Let $S = (AN, \mu_0, C, F, G)$ be a stochastic activity network and $\mu \in SR(AN, \mu_0)$. The set of* next stable markings *for S in μ upon completion of a is the set*

$$NS(\mu, a) = \left\{ \mu' \;\middle|\; \begin{array}{l} \exists \text{ a stable step } s \text{ from } \mu \text{ to } \mu' \text{ such that} \\ a \text{ is the activity of the first configuration of } s \end{array} \right\}.$$

This set allows us to focus our attention on those stable markings that may be reached from a particular stable marking by completion of a specific timed activity. In order to insure that the probability distribution over the next stable markings is invariant over activity choices between instantaneous activities, we must define a measure on stable steps that captures the probability that a stable step will be taken given that a set of activity choices is made in a manner such that the step is possible. The case construct allows us to define this probability. Specifically,

Definition 14 *Let* $S = (AN, \mu_0, C, F, G)$ *be a SAN and let* s *be a path of* S *such that* $s = \langle \mu^1, a^1, c^1 \rangle \langle \mu^2, a^2, c^2 \rangle \cdots \langle \mu^n, a^n, c^n \rangle$. *Then,* the probability of s, *denoted* $Pr(s)$, *is*

$$C_{a^1}(\mu^1_{IP(a^1) \cup OP(a^1)}, c^1) \times C_{a^2}(\mu^2_{IP(a^2) \cup OP(a^2)}, c^2) \ldots \times C_{a^n}(\mu^n_{IP(a^n) \cup OP(a^n)}, c^n)$$

where $\times$ *is taken to be normal multiplication on the set of real numbers.*

This function defines the probability that a stable step will be taken given that a set of activity choices was made such that the step may occur. Since we want to insure that the probability distribution over the next stable markings upon completion of a timed activity is invariant over possible sets of activity choices, we consider the set of steps from some stable marking μ to a stable marking $\mu' \in NS(\mu, a)$, by completion of a timed activity a that may complete in μ. Formally, let

$$P^a_{\mu,\mu'} = \{ s \mid s \text{ is a stable step from } \mu \text{ to } \mu' \text{ with first activity } a \}.$$

Different stable steps in this set can arise from different sets of activity choices. In order to specify the set of stable steps from one marking to another upon completion of some timed activity for a single set of activity choices, it helps to define a relation relating stable steps that can occur under a single set of activity choices. Specifically, define the relation R on $P^a_{\mu,\mu'}$ to be

$$R = \left\{ (s, s') \;\middle|\; \begin{array}{l} \text{for every configuration } \langle \mu, a, c \rangle \text{ in } s \text{ and configuration} \\ \langle \mu', a', c' \rangle \text{ in } s' \text{ such that } \mu = \mu',\ a = a' \end{array} \right\}.$$

Thus, two stable steps are related if, for every marking they share in common, the same activity choice is made. Note that while R is not an equivalence relation, it is a compatibility relation (i.e., it is reflexive and symmetric). While a compatibility relation does not necessarily define a partition of a set, it does define a covering of a set, by the *maximal compatibility classes* of the relation. Recall that (as in [19]) a subset $C \subseteq P^a_{\mu,\mu'}$ is called a *maximal compatibility class* if every element of C is related to every other element of C and no element of $P^a_{\mu,\mu'} - C$ is related to all the elements of C. Each maximal compatibility class contains stable steps that correspond to a single set of activity choices. More specifically, note that all stable steps in C correspond to steps from μ to μ' by completion of timed activity a under some set of activity choices. The probability that μ' is reached from μ by completion of a, given that a particular set of

activity choices has been made, is thus the sum of the probabilities of all stable steps in C, i.e.,

$$P(C) = \sum_{s \in C} Pr(s).$$

All activity choices within stable steps correspond to choices between active instantaneous activities and, hence, are not probabilistically specified. Therefore, for a stochastic activity network to be completely probabilistically specified, $P(C)$ must be the same for all maximal compatibility classes C. To express this precisely, we introduce the notion of a *well-specified stochastic activity network*.

Definition 15 *A stochastic activity network* $S = (AN, \mu_0, C, F, G)$ *is* well-specified *if, for every marking* $\mu \in SR(AN, \mu_0)$*, each activity* a *that may complete in* μ*, and all* $\mu' \in NS(\mu, a)$*,* $P(C)$ *is identical for all maximal compatibility classes* C *of* R *defined on* $P^a_{\mu,\mu'}$*.*

The above definition identifies a class of networks whose behavior is completely specified, probabilistically, with respect to all notions of state that we intend to consider.

It is interesting to compare the notion just presented to the "well-behaved" notion used previously. In particular, one can show that every activity network well-behaved in some marking μ_0 is well-specified for any choice of activity time distributions, reactivation functions, and case probabilities. We state this fact as a theorem.

Theorem 3 *If an activity network is well-behaved in a marking* μ_0*, then it is well-specified for any choice of activity time distributions, reactivation functions, and case distributions.*

Proof:

Suppose an activity network AN is well-behaved in a marking μ_0. Then, for every marking μ reachable from μ_0, no infinite sequence of activities can complete in μ. Thus AN is stabilizing in μ_0. Augment AN with arbitrary activity distributions, reactivation functions, case distributions, and initial marking μ_0 to form a SAN. Now, recall that this SAN is well-specified if for every stable marking μ reachable from μ_0, each activity a that may complete in μ, and each $\mu' \in NS(\mu, a)$, $P(C)$ is identical for all maximal compatibility classes C or R defined on $P^a_{\mu,\mu'}$. Consider an arbitrary stable marking μ reachable from μ_0 and activity a that may complete in μ. Since S is well-behaved, one of three situations may arise upon completion of a in μ.

In the first situation, all markings in the set of next possible markings are stable. There is thus only one maximal compatibility class for this marking and activity, and the criterion of Definition 15 is satisfied.

In the second situation, at least one marking in the set of next possible markings is unstable, and all possible unstable markings that may be reached before an-

other stable marking is reached have at most one instantaneous activity enabled. Then, as in the first situation, there is only one maximal compatibility class for this marking and activity, and therefore the criterion of Definition 15 is satisfied.

In the third situation, at least one marking in the set of next possible markings is unstable, and some unstable marking that may be reached before another stable marking is reached that has two or more instantaneous activities enabled. But, since S is well-behaved, all possible sequences of markings reachable from that marking will result in the same next stable marking. Thus, while there may be more than one compatibility class for the marking and activity, they all result in the same single marking with a probability of one, and hence, $P(C)$ is identical for all maximal compatibility classes. Again, the criterion of Definition 15 is satisfied for this marking and activity.

Since the criterion of the definition is satisfied for each possible situation for every reachable marking and activity that may complete in the marking, the SAN is well-specified for any choice of activity time distributions, reactivations, and case distributions. □

It should be noted, however, that the converse of 3 is not a theorem, and hence that well-specified SANs are more general than well-behaved SANs. To see this, consider the stochastic activity network of Figure 6. This SAN is well-specified

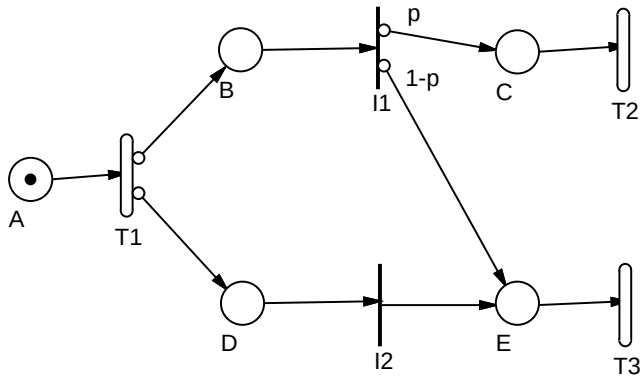


Fig. 6. A Well-Specified, but not Well-Behaved, Stochastic Activity Network

in the pictured marking, since the activity choice that is made after completion of the enabled timed activity does not affect the distribution of the next stable markings. It is not well-behaved, though, since two instantaneous activities may be enabled and more than one stable marking can be reached from the current marking. Recall that in order for an activity network to be well-behaved, whenever a reached marking has two or more instantaneous activities

enabled, all possible sequences of reachable markings must result in the same stable marking.

Any algorithm to determine whether a given stochastic activity network is well-specified must check that the probability distribution over each next stable markings does not depend on the set of activity choices that is made. This condition can be checked using techniques developed to find the set of all maximal compatibles [20]. The following algorithm checks this for a particular stable marking and timed activity.

Algorithm 1 *(Determines whether the next stable marking probability distribution is invariant over possible sets of activity choices for a stable marking μ and activity a that can complete in μ, and if it is, computes this distribution.)*

Compute the set of all stable steps for which the initial marking is μ and timed activity is a.
Group the set of stable steps computed above according to the resulting marking of each step. The subset containing the stable steps from μ to μ' by completion of timed activity a is denoted by $P^a_{\mu,\mu'}$.
For each subset $P^a_{\mu,\mu'}$:
    Construct the set of maximal compatibles of R on $P^a_{\mu,\mu'}$.
    Compute $P(C)$ for each maximal compatible C.
    If $P(C)$ is not identical for all compatibles C, then
        Signal SAN is not well-specified and abort algorithm.
Next $P^a_{\mu,\mu'}$.
Return next stable marking probability distribution for marking and activity.

For convenience, we denote this distribution by the function $h_{\mu,a} : NS(\mu, a) \rightarrow [0, 1]$, where for $\mu' \in NS(\mu, a)$, $h_{\mu,a}(\mu')$ is the probability that μ' will be reached given that the SAN is in μ and a completes.

The following example illustrates the use of Algorithm 1. Specifically, consider the stochastic activity network of Figure 7. Here the case distribution for each activity is denoted by the number next to each case for the activity. In addition, markings are denoted by vectors, assuming the usual lexicographic ordering of place names. With these facts in mind, we will show (using Algorithm 1) that the next stable marking probability distribution is invariant over all possible sets of activity choices for the initial marking 1000000 (lexicographic ordering on place names) and timed activity $T1$. The algorithm first computes the set of all stable steps, which is shown in Figure 8. These steps are then used to determine the set of possible next stable markings ($NS(1000000, T1)$), which is found to be $\{0001010, 0000011\}$. The set of stable steps is then split into two subsets, according to their resulting markings. These two subsets serve as input to the portion of the algorithm that computes sets of maximal compatibles, checks to see that the probability measure is invariant over all possible compatibles, and computes the next stable marking probabilities. The algorithm then computes the set of maximal compatibles corresponding to the resulting marking 0001010;

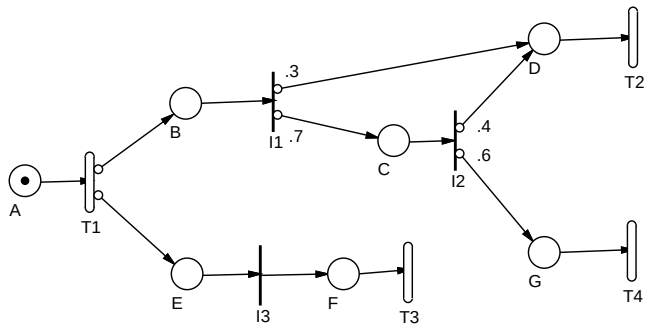


Fig. 7. A Well-Specified Stochastic Activity Network

$$\left\{\begin{array}{l}
\langle 1000000, T1, 1\rangle\langle 0100100, I1, 1\rangle\langle 0001100, I3, 1\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I2, 1\rangle\langle 0001100, I3, 1\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I2, 2\rangle\langle 0000101, I3, 1\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I3, 1\rangle\langle 0010010, I2, 1\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I3, 1\rangle\langle 0010010, I2, 2\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I3, 1\rangle\langle 0100010, I1, 1\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I3, 1\rangle\langle 0100010, I1, 2\rangle\langle 0010010, I2, 1\rangle, \\
\langle 1000000, T1, 1\rangle\langle 0100100, I3, 1\rangle\langle 0100010, I1, 2\rangle\langle 0010010, I2, 2\rangle
\end{array}\right\}$$

Fig. 8. Set of Stable Steps for the Stochastic Activity Network of Figure 2.7 from Marking 10000000 by Completion of Activity $T1$.

the set consist of the three elements

$$C_1 = \{\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I2, 1\rangle\langle 0001100, I3, 1\rangle, \\ \langle 1000000, T1, 1\rangle\langle 0100100, I1, 1\rangle\langle 0001100, I3, 1\rangle\},$$

$$C_2 = \{\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I3, 1\rangle\langle 0010010, I2, 1\rangle, \\ \langle 1000000, T1, 1\rangle\langle 0100100, I1, 1\rangle\langle 0001100, I3, 1\rangle\}, \text{ and}$$

$$C_3 = \{\langle 1000000, T1, 1\rangle\langle 0100100, I3, 1\rangle\langle 0100010, I1, 2\rangle\langle 0010010, I2, 1\rangle, \\ \langle 1000000, T1, 1\rangle\langle 0100100, I3, 1\rangle\langle 0100010, I1, 1\rangle\}.$$

$P(C)$ is then computed for each maximal compatible C, and $P(C_1) = P(C_2) = P(C_3) = .58$. Similarly, computing the set of maximal compatibles of stable steps with resulting marking 0000011, we obtain:

$$C_1 = \{\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I2, 2\rangle\langle 0000101, I3, 1\rangle\}$$
$$C_2 = \{\langle 1000000, T1, 1\rangle\langle 0100100, I1, 2\rangle\langle 0010100, I3, 1\rangle\langle 0010010, I2, 2\rangle\}$$
$$C_3 = \{\langle 1000000, T1, 1\rangle\langle 0100100, I3, 1\rangle\langle 0100010, I1, 2\rangle\langle 0010010, I2, 2\rangle\}$$

For these compatibles, $P(C_1) = P(C_2) = P(C_3) = .42$. Since the probability measure is the same for all the maximal compatibles in a set, for both sets, the next stable marking probability distribution is invariant over the set of possible activity choices for this SAN, starting marking, and timed activity.

By definition, then, a stochastic activity network is well-specified if this distribution is invariant for all stable reachable markings and activities that may complete in these markings. In cases in which the set of stable reachable markings is finite, we define the following algorithm which determines whether a SAN is well-specified.

Algorithm 2 *(Determines whether a SAN with a finite reachability set is well-specified and computes next stable marking probability distributions)*

For each $\mu \in SR(AN, \mu_0)$ and activity a that may complete in μ:
Determine whether the next stable marking probability distribution is invariant over possible sets of activity choices for this choice of μ and a.
If distribution is not invariant for this μ and a, then
Signal SAN is not well-specified and abort algorithm.
Next $\mu \in SR(AN, \mu_0)$ and activity a that may complete in μ.
Signal stochastic activity network is well-specified.

Note that this algorithm is simply an iterative application of Algorithm 1. While Algorithm 1 suffices to determine whether the next stable marking probability distribution is invariant for a particular marking μ and activity a, its performance can be improved upon if information concerning the structure of the network is used. This technique makes use of the concept of *dependent instantaneous activities*. Specifically,

Definition 16 *Let I_1 and I_2 be instantaneous activities of some activity network. Then I_1 and I_2 are* dependent *if*

$$(IP(I_1) \cup OP(I_1)) \cap (IP(I_2) \cup OP(I_2)) \neq \emptyset.$$

Informally, then, two instantaneous activities are dependent if they have common input or output places. Two activities that are dependent can affect each other by changing the markings of each other's input or output places. In order to identify instantaneous activities that can affect each other through a sequence of completions, we look at the transitive closure of a relation based on the above definition. Specifically, let DEP denote a relation on the set of instantaneous activities of a SAN such that $I_1\ DEP\ I_2$ if I_1 and I_2 are dependent. Furthermore, let $\overset{*}{DEP}$ denote the transitive closure of DEP. It is easy to see that $\overset{*}{DEP}$ is an equivalence relation; thus, it partitions the set of instantaneous activities. The blocks of the partition are sets of activities whose order of completion may affect the probability distribution of the next stable markings. On the other hand, pairs of activities from different blocks cannot affect each other by completing (since activities can only change the markings of their input or output places,

and case probabilities depend only on input and output places). This suggests that steps within each subnetwork defined by activities in each block can be considered individually and combined to determine the total probability for a possible next stable marking.

In order to explore this idea in more detail, we define the notion of an instantaneous subnetwork of a SAN constructed from a set of activities.

Definition 17 *Given a stochastic activity network $S = (AN, \mu_0, C, F, G)$, underlying activity network $AN = (P, A, I, O, \gamma, \tau, \iota, o)$, and set of instantaneous activities $A' \subseteq A$, the* instantaneous subnetwork of S with respect to A' *is a structure (M', μ_0', C') where*

1. *$M' = (P', A', I', O', \gamma', \tau', \iota', o')$ is an activity network with*
   (a) *$P' = \{p \mid p \in P$ and $p \in IP(a) \cup OP(a)$ for some $a \in A'\}$,*
   (b) *A' is some specified set of instantaneous activities,*
   (c) *$I' = \{g \mid g \in I$ and $g \in \iota^{-1}(a)$ for some $a \in A'\}$,*
   (d) *$O' = \{g \mid g \in O$ and $g \in o^{-1}(a, c)$ for some $a \in A'$ and $c = 1, 2, \ldots, \gamma(a)\}$, and*
   (e) *γ', τ', ι', and o' are the functions γ, τ, ι, and o, respectively, restricted to P', A', I', and O'.*
2. *$\mu_0' = \mu_0$ restricted to places P', and*
3. *C' is the function C restricted to A'.*

While (M', mu_0', C') does not fit the definition of a stochastic activity network precisely since the initial marking is not stable, it does provide us with a network made up of instantaneous activities in which all the case probabilities are specified. The revised algorithm presumes that such a subnetwork is constructed for each set of activities corresponding to a block of the partition defined by $\overset{*}{DEP}$. By the nature of $\overset{*}{DEP}$, these subnetworks do not interact with one another. This fact is exploited in the revised algorithm presented below.

Unlike Algorithm 1, which computes the set of all stable steps for the given marking and activity immediately, the revised algorithm accomplishes the same goal in two smaller steps. First, it computes the set of "next possible markings" for the given starting marking and activity. The set of next possible markings is the set of markings that can be reached by one application of the yields relation, i.e., for a given marking μ and activity a

$$NP(\mu, a) = \{\mu' \mid \mu \overset{a,c}{\rightarrow} \mu' \text{ for some } a \in A \text{ and } c \in \{1, \ldots, \gamma(a)\}\}.$$

Each of these markings can be either stable or unstable. If a marking is stable, then it is a next stable marking for the specified starting marking and activity. Furthermore, its probability of occurrence is just the sum of the probabilities of

all cases that lead to that marking. Since only probabilistically specified activity choices were made in reaching the marking, the network is well-specified. A more complicated situation exists for each unstable marking in $NP(\mu, a)$.

These markings are those unstable markings that can be reached after one application of the yields relation and, hence, represent situations where one or more instantaneous activities must be completed to reach possible next stable markings. To determine these markings, consider the instantaneous subnetworks previously constructed. First, note that although the sets of places defined by each subnetwork are disjoint, they may not partition the set of places of the entire network, since there may be places that are connected only to timed activities. The markings of these places will not change by completion of instantaneous activities and hence will remain the same in all next stable markings of the network reached from this marking. In the algorithm that follows, the marking of the places connected only to timed activities is denoted by μ'_s, for each $\mu' \in NP(\mu, a)$. Similarly, the initial (unstable) markings of each of the n subnetworks are denoted by $\mu'_1, \mu'_2, \ldots, \mu'_n$, respectively, for each $\mu' \in NP(\mu, a)$. Now, since the markings of places of different subnetworks are independent of each other, sets of next stable markings can be computed for each subnetwork independently and be combined to obtain "global" next stable markings for the entire network.

Computation of the local next stable markings, and the subsequent check that the probabilities these markings are invariant over possible sets of activity choices for each subnetwork, is done in a manner similar to that of Algorithm 1, except that the initial marking of each of the possible paths to a stable marking is not itself stable. Since these paths are suffixes of stable steps, we call them *partial stable steps*. A partial stable step is a stable step without the initial configuration. Except for this difference, the invariant check and computation of probabilities are done exactly as in Algorithm 1. In the algorithm presented below, for each subnetwork i, the set of (subnetwork) next stable markings is denoted by NS_i and the probability that subnetwork marking $\mu''_{i,j}$ will be reached from subnetwork marking μ'_i is denoted by $\hat{h}_{\mu'_i}(\mu''_{i,j})$.

After the possible next stable markings and probabilities of these markings have been computed for each subnetwork, they are combined to construct next stable markings and probabilities for the entire network. Possible next stable markings for the entire network are constructed by forming all possible combinations of μ'_s's and subnetwork next stable markings. Each marking constructed that way is denoted by the concatenation of its constituent subnetwork markings together with μ'_s. The probability of each of these global markings is then computed as the product of the probabilities of each of the constituent markings. Since each global marking could also be reached in other ways (i.e., from another $\mu' \in NP(\mu', a)$), the computed probability obtained in each way is summed to obtain the total probability for this next stable marking.

A more precise description of the algorithm is the following.

Algorithm 3 *(Uses concept of instantaneous subnetworks to determine, given a stable marking μ and activity a that can complete in μ, whether the next stable marking probability distribution is invariant over possible sets of activity choices, and if it is, computes this distribution.)*

Let $NS(\mu, a)$ *equal the null set.*
Let $h_{\mu,a} = 0$.
Compute $NP(\mu, a)$.
For each $\mu' \in NP(\mu, a)$*:*
    Let $\bar{h}_{\mu,a}(\mu') = \sum_{c \text{ such that } \mu \xrightarrow{a,c} \mu'} C_a(\mu_{IP(a) \cup OP(a)}, c)$.
    If μ' *is stable then*
        Add μ' *to* $NS(\mu, a)$.
        Let $h_{\mu,a}(\mu') = \bar{h}_{\mu,a}(\mu')$.
    else
        For each instantaneous subnetwork i, $i = 1$ *to* n*:*
            Restrict μ' *to places of the subnetwork (this is denoted by* μ'_i*).*
            Compute the set of all partial stable steps of the subnetwork with initial marking μ'_i.
            Compute the set of resulting stable markings from the set of partial stable steps. Label these k_i *markings* $\mu''_{i,1}, \mu''_{i,2}, \ldots, \mu''_{i,k_i}$.
            Group the set of partial stable steps computed above according to their resulting markings. $P_{\mu'_i,\mu''_{i,j}}$ *denotes the subset containing partial stable steps from* μ'_i *to* $\mu''_{i,j}$.
            For each $P_{\mu'_i,\mu''_{i,j}}$, $j = 1$ *to* k_i*:*
                Construct the set of maximal compatibles of R *on* $P_{\mu'_i,\mu''_{i,j}}$.
                Compute $P(C)$ *for each maximal compatible* C.
                If $P(C)$ *is not identical for all compatibles* C *then*
                    Signal SAN is not well-specified and abort algorithm.
                else
                    Add $\mu''_{i,j}$ *to* $NS_i(\mu'_i)$.
                    Let $\hat{h}_{\mu'_i}(\mu''_{i,j}) = P(C)$.
            Next j.
        Next i.
    {* *Now form global next stable marking from subnetwork results* *}
    For $j_1 = 1$ *to* k_1
        For $j_2 = j_1$ *to* k_2
        $\cdots$
            For $j_n = j_{n-1}$ *to* k_n

                Add $\mu'_{1,j_1}\mu'_{2,j_2}\cdots\mu'_{n,j_n}\mu'_s$ *to* $NS(\mu, a)$.
                Let $h_{\mu,a}(\mu'_{1,j_1}\mu'_{2,j_2}\cdots\mu'_{n,j_n}\mu'_s) =$
                $h_{\mu,a}(\mu'_{1,j_1}\mu'_{2,j_2}\cdots\mu'_{n,j_n}\mu'_s) + \bar{h}_{\mu,a}(\mu') \prod_{i=1 \text{ to } n} \hat{h}_{\mu'_i}(\mu'_{i,j_i})$.
            Next j_n.

Next j_2.

Next j_1.
Next $\mu' \in NP(\mu, a)$.
Return next stable marking probability distribution for marking and activity.

This algorithm has significant advantages over Algorithm 1 for systems that have several instantaneous subnetworks, and reduces to Algorithm 1 when there is a single subnetwork. To illustrate this, consider again the stochastic activity network of Figure 7, which was shown using Algorithm 1 to be well-specified. Figure 2.9 depicts this activity network with the instantaneous subnetworks outlined. As shown by Algorithm 3, each of these subnetworks can be analyzed

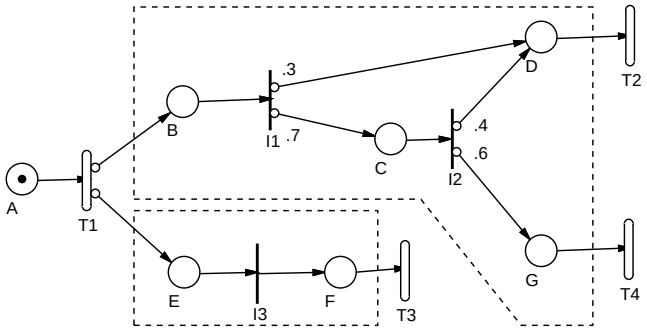


Fig. 9. A Well-Specified Stochastic Activity Network with Instantaneous Subnetworks Noted

separately to check that the network is well-specified and to compute the next stable state probability distribution. In order to do this, we must first compute the set of next possible markings, which in this case is $\{0100100\}$. The probability of this marking, denoted $\bar{h}_{1000000,T1}(0100100)$, is 1. Now, since the marking is not stable, the set of next stable states and their probabilities must be computed for each instantaneous subnetwork. For the first subnetwork, which contains $I1$ and $I2$, the set of partial stable steps is

$$\left\{ \begin{array}{l} \langle 1000, I1, 1\rangle, \\ \langle 1000, I1, 2\rangle\langle 0100, I2, 1\rangle, \\ \langle 1000, I1, 2\rangle\langle 0100, I2, 2\rangle \end{array} \right\}$$

and the set of possible next stable markings, NS_1, is $\{0010, 0001\}$. There is one maximal compatible corresponding to each possible next stable marking. The first,

$$C = \{\langle 1000, I1, 1\rangle, \langle 1000, I1, 2\rangle, \langle 0100, I2, 1\rangle\},$$

has probability $P(C) = .58$, and hence $\hat{h}_{0010} = .58$. The second, corresponding to the stable marking 0001, is

$$C = \{\langle 1000, I1, 2\rangle\langle 0100, I2, 2\rangle\}$$

and has probability $\hat{h}_{1000}(0001) = .42$. The computations for the second subnetwork are even simpler. For it, the set of partial stable steps is the singleton set

$$\{\langle 10, I3, 1\rangle\}$$

where the set of possible stable markings, NS_2, is $\{01\}$ and $\hat{h}_{10}(01) = 1$.

Since the next stable state probabilities are invariant for each subnetwork, they are invariant for the entire network. The global next stable marking probabilities are computed by forming possible combinations of the local next stable markings. When this is done, the set of next stable markings is found to be $NS(1000000) = \{0001010, 0000011\}$ with probabilities $h_{1000000,T1}(0001010) = .58$ and $h_{1000000,T1}(0000011) = .42$. This result matches that computed previously using Algorithm 1.

4 Conclusion

This chapter has presented a formal definition of activity networks and stochastic activity networks, formally described their behavior, and specified conditions which describe when their behavior is completely probabilistically defined. By providing formal definitions of these nets, we were able to precisely define when they can be used for evaluation. Other publications have defined the framework for defining reward variables on SANs [21] and the stochastic process representations of the behavior of SANs [22], when these conditions are met.

References

1. J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Transactions on Computers*, vol. C-22, no. 10, pp. 720–731, Aug. 1980.
2. M. K. Molloy, "Performance analysis using stochastic Petri nets," *IEEE Transactions on Computers*, vol. C-31, pp. 913–917, 1982.
3. G. Florin and S. Natkin, "Les Reseaux de Petri Stochastiques," *Technique et Science Informatiques*, vol. 4, no. 1, pp. 143–160, 1985.
4. B. Beyaert, G. Florin, P. Lonc, and S. Natkin, "Evaluation of computer systems dependability using stochastic Petri nets," in *Proc. 11th Int. Symp. Fault-Tolerant Computing (FTCS-11)*, Portland, Maine, USA, 1981, pp. 79–81, IEEE Computer Society Press.
5. W. H. Sanders and J. F. Meyer, "METASAN: A Performability Evaluation Tool Based on Stochastic Activity Networks," in *Proceedings of the IEEE-ACM Fall Joint Computer Conference*, Dallas, TX, November 1986, pp. 807–816.

6. W. H. Sanders, W. D. Obal II, M. A. Qureshi, and F. K. Widjanarko, "The *UltraSAN* Modeling Environment," *Performance Evaluation*, vol. 24, no. 1, pp. 89–115, October-November 1995.
7. D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders, "Möbius: An extensible tool for performance and dependability modeling," in *Computer Performance Evaluation: Modelling Techniques and Tools: Proceedings of the 11th International Conference, TOOLS 2000*, B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith, Eds. vol. 1786 of *Lecture Notes in Computer Science*, pp. 332–336, Berlin, Springer-Verlag.
8. W. H. Sanders, "Integrated frameworks for multi-level and multi-formalism modeling," in *Proceedings of PNPM'99: 8th International Workshop on Petri Nets and Performance Models*, Zaragoza, Spain, September 1999, pp. 2–9.
9. J. F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: structure, behavior and application," *Proc. International Workshop on Timed Petri Nets*, pp. 106–115, 1985.
10. A. Movaghar and J. F. Meyer, "Performability modeling with stochastic activity networks," in *Proc. 1984 Real-Time Systems Symposium*, Austin, TX, December 1984.
11. P. S. Thiagarajan, "Elementary net systems," in *Petri nets: central models and their properties*, W. Brauer, Ed. 1986, vol. 254 of *Lecture Notes in Computer Science*, pp. 26–59, Berlin, Springer-Verlag.
12. A. Movaghar, *Performability modeling with stochastic activity networks*, Ph.D. thesis, University of Michigan, 1985.
13. J. L. Peterson, *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.
14. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, N. Reading, MA, 1980.
15. W. H. Sanders, *Construction and solution of performability models based on stochastic activity networks*, Ph.D. thesis, University of Michigan, 1988.
16. M. A. Qureshi, W. H. Sanders, A. P. A. van Moorsel, and R. German, "Algorithms for the generation of state-level representations of stochastic activity networks with general reward structures," *IEEE Transactions on Software Engineering*, vol. 22, no. 9, pp. 603–614, Sept. 1996.
17. D. D. Deavours and W. H. Sanders, "An efficient well-specified check," in *Proceedings of PNPM'99: 8th International Workshop on Petri Nets and Performance Models*, Zaragoza, Spain, September 1999, pp. 124–133.
18. G. Ciardo and R. Zijal, "Well-defined stochastic Petri nets," in *Proceedings of the Fourth International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'96)*, San Jose, California, Feb. 1996, pp. 278–284.
19. J. P. Tremblay and R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw-Hill, New York, 1975.
20. Z. Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1978.
21. W. H. Sanders and J. F. Meyer, "A unified approach for specifying measures of performance, dependability, and performability," *Dependable Computing and Fault Tolerant Systems*, vol. 4, pp. 215–237, 1991.
22. W. H. Sanders and J. F. Meyer, "Reduced base model construction methods for stochastic activity networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 1, pp. 25–36, Jan. 1991.