Performance Modelling — Lecture 9
Using a GSPN for Performance Evaluation

Jane Hillston
School of Informatics
The University of Edinburgh
Scotland

16th February 2017
From SPN to Markov Process

Generating the Markov process underlying an SPN model is very straightforward.
From SPN to Markov Process

Generating the Markov process underlying an SPN model is very straightforward.

There is a one-to-one correspondence between the nodes of reachability graph of the SPN and the states in the state transition diagram of the Markov process.
Generating the Markov process underlying an SPN model is very straightforward.

There is a one-to-one correspondence between the nodes of reachability graph of the SPN and the states in the state transition diagram of the Markov process.

- We associate a state, $x_i$, in the Markov process with every marking, $M_i$, in the reachability graph of the SPN;
From SPN to Markov Process

Generating the Markov process underlying an SPN model is very straightforward.

There is a one-to-one correspondence between the nodes of reachability graph of the SPN and the states in the state transition diagram of the Markov process.

- We associate a state, $x_i$, in the Markov process with every marking, $M_i$, in the reachability graph of the SPN;
- The transition rate from state $x_i$ (corresponding to marking $M_i$) to state $x_j$ ($M_j$), is obtained as the sum of the firing rates of the transitions that are enabled in $M_i$ and whose firings generate marking $M_j$. 
Simple Example

\[
Q = \begin{pmatrix}
-\lambda_2 & \lambda_2 \\
\lambda_1 + \lambda_3 & -(\lambda_1 + \lambda_3)
\end{pmatrix}
\]
From GSPN to Markov Process

GSPN had two additional features compared to SPN: inhibitor arcs and immediate transitions.
From GSPN to Markov Process

GSPN had two additional features compared to SPN: inhibitor arcs and immediate transitions.

The effect of the inhibitor arcs is only to eliminate some potential markings and transitions from the reachability graph. Once such a graph is constructed it can be mapped to a Markov process just as in the case for SPN.
From GSPN to Markov Process

GSPN had two additional features compared to SPN: inhibitor arcs and immediate transitions.

The effect of the inhibitor arcs is only to eliminate some potential markings and transitions from the reachability graph. Once such a graph is constructed it can be mapped to a Markov process just as in the case for SPN.

The effect of the immediate transitions is to create some markings which do not correspond to states in a Markov process, so-called vanishing markings.
Vanishing and Tangible Markings

If a marking in a GSPN enables an immediate transition, by the firing rules, the immediate transition must fire instantaneously, and so the marking will be changed again without any time elapsing.
Vanishing and Tangible Markings

If a marking in a GSPN enables an immediate transition, by the firing rules, the immediate transition must fire instantaneously, and so the marking will be changed again without any time elapsing.

This is in contrast with a Markov process, where the sojourn time in each state is exponentially distributed by definition.
Vanishing and Tangible Markings

If a marking in a GSPN enables an immediate transition, by the firing rules, the immediate transition must fire instantaneously, and so the marking will be changed again without any time elapsing.

This is in contrast with a Markov process, where the sojourn time in each state is exponentially distributed by definition.

In contrast if a marking enables only timed transitions there will be an exponentially distributed delay before a transition fires, which corresponds to an exponentially distributed sojourn time in the corresponding state in the Markov process.
Vanishing and Tangible Markings

If a marking in a GSPN enables an immediate transition, by the firing rules, the immediate transition must fire instantaneously, and so the marking will be changed again without any time elapsing.

This is in contrast with a Markov process, where the sojourn time in each state is exponentially distributed by definition.

In contrast if a marking enables only timed transitions there will be an exponentially distributed delay before a transition fires, which corresponds to an exponentially distributed sojourn time in the corresponding state in the Markov process.

Such markings are called tangible as opposed to vanishing.
Generating a Markov Process from a GSPN Reader-Writer Example Generating Performance Measures Assumptions

Eliminating Vanishing Markings

Vanishing markings must be *eliminated* from the reachability graph before the state space of the Markov process can be generated.
Eliminating Vanishing Markings

Vanishing markings must be eliminated from the reachability graph before the state space of the Markov process can be generated.

If immediate transitions from a marking can lead to two or more different markings, the transition rates to these markings need to be adjusted (according to the decomposition principle).
Eliminating Vanishing Markings: a single immediate transition

- Suppose a vanishing marking $M_v$, enables a single immediate transition, $T_j$;

```
Mp
  "Ti"  "ri"

Mv
  "Tj"  "∞"

Ms
```
Eliminating Vanishing Markings: a single immediate transition

- Suppose a vanishing marking $M_v$, enables a single immediate transition, $T_j$;
- There will be a single successor marking, $M_s$, which is the result of firing $T_j$. 

\[ \text{Eliminating Vanishing Markings: a single immediate transition} \]
Eliminating Vanishing Markings: a single immediate transition

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$ with rate $r_i$

- delete $M_v$;
Eliminating Vanishing Markings: a single immediate transition

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$ with rate $r_i$:

- delete $M_v$;
- we draw an arc from $M_p$ to $M_s$ labelled $T_i + T_j$, with the rate the same as the rate of $T_i$;
Eliminating Vanishing Markings: a single immediate transition

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$ with rate $r_i$:

- delete $M_v$;
- we draw an arc from $M_p$ to $M_s$ labelled $T_i + T_j$, with the rate the same as the rate of $T_i$;
- delete the arcs corresponding to $T_i$ and $T_j$. 
Eliminating Vanishing Markings: a single immediate transition

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$ with rate $r_i$:

- delete $M_v$;
- we draw an arc from $M_p$ to $M_s$ labelled $T_i + T_j$, with the rate the same as the rate of $T_i$;
- delete the arcs corresponding to $T_i$ and $T_j$. 

\[ M_p \overset{T_i+T_j}{\rightarrow} M_s \]
Suppose a vanishing marking $M_v$, enables $n$ immediate transitions, $T_{j_1}, \ldots, T_{j_n}$;
Eliminating Vanishing Markings: multiple immediate transitions

- Suppose a vanishing marking $M_v$, enables $n$ immediate transitions, $T_{j_1}, \ldots, T_{j_n}$;
- There will be a successor marking, $M_{s_n}$, for the firing of each $T_{j_n}$.

```
\begin{tikzpicture}
  \node (mp) at (0,0) {$M_p$};
  \node (mv) at (0,-2) {$M_v$};
  \node (ms1) at (-2,-4) {$M_{s_1}$};
  \node (msn) at (2,-4) {$M_{s_n}$};
  \node (ti) at (0,-3.5) {$T_i$};
  \node (ri) at (0,-3.5) {$r_i$};
  \draw[->] (mp) -- (mv) node[midway, above] {$T_i$};
  \draw[->] (mv) -- (ms1) node[midway, above] {$T_{j_1}$};
  \draw[->] (mv) -- (msn) node[midway, above] {$T_{j_n}$};
  \node at (0,-4.5) {$p_1$};
  \node at (0,-4.5) {$p_n$};
  \node at (-1,-4) {$\infty$};
  \node at (1,-4) {$\infty$};
\end{tikzpicture}
```
Eliminating Vanishing Markings: multiple immediate transitions

- Suppose a vanishing marking $M_v$, enables $n$ immediate transitions, $T_{j_1}, \ldots, T_{j_n}$;
- There will be a successor marking, $M_{s_n}$, for the firing of each $T_{j_n}$.
- These transitions will be in conflict and so each must have a probability $p_k$ such that $\sum_{k=1}^n p_n = 1$
Eliminating Vanishing Markings: multiple immediate transitions

To eliminate \( T_j \) for each predecessor marking \( M_p \), linked to \( M_v \) by an arc labelled \( T_i \)

- delete \( M_v \);
Eliminating Vanishing Markings: multiple immediate transitions

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$

- delete $M_v$;
- for each successor marking, we draw an arc from $M_p$ to $M_{s_k}$ labelled $T_i + T_{jk}$, with the rate the same as the rate of $T_i$ multiplied by the probability of taking arc $T_{jk}$, i.e. $p_k$;
Eliminating Vanishing Markings: multiple immediate transitions

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$

- delete $M_v$;
- for each successor marking, we draw an arc from $M_p$ to $M_{s_k}$ labelled $T_i + T_{jk}$, with the rate the same as the rate of $T_i$ multiplied by the probability of taking arc $T_{jk}$, i.e. $p_k$;
- delete the arcs corresponding to $T_i$ and $T_j$. 
Generating a Markov Process from a GSPN Reader-Writer Example Generating Performance Measures Assumptions

Eliminating Vanishing Markings: multiple immediate transitions

To eliminate $T_j$ for each predecessor marking $M_p$, linked to $M_v$ by an arc labelled $T_i$

- delete $M_v$;
- for each successor marking, we draw an arc from $M_p$ to $M_{s_k}$ labelled $T_i + T_{j_k}$, with the rate the same as the rate of $T_i$ multiplied by the probability of taking arc $T_{j_k}$, i.e. $p_k$;
- delete the arcs corresponding to $T_i$ and $T_j$. 

Jane Hillston
School of Informatics
The University of Edinburgh
Scotland

Performance Modelling — Lecture 9 Using a GSPN for Performance Evaluation
Eliminating Vanishing Markings

This procedure is systematically applied to all the vanishing markings in the reachability graph.
Eliminating Vanishing Markings

- This procedure is systematically applied to all the vanishing markings in the reachability graph.
- In the end, all arcs in the modified reachability graph will have a rate originating from a single timed transition associated with the arc.
Eliminating Vanishing Markings

- This procedure is systematically applied to all the vanishing markings in the reachability graph.
- In the end, all arcs in the modified reachability graph will have a rate originating from a single timed transition associated with the arc.
- This rate may have been adjusted during the elimination of vanishing markings to reflect the relative probability of immediate transitions enabled after the timed transition.
Eliminating Vanishing Markings

- This procedure is systematically applied to all the vanishing markings in the reachability graph.
- In the end, all arcs in the modified reachability graph will have a rate originating from a single timed transition associated with the arc.
- This rate may have been adjusted during the elimination of vanishing markings to reflect the relative probability of immediate transitions enabled after the timed transition.
- It is this modified reachability graph that is used to generate the Markov process underlying a GSPN model.
Reader-Writer Example

Consider again the system in which there is a set of processes who share access to a common database.
Reader-Writer Example

- Consider again the system in which there is a set of processes who share access to a common database.
- On any particular access a process may wish to perform either a read or a write.
Reader-Writer Example

- Consider again the system in which there is a set of processes who share access to a common database.
- On any particular access a process may wish to perform either a read or a write.
- Any number of readers may access the database concurrently;
Reader-Writer Example

- Consider again the system in which there is a set of processes who share access to a common database.
- On any particular access a process may wish to perform either a read or a write.
- Any number of readers may access the database concurrently;
- a writer requires exclusive access.
Reader-Writer Example

- Consider again the system in which there is a set of processes who share access to a common database.
- On any particular access a process may wish to perform either a read or a write.
- Any number of readers may access the database concurrently;
- a writer requires exclusive access.
- Between accesses processes undertake independent work (concurrently).
GSPN model of the reader-writer system

Jane Hillston School of Informatics The University of Edinburgh Scotland
Performance Modelling — Lecture 9 Using a GSPN for Performance Evaluation
Reachability graph

Jane Hillston School of Informatics The University of Edinburgh Scotland
Performance Modelling — Lecture 9 Using a GSPN for Performance Evaluation
Reachability set of the reader-writer model

<table>
<thead>
<tr>
<th>( M )</th>
<th>State</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_0 )</td>
<td>((2, 0, 0, 0, 1, 0, 0))</td>
<td>tangible</td>
</tr>
<tr>
<td>( M_1 )</td>
<td>((1, 1, 0, 0, 1, 0, 0))</td>
<td>vanishing</td>
</tr>
<tr>
<td>( M_2 )</td>
<td>((1, 0, 1, 0, 1, 0, 0))</td>
<td>vanishing</td>
</tr>
<tr>
<td>( M_3 )</td>
<td>((1, 0, 0, 1, 1, 0, 0))</td>
<td>vanishing</td>
</tr>
<tr>
<td>( M_4 )</td>
<td>((1, 0, 0, 0, 1, 1, 0))</td>
<td>tangible</td>
</tr>
<tr>
<td>( M_5 )</td>
<td>((1, 0, 0, 0, 0, 0, 1))</td>
<td>tangible</td>
</tr>
<tr>
<td>( M_6 )</td>
<td>((0, 1, 0, 0, 1, 1, 0))</td>
<td>vanishing</td>
</tr>
<tr>
<td>( M_7 )</td>
<td>((0, 0, 1, 0, 1, 1, 0))</td>
<td>vanishing</td>
</tr>
<tr>
<td>( M_8 )</td>
<td>((0, 0, 0, 1, 1, 1, 0))</td>
<td>tangible</td>
</tr>
<tr>
<td>( M_9 )</td>
<td>((0, 0, 0, 0, 1, 2, 0))</td>
<td>tangible</td>
</tr>
<tr>
<td>( M_{10} )</td>
<td>((0, 1, 0, 0, 0, 0, 1))</td>
<td>vanishing</td>
</tr>
<tr>
<td>( M_{11} )</td>
<td>((0, 0, 1, 0, 0, 0, 1))</td>
<td>tangible</td>
</tr>
<tr>
<td>( M_{12} )</td>
<td>((0, 0, 0, 1, 0, 0, 1))</td>
<td>tangible</td>
</tr>
</tbody>
</table>
Reachability graph
Reachability graph
Reachability graph
Reachability graph
Reachability graph
Reachability graph
Reduced reachability graph
Performance Measures from GSPN

The steady state probability distribution, $\pi$, is still the basis of performance evaluation.
Performance Measures from GSPN

The steady state probability distribution, $\pi$, is still the basis of performance evaluation.

In other words the aim is to derive performance characteristics of the system based on the steady state probability of being in certain states, or markings.
Performance Measures from GSPN

The steady state probability distribution, $\pi$, is still the basis of performance evaluation.

In other words the aim is to derive performance characteristics of the system based on the steady state probability of being in certain states, or markings.

In GSPN we can identify the states we are interested in by their characteristics at the GSPN level.
Throughput and Average Marking

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:
Throughput and Average Marking

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:

- For transitions this will be the **throughput**: the number of times the transition fires per unit time.
Throughput and Average Marking

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:

- For transitions this will be the **throughput**: the number of times the transition fires per unit time.
- For places this will be the **average marking**: the expected number of tokens in the place at steady state.
Throughput and Average Marking

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:

- For transitions this will be the **throughput**: the number of times the transition fires per unit time.
- For places this will be the **average marking**: the expected number of tokens in the place at steady state.
- The **probability enabled** may also be recorded for each transition.
Throughput and Average Marking

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:

- For transitions this will be the **throughput**: the number of times the transition fires per unit time.
- For places this will be the **average marking**: the expected number of tokens in the place at steady state.
- The **probability enabled** may also be recorded for each transition.
- Sometimes the **steady state distribution** of tokens in a place will be given.
**Throughput and Average Marking**

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:

- For transitions this will be the **throughput**: the number of times the transition fires per unit time.
- For places this will be the **average marking**: the expected number of tokens in the place at steady state.
- The **probability enabled** may also be recorded for each transition.
- Sometimes the **steady state distribution** of tokens in a place will be given.
Throughput and Average Marking

Most SPN/GSPN tools will automatically report the performance measures associated with each element of the Petri net:

- For transitions this will be the **throughput**: the number of times the transition fires per unit time.
- For places this will be the **average marking**: the expected number of tokens in the place at steady state.
- The **probability enabled** may also be recorded for each transition.
- Sometimes the **steady state distribution** of tokens in a place will be given.

We can often derive the measures we are interested in directly from these default measures.
Throughput and Average Marking: example

In the reader-writer model the throughput of transition $T_6$ plus the throughput of transition $T_7$ will give us the throughput of the database in terms of accesses/unit time.
In the reader-writer model the throughput of transition $T_6$ plus the throughput of transition $T_7$ will give us the throughput of the database in terms of accesses/unit time.

Similarly the average number of readers in the system at steady state will be exactly the average marking of place $P_6$. 
Other measures

At the GSPN level the states we are interested in can be identified either by whether a particular transition is enabled, or by whether a particular place is marked.
Other measures

At the GSPN level the states we are interested in can be identified either by whether a particular transition is enabled, or by whether a particular place is marked.

To derive other performance measures we associate a value or reward with each of the markings we are interested in, just as we did when working directly at the Markov process level.
Other measures

At the GSPN level the states we are interested in can be identified either by whether a particular transition is enabled, or by whether a particular place is marked.

To derive other performance measures we associate a value or reward with each of the markings we are interested in, just as we did when working directly at the Markov process level.

For example, to derive the utilisation of the database in the reader-writer system, we associate a reward 1 with any marking in which transitions $T_6$ or $T_7$ are enabled, and a reward 0 with all other markings.
PIPE output for Reader-Writer example

![CSPN Analysis](image)

### GSPN Steady State Analysis Results

#### Set of Tangible States

<table>
<thead>
<tr>
<th>M0</th>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Steady State Distribution of Tangible States

<table>
<thead>
<tr>
<th>Marking</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>0.19745</td>
</tr>
<tr>
<td>M1</td>
<td>0.15287</td>
</tr>
<tr>
<td>M2</td>
<td>0.03185</td>
</tr>
<tr>
<td>M3</td>
<td>0.30573</td>
</tr>
<tr>
<td>M4</td>
<td>0.30573</td>
</tr>
<tr>
<td>M5</td>
<td>0.00318</td>
</tr>
<tr>
<td>M6</td>
<td>0.00318</td>
</tr>
</tbody>
</table>
### PIPE output for Reader-Writer example

#### Average Number of Tokens on a Place

<table>
<thead>
<tr>
<th>Place</th>
<th>Number of Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>0.57962</td>
</tr>
<tr>
<td>P1</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>0.30573</td>
</tr>
<tr>
<td>P3</td>
<td>0.30892</td>
</tr>
<tr>
<td>P4</td>
<td>0.0414</td>
</tr>
<tr>
<td>P5</td>
<td>0.76433</td>
</tr>
<tr>
<td>P6</td>
<td>0.23567</td>
</tr>
</tbody>
</table>

#### Token Probability Density

<table>
<thead>
<tr>
<th></th>
<th>$\mu=0$</th>
<th>$\mu=1$</th>
<th>$\mu=2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>0.61783</td>
<td>0.18471</td>
<td>0.19745</td>
</tr>
<tr>
<td>P1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>0.69427</td>
<td>0.30573</td>
<td>0</td>
</tr>
<tr>
<td>P3</td>
<td>0.69108</td>
<td>0.30892</td>
<td>0</td>
</tr>
<tr>
<td>P4</td>
<td>0.96178</td>
<td>0.03503</td>
<td>0.00318</td>
</tr>
<tr>
<td>P5</td>
<td>0.23567</td>
<td>0.76433</td>
<td>0</td>
</tr>
<tr>
<td>P6</td>
<td>0.76433</td>
<td>0.23567</td>
<td>0</td>
</tr>
</tbody>
</table>
PIPE output for Reader-Writer example

Throughput of Timed Transitions

<table>
<thead>
<tr>
<th>Transition</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0</td>
<td>7.64331</td>
</tr>
<tr>
<td>T5</td>
<td>3.82166</td>
</tr>
<tr>
<td>T6</td>
<td>3.82166</td>
</tr>
</tbody>
</table>

Sojourn times for tangible states

<table>
<thead>
<tr>
<th>Marking</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>0.05</td>
</tr>
<tr>
<td>M1</td>
<td>0.04</td>
</tr>
<tr>
<td>M2</td>
<td>0.00833</td>
</tr>
<tr>
<td>M3</td>
<td>0.2</td>
</tr>
<tr>
<td>M4</td>
<td>0.2</td>
</tr>
<tr>
<td>M5</td>
<td>0.01</td>
</tr>
<tr>
<td>M6</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Assumptions

Previous assumptions about Markov processes are still required but can now be interpreted at the GSPN level.
Assumptions

Previous assumptions about Markov processes are still required but can now be interpreted at the GSPN level.

Finite implies that the number of markings in the reachability set of a model (both tangible and vanishing markings) is finite. It can be shown that a GSPN is finite if it is bounded.
Assumptions

Previous assumptions about Markov processes are still required but can now be interpreted at the GSPN level.

**Finite** implies that the number of markings in the reachability set of a model (both tangible and vanishing markings) is *finite*. It can be shown that a GSPN is finite if it is *bounded*.

**Time homogeneity** implies that the firing characteristics/system dynamics do not change over time. These characteristics are not necessarily static—marking dependent rates do vary—but the firing rate can depend only on the state of the model, not on how long it has been running.
Assumptions

Previous assumptions about Markov processes are still required but can now be interpreted at the GSPN level.

**Finite** implies that the number of markings in the reachability set of a model (both tangible and vanishing markings) is finite. It can be shown that a GSPN is finite if it is bounded.

**Time homogeneity** implies that the firing characteristics/system dynamics do not change over time. These characteristics are not necessarily static—marking dependent rates do vary—but the firing rate can depend only on the state of the model, not on how long it has been running.

**Irreducibility** implies that it is possible to reach an arbitrary state from every other state.