Analysis of Process Models: Introduction, state space analysis and simulation in CPN Tools

prof.dr.ir. Wil van der Aalst
What is a Petri net?

- A graphical notion (model = picture?)
- A mathematical notion (model = graph?)
- A programming notion (model = program?)

- A solver independent medium
- Starting point for a variety of analysis approaches
Analysis

- Analysis is typically model-driven to allow e.g. what-if questions.
- Models of both operational processes and/or the information systems can be analyzed.
- Types of analysis:
  - validation
  - verification
  - performance analysis
Three types of analysis techniques

1. Reachability/coverability graph
2. Structural techniques
   • Place and transition invariants
   • Marking equation
   • Traps, siphons, etc.
3. Simulation
   • Each can be applied to both classical and high-level Petri nets.
   • Nevertheless, for the second we restrict ourselves to classical Petri nets.

Mapping technique/use:
   • reachability graph (validation, verification)
   • invariants (validation, verification)
   • simulation (validation, performance analysis)
Informal introduction ...
Examples of generic questions given a marked Petri net

- **terminating**
  it has only finite occurrence sequences
- **deadlock-free**
  each reachable marking enables a transition
- **live**
  each reachable marking enables an occurrence sequence containing all transitions
- **bounded**
  each place has an upper bound that holds for all reachable markings
- **1-safe**
  1 is a bound for each place $p$
- **reversible**
  $m_0$ is reachable from each reachable marking, i.e., the initial marking is a so-called home marking.
Reachability graph

Five reachable states. Traffic lights are safe!
**Alternative notation**

- **terminating**
  it has only finite occurrence sequences

- **deadlock-free**
  each reachable marking enables a transition

- **live**
  each reachable marking enables an occurrence sequence containing all transitions

- **bounded**
  each place has an upper bound that holds for all reachable markings

- **1-safe**
  1 is a bound for each place

- **reversible**
  $m_0$ is reachable from each reachable marking, i.e., the initial marking is a so-called home marking.
Reachability graph (2)

- Graph containing a node for each reachable state.
- Constructed by starting in the initial state, calculate all directly reachable states, etc.
- Expensive technique.
- Only feasible if finitely many states (otherwise use coverability graph).
- Difficult to generate diagnostic information.
Infinite reachability graph

Therefore tools use a coverability graph which is always finite!
Exercise: Construct reachability graph
Exercise: Dining philosophers

- 4 philosophers sharing 4 chopsticks: A philosopher is either in state eating or thinking and needs two chopsticks to eat.
- Model as a Petri net and construct reachability graph.
See also: www.workflowcourse.com

Reachability graph: The philosophers

Vector = (Position of the tokens)

D:\www\wvdaalst\workflowcourse\examples\philosopher4_RG.swf
Analysis in CPN Tools

- Only state-space analysis, i.e., no invariants.
- Generate report in text file.
- State-space visualization from version 2.2.
- Steps:
  1. Enter the State Space Tool (to generate ML code)
  2. Calculating the state space
  3. Calculating the SCC graph (to calculate home states and fairness)
  4. Save/view state space report
Example
Create report

![Image of a diagram with labeled nodes and arcs, and a screenshot of a report generated by CPN Tools showing state space statistics and boundedness properties.]
CPN Tools state space report for: 
/cygdrive/D/courses/BIS-2011/CPN files/voting-bank-etc/bank.cpn 
Report generated: Sun Mar 27 14:01:43 2011

Statistics

State Space
  Nodes: 24
  Arcs: 44
  Secs: 0
  Status: Full

Scc Graph
  Nodes: 24
  Arcs: 44
  Secs: 0
Boundedness Properties

Best Integer Bounds

<table>
<thead>
<tr>
<th></th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>main'database 1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>main'deposit 1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>main'withdraw 1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Best Upper Multi-set Bounds

- main'database 1: `1`(1,0)++ `1`(2, (~5))++ `1`(2, (~1))++ `1`(2,0)++ `1`(2,3)++ `1`(2,4)++ `1`(2,7)++ `1`(2,8)++ `1`(2,11)++ `1`(2,12)++ `1`(2,15)++ `1`(2,16)+`1`(2,20)++ `1`(3, (~9))++ `1`(3,0)
- main'deposit 1: `5`(2,4)
- main'withdraw 1: `1`(2,5)++ `1`(3,9)

Best Lower Multi-set Bounds

- main'database 1: `1`(1,0)
- main'deposit 1: empty
- main'withdraw 1: empty
Report (3)

Home Markings
[24]

Liveness Properties

Dead Markings
[24]

Dead Transition Instances
None

Live Transition Instances
None

Fairness Properties

No infinite occurrence sequences.
State Space
- Nodes: 28
- Arcs: 42
- Secs: 0
- Status: Full

Scc Graph
- Nodes: 1
- Arcs: 0
- Secs: 0

Boundedness Properties

Best Integer Bounds

<table>
<thead>
<tr>
<th></th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>New_Page'track 1</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>New_Page'train 1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Home Properties

Home Markings
All

Liveness Properties

Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All

Fairness Properties
New_Page'move 1 Impartial
Another example
CPN Tools state space report for:
C:\Program Files\CPN Tools\Samples\DiningPhilosophers\DiningPhilosophers.cpn

Statistics

State Space
- Nodes: 11
- Arcs: 30
- Secs: 0
- Status: Full

Scc Graph
- Nodes: 1
- Arcs: 0
- Secs: 0
## Boundedness Properties

### Best Integer Bounds

<table>
<thead>
<tr>
<th>Page</th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Eat 1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>'Think 1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>'Unused_Chopsticks 1</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

### Best Upper Multi-set Bounds

<table>
<thead>
<tr>
<th>Page</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Eat 1</td>
<td>1<code>ph(1)++ 1</code>ph(2)++ 1<code>ph(3)++ 1</code>ph(4)++</td>
</tr>
<tr>
<td>'Think 1</td>
<td>1<code>ph(1)++ 1</code>ph(2)++ 1<code>ph(3)++ 1</code>ph(4)++</td>
</tr>
<tr>
<td>'Unused_Chopsticks 1</td>
<td>1<code>cs(1)++ 1</code>cs(2)++ 1<code>cs(3)++ 1</code>cs(4)++ 1`cs(5)</td>
</tr>
</tbody>
</table>

### Best Lower Multi-set Bounds

<table>
<thead>
<tr>
<th>Page</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Eat 1</td>
<td>empty</td>
</tr>
<tr>
<td>'Think 1</td>
<td>empty</td>
</tr>
<tr>
<td>'Unused_Chopsticks 1</td>
<td>empty</td>
</tr>
</tbody>
</table>
Report (3)

Home Properties

Home Markings
All

Liveness Properties

Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All

Fairness Properties

Page'Put_Down_Chopsticks 1 Impartial
Page'Take_Chopsticks 1 Impartial

strongest fairness property, i.e., there are infinite firing sequences and in each infinite firing sequence t occurs infinitely often
Fairness properties

- Are only relevant if there are Infinite Firing Sequences (IFS), otherwise CPN Tools reports: "no infinite occurrence sequences".
- Given a transition $t$ it is often desirable that $t$ appears infinitely often in an IFS.
- Properties reported by CPN Tools
  - $t$ is **impartial**: $t$ occurs infinitely often in every IFS.
  - $t$ is **fair**: $t$ occurs infinitely often in every IFS where $t$ is enabled infinitely often.
  - $t$ is **just**: $t$ occurs infinitely often in every IFS where $t$ is continuously enabled from some point onward
  - **No fairness**: not just, i.e., there is an IFS where $t$ is continuously enabled from some point onward and does not fire anymore
Example

Fairness Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>main1'a 1</td>
<td>Just</td>
</tr>
<tr>
<td>main1'b 1</td>
<td>Just</td>
</tr>
<tr>
<td>main1'c 1</td>
<td>Impartial</td>
</tr>
</tbody>
</table>
Example

Fairness Properties

<table>
<thead>
<tr>
<th>main2'x 1</th>
<th>No Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>main2'y 1</td>
<td>No Fairness</td>
</tr>
</tbody>
</table>
Example

Fairness Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>main3't1 1</td>
<td>Fair</td>
</tr>
<tr>
<td>main3't2 1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>main3't3 1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>main3't4 1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>main3't5 1</td>
<td>No Fairness</td>
</tr>
<tr>
<td>main3't6 1</td>
<td>Just</td>
</tr>
</tbody>
</table>
Indicate for each transition whether it is impartial, fair, or just (or satisfies no fairness property)
• t1, t2, and t3 are all impartial because it is not possible to construct an infinite firing sequence where not all of these transitions appear infinitely often. If one stops executing one of these transitions, the system will block after a while.

• t4 has no fairness as it is possible to construct an infinite firing sequence where t4 remains enabled but never fires.
Simulation

• Most widely used analysis technique.
• From a technical point of view just a "walk" in the reachability graph.
• By making many "walks" (in case of transient behavior) or a very "long walk" (in case of steady-state) behavior, it is possible to make reliable statements about properties/ performance indicators.
• Used for validation and performance analysis.
• Cannot be used to prove correctness!
**Stochastic process**

- Simulation of a deterministic system is not very interesting.
- Simulation of an untimed system is not interesting.
- In a timed and non-deterministic system, durations and probabilities are described by some probability distribution.
- In other words, we simulate a stochastic process!
- CPN allows for the use of distributions using some internal random generator.
Uniform distribution

- **Probability density function (PDF)**
- **Cumulative distribution function (CDF)**
Negative exponential distribution
Normal distribution
Distributions in CPN Tools

There is standard library with stochastic functions:

- uniform(a:real, b:real) : real
- exponential(r:real) : real
- normal(n:real, v:real) : real
- erlang (n:int, r:real) : real
- Etc.

A nice additional function is also C.ran() which returns a randomly selected element of finite color set C, e.g.,

```
color C = int with 1..5;
fun select1to5() = C.ran()
```
returns a number between 1 and 5
color BT = unit;
color Dice = int with 1..6;
var d : Dice;

or even simpler …
Example (2)

```
color INT = int;
color TINT = int timed;
color Dice = int with 1..6;
color Delay = int with 0..99;
```

```
(x+1)@+(Delay.ran())
```

```
0
```

```
(x+1)@+(Delay.ran())
```

```
TINT
```

```
trigger
```

```
throw_dice
```

```
Dice.ran()
```

```
outcome
```

```
Dice
```
colset Dice = int with 1..6;
After 2055 times throwing the dices ... five 4's

colset Dice = int with 1..6;

colset Dice = int with 1..6;
val it = 0 : int
bernoulli(0.5)
val it = 5 : int
binomial(10,0.5)
val it = 6 : int
discrete (1,10)
val it = 23.9142997372 : real
erlang(10,0.5)
val it = 0.0112478388921 : real
exponential(100.0)
val it = 8.85471291714 : real
normal(10.0,5.0)
val it = 1 : int
poisson(0.5)
val it = 16.3058542628 : real
uniform(10.0,20.0)
fun iat() = round(exponential(0.05));
fun pt() = round(normal(10.0,1.0));
alternative notation $[b] \%\{v = \text{if } b \text{ then } 1 \text{\`}v \text{ else empty} \}$
Adding hierarchy
Example revisited
Subruns and confidence intervals

• A single run does **not** provide information about reliability of results.
• Therefore, multiple runs or one run cut into parts: subruns.
• If the subruns are assumed to be **mutually independent**, one can calculate a **confidence interval**, e.g., the flow time is with 95% confidence within the interval 5.5+/-0.5 (i.e. [5,6]).
Two possible settings

Steady-state analysis (I)

Steady-state analysis (II)
More on calculating confidence intervals
is not the same as

although the average over the subrun results is the same (5.7)
"low level" measurements

aggregation per subrun
(average, min, max, variance, etc.)

subruns = 11
average = 5.7
standard deviation = 0.21

confidence = 0.9
confidence interval = [5.7-0.117, 5.7+0.117] = [5.58, 5.82]
Using monitors in CPN Tools

Note that these statistics have been calculated for data that is not necessarily independent or identically distributed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Count</th>
<th>Avg</th>
<th>Min</th>
<th>Max</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marking_size_ServerBusy_1</td>
<td>201</td>
<td>0.828407</td>
<td>0</td>
<td>1</td>
<td>11131</td>
</tr>
<tr>
<td>Queue_Length</td>
<td>209</td>
<td>1.992993</td>
<td>0</td>
<td>9</td>
<td>11131</td>
</tr>
<tr>
<td>Server_Utilization</td>
<td>118</td>
<td>0.828407</td>
<td>0</td>
<td>1</td>
<td>11131</td>
</tr>
</tbody>
</table>

Untimed statistics

<table>
<thead>
<tr>
<th>Name</th>
<th>Count</th>
<th>Sum</th>
<th>Avg</th>
<th>Std</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count_trans_occur_Arrivals_Arrive_1</td>
<td>107</td>
<td>107</td>
<td>1.000000</td>
<td>0.000000</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Processed_A_Jobs</td>
<td>46</td>
<td>46</td>
<td>1.000000</td>
<td>0.000000</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Queue_Delay</td>
<td>100</td>
<td>19933</td>
<td>199 330000</td>
<td>252.527148</td>
<td>0</td>
<td>1255</td>
</tr>
</tbody>
</table>

Simulation steps executed: 307
Model time: 11131
Generated: Wed Feb 15 15:11:03
Example of a simulation model

- Gas station with one pump and space for 4 cars (3 waiting and 1 being served).
- Service time: uniform distribution between 2 and 5 minutes.
- Poisson arrival process with mean time between arrivals of 4 minutes.
- If there are more than 3 cars waiting, the "sale" is lost.
- Questions: flow time, waiting time, utilization, lost sales, etc.
color Car = string

Top-level page: main

color Car = string

- environment
- gas_station
- arrive Car
- drive_on Car
- depart Car
color Car = string;
color Pump = unit;
color TCar = Car timed;
color Queue = list Car;
var c:Car;
var q:Queue;
fun len(q:Queue) = if q=[] then 0
            else 1+len(tl(q));
Interesting performance indicators:

- Calculation of flow time (average, variance, maximum, minimum, service level, etc.).
- Calculation of waiting times (average, variance, maximum, minimum, service level, etc.).
- Calculation of lost sales (average).
- Probability of no space left.
- Probability of no cars waiting.
Alternatives

Model the following alternatives:

- 6 waiting spaces
- 2 pumps
- 1 faster pump
Experiments
(note time dimension * 1000; not needed in CPN Tools Version 3)
CPN Tools (Version 2.2.0 - September 2006)

Tool box
Help
Options
History

fuelstation.cn

Step: 0
Time: 0
Options
History

Declarations

- Standard declarations
  - colset Stime = int;
  - colset CarId = string;
  - colset Car = product CarId * Stime;
  - colset Pump = unit timed;
  - colset Queue = list Car;
  - colset Tint = int timed;
  - colset INT=int;
  - var c:Car;
  - var cid:CarId;
  - var t:Stime;
  - var q:Queue;
  - var ij,k,l:INT;
  - fun IsIn(q:Queue) = if q=[] then 0

Monitors

- flowtime_drive_on
- flowtime_depart
- percentage_serviced
- Marking_size_gas_station'fill_up_1
- Marking_size_gas_station 'pump_f
- List_length_dc_gas_station'queue

environment

monitors
Number of cars being served
Number of pumps free
Length of queue
Flow time for cars not served
Flow time for cars that have been served
Percentage of cars served
Just one run ...

Average flow is
time 7.359
Subruns in CPN Tools

$\times 10$

CPN'Replications.nreplications 10
<table>
<thead>
<tr>
<th>Name</th>
<th>Avrg</th>
<th>90% Half Length</th>
<th>95% Half Length</th>
<th>99% Half Length</th>
<th>StD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count iid</td>
<td>19128.800000</td>
<td>25.132533</td>
<td>31.014615</td>
<td>44.561228</td>
<td>43.358454</td>
<td>19056</td>
<td>19181</td>
</tr>
<tr>
<td>Max iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Avg iid</td>
<td>0.891917</td>
<td>0.014062</td>
<td>0.017353</td>
<td>0.024932</td>
<td>0.024259</td>
<td>0.841207</td>
<td>0.924649</td>
</tr>
</tbody>
</table>

**Average queue length** [0.878, 0.906]

<table>
<thead>
<tr>
<th>Name</th>
<th>Avrg</th>
<th>90% Half Length</th>
<th>95% Half Length</th>
<th>99% Half Length</th>
<th>StD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count iid</td>
<td>18255.600000</td>
<td>50.265065</td>
<td>62.029230</td>
<td>89.122456</td>
<td>86.716909</td>
<td>18110</td>
<td>18360</td>
</tr>
<tr>
<td>Max iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Avg iid</td>
<td>0.797933</td>
<td>0.003853</td>
<td>0.004755</td>
<td>0.006832</td>
<td>0.006648</td>
<td>0.790479</td>
<td>0.810488</td>
</tr>
</tbody>
</table>

**Average # pumps busy** [0.794, 0.801]

<table>
<thead>
<tr>
<th>Name</th>
<th>Avrg</th>
<th>90% Half Length</th>
<th>95% Half Length</th>
<th>99% Half Length</th>
<th>StD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count iid</td>
<td>18255.600000</td>
<td>50.265065</td>
<td>62.029230</td>
<td>89.122456</td>
<td>86.716909</td>
<td>18110</td>
<td>18360</td>
</tr>
<tr>
<td>Max iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Avg iid</td>
<td>0.202067</td>
<td>0.003853</td>
<td>0.004755</td>
<td>0.006832</td>
<td>0.006648</td>
<td>0.189512</td>
<td>0.209521</td>
</tr>
</tbody>
</table>

**Average # pumps free** [0.198, 0.206]
### Average flow time

![Table](attachment://table.png)

**Average flow time** \[7.352, 7.460\]

### Average fraction served

![Table](attachment://table.png)

**Average fraction served** \[0.910, 0.915\]

*Generated: Sun Mar 28 19:14:39 2010*
## Results

<table>
<thead>
<tr>
<th></th>
<th>Average flow time</th>
<th>Average fraction served</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base case</strong></td>
<td>[7.352, 7.460]</td>
<td>[0.910, 0.915]</td>
</tr>
</tbody>
</table>
2 pumps
Average queue length [0.105, 0.111]

Average # pumps busy [0.867, 0.878]

Average # pumps free [1.122, 1.133]
### Average Flow Time

Average flow time: [3.916, 3.944]

### Average Fraction Served

Average fraction served: [0.996, 0.997]

---

**Table: flowtime_depart**

<table>
<thead>
<tr>
<th>avg_iid</th>
<th>1.127238</th>
<th>0.005708</th>
<th>0.007044</th>
<th>0.010121</th>
<th>0.009848</th>
<th>1.116292</th>
<th>1.146355</th>
</tr>
</thead>
<tbody>
<tr>
<td>count_iid</td>
<td>9961.900000</td>
<td>3.318017</td>
<td>4.094574</td>
<td>5.883009</td>
<td>5.724218</td>
<td>9953</td>
<td>9970</td>
</tr>
<tr>
<td>max_iid</td>
<td>11600.600000</td>
<td>343.702375</td>
<td>424.143357</td>
<td>609.401375</td>
<td>592.952724</td>
<td>10785</td>
<td>12662</td>
</tr>
<tr>
<td>sum_iid</td>
<td>39154411.600000</td>
<td>131855.985224</td>
<td>162715.896659</td>
<td>233787.207843</td>
<td>227476.942953</td>
<td>38778392</td>
<td>39503646</td>
</tr>
<tr>
<td>avg_iid</td>
<td>3930.421027</td>
<td>13.717449</td>
<td>16.927915</td>
<td>24.321718</td>
<td>23.665238</td>
<td>3893.412851</td>
<td>3966.627774</td>
</tr>
</tbody>
</table>

**Table: flowtime_drive_on**

| count_iid | 38.100000 | 3.318017 | 4.094574 | 5.883009 | 5.724218 | 30 | 47 |
| max_iid | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 | 0 |
| min_iid | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 | 0 |
| sum_iid | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 | 0 |
| avg_iid | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 | 0 |

**Table: percentage_serviced**

| count_iid | 10000.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 10000 | 10000 |
| max_iid | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 | 0 |
| min_iid | 9961.900000 | 3.318017 | 4.094574 | 5.883009 | 5.724218 | 9953 | 9970 |
| sum_iid | 0.996190 | 0.000332 | 0.000409 | 0.000588 | 0.000572 | 0.995300 | 0.997000 |

Generated: Sun Mar 28 19:37:17 2010
## Results

<table>
<thead>
<tr>
<th></th>
<th>Average flow time</th>
<th>Average fraction served</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base case</td>
<td>[7.352, 7.460]</td>
<td>[0.910, 0.915]</td>
</tr>
<tr>
<td>Two pumps</td>
<td>[3.916, 3.944]</td>
<td>[0.996, 0.997]</td>
</tr>
</tbody>
</table>
6 places to queue
### Average queue length

- Minimum: 1.691
- Maximum: 1.770

### Average # pumps busy

- Minimum: 0.841
- Maximum: 0.851

### Average # pumps free

- Minimum: 0.149
- Maximum: 0.159
## Average Flow Time

- Range: [10.518, 10.774]

### Flowtime Depart

<table>
<thead>
<tr>
<th>count_iid</th>
<th>9678.900000</th>
<th>18.247226</th>
<th>22.517854</th>
<th>32.353238</th>
<th>31.479976</th>
<th>9630</th>
<th>9744</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_iid</td>
<td>30066.900000</td>
<td>299.730152</td>
<td>369.879763</td>
<td>531.436441</td>
<td>517.092180</td>
<td>29374</td>
<td>31188</td>
</tr>
<tr>
<td>min_iid</td>
<td>9678.900000</td>
<td>0.001825</td>
<td>0.002252</td>
<td>0.003235</td>
<td>0.003148</td>
<td>0.963000</td>
<td>0.974400</td>
</tr>
<tr>
<td>sum_iid</td>
<td>103039273.400000</td>
<td>1101932.891192</td>
<td>1359832.078492</td>
<td>1953781.721972</td>
<td>1901046.243765</td>
<td>100848139</td>
<td>106400048</td>
</tr>
<tr>
<td>avg_iid</td>
<td>10646.221288</td>
<td>127.439617</td>
<td>157.265911</td>
<td>225.956768</td>
<td>219.857858</td>
<td>10395.074418</td>
<td>11048.810800</td>
</tr>
</tbody>
</table>

## Average Fraction Served

- Range: [0.966, 0.970]

### Flowtime Drive On

<table>
<thead>
<tr>
<th>count_iid</th>
<th>321.100000</th>
<th>18.247226</th>
<th>22.517854</th>
<th>32.353238</th>
<th>31.479976</th>
<th>256</th>
<th>370</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>min_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>sum_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>avg_iid</td>
<td>0.967890</td>
<td>0.001825</td>
<td>0.002252</td>
<td>0.003235</td>
<td>0.003148</td>
<td>0.963000</td>
<td>0.974400</td>
</tr>
</tbody>
</table>

### Percentage Serviced

- Count: 10000.000000, 0.000000, 0.000000, 0.000000, 0.000000, 10000, 10000
- Sum: 9678.900000, 18.247226, 22.517854, 32.353238, 31.479976, 9630, 9744
- Avg: 0.967890, 0.001825, 0.002252, 0.003235, 0.003148, 0.963000, 0.974400

*Generated: Sun Mar 28 19:51:13 2010*
## Results

<table>
<thead>
<tr>
<th></th>
<th>Average flow time</th>
<th>Average fraction served</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base case</td>
<td>[7.352, 7.460]</td>
<td>[0.910, 0.915]</td>
</tr>
<tr>
<td>Two pumps</td>
<td>[3.916, 3.944]</td>
<td>[0.996, 0.997]</td>
</tr>
<tr>
<td>Six places</td>
<td>[10.518, 10.774]</td>
<td>[0.966, 0.970]</td>
</tr>
</tbody>
</table>
faster pump

[2-5] => [1.4-3.5]
### Statistics

<table>
<thead>
<tr>
<th>Name</th>
<th>Avrg</th>
<th>90% Half Length</th>
<th>95% Half Length</th>
<th>99% Half Length</th>
<th>StD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>List_length_dc_gas_station'queue_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_iid</td>
<td>19779.800000</td>
<td>15.691641</td>
<td>19.364153</td>
<td>27.822059</td>
<td>27.071100</td>
<td>19731</td>
<td>19827</td>
</tr>
<tr>
<td>max_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>min_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>avrg_iid</td>
<td>0.388141</td>
<td>0.010423</td>
<td>0.012863</td>
<td>0.018481</td>
<td>0.017982</td>
<td>0.358039</td>
<td>0.419027</td>
</tr>
<tr>
<td>Marking_size_gas_station'fill_up_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_iid</td>
<td>19557.600000</td>
<td>31.383282</td>
<td>38.728306</td>
<td>55.644117</td>
<td>54.142200</td>
<td>19460</td>
<td>19652</td>
</tr>
<tr>
<td>max_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>min_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>avrg_iid</td>
<td>0.600104</td>
<td>0.004608</td>
<td>0.005687</td>
<td>0.008170</td>
<td>0.007950</td>
<td>0.586972</td>
<td>0.612357</td>
</tr>
<tr>
<td>Marking_size_gas_station'pump_free_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_iid</td>
<td>19557.600000</td>
<td>31.383282</td>
<td>38.728306</td>
<td>55.644117</td>
<td>54.142200</td>
<td>19460</td>
<td>19652</td>
</tr>
<tr>
<td>max_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>min_iid</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>avrg_iid</td>
<td>0.399896</td>
<td>0.004608</td>
<td>0.005687</td>
<td>0.008170</td>
<td>0.007950</td>
<td>0.387643</td>
<td>0.413028</td>
</tr>
</tbody>
</table>

- **Average queue length**: [0.378, 0.399]
- **Average # pumps busy**: [0.595, 0.605]
- **Average # pumps free**: [0.395, 0.405]
Average flow time [4.000, 4.065]

Average fraction served [0.976, 0.979]
## Results

<table>
<thead>
<tr>
<th></th>
<th>Average flow time</th>
<th>Average fraction served</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base case</strong></td>
<td>[7.352, 7.460]</td>
<td>[0.910, 0.915]</td>
</tr>
<tr>
<td><strong>Two pumps</strong></td>
<td>[3.916, 3.944]</td>
<td>[0.996, 0.997]</td>
</tr>
<tr>
<td><strong>Six places</strong></td>
<td>[10.518, 10.774]</td>
<td>[0.966, 0.970]</td>
</tr>
<tr>
<td><strong>Faster pump</strong></td>
<td>[4.000, 4.065]</td>
<td>[0.976, 0.979]</td>
</tr>
</tbody>
</table>
Insights obtained from simulation

- Adding a pump significantly reduces the flow time (from approx. 7.4 to approx. 3.9 minutes) and reduces the percentage not served (from approx. 9% to approx. 1%).
- Adding more waiting places significantly increases the flow time (from approx. 7.4 to approx. 10.6 minutes) but reduces the percentage not served (approx. 9% to approx. 3%).
- Installing a faster pump significantly reduces the flow time (from approx. 7.4 to approx. 4.0 minutes) and reduces the percentage not served (from approx. 9% to approx. 3%).
Analytical models versus Simulation models
Example: M/M/1 queue

- arrival rate $\lambda$ (average interarrival time = $1/\lambda$)
- service rate $\mu$ (average interarrival time = $1/\mu$)
- utilization $\rho = \lambda/\mu$
- average nof cases in system $L = \rho/(1 - \rho)$
- average flow time $S = 1/(\mu-\lambda)$

Example:
- $\lambda = 1/100$ and $\mu = 1/50$
- $\rho = 0.5$
- $L = 1$
- $S = 100$
M/M/1 queue

- $\lambda = 1/100$
- $\mu = 1/50$
- $\rho = 0.5$
- $L = 1$
- $S = 100$

- $\lambda = 1/100$
- $\mu = 1/80$
- $\rho = 0.8$
- $L = 4$
- $S = 400$

- $\lambda = 1/100$
- $\mu = 1/99$
- $\rho = 0.99$
- $L = 99$
- $S = 9900$
CPN model with monitors
Creating monitors
Single run

CPN Tools Simulation Performance Report
Net Directory: C:\courses\BIS-2008\CPN\simulation\output\PerfReport.html

Note that these statistics have been calculated for data that is not necessarily independent or identically distributed.

<table>
<thead>
<tr>
<th>Timed statistics</th>
<th>Name</th>
<th>Count</th>
<th>Avrg</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>List_length_dc_example_net_queue_1</td>
<td>2002</td>
<td>0.417271</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Marking_size_example_net_busy_1</td>
<td>2002</td>
<td>0.486339</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Marking_size_example_net_free_1</td>
<td>2002</td>
<td>0.513661</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Untimed statistics</th>
<th>Name</th>
<th>Count</th>
<th>Sum</th>
<th>Avrg</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flow_time</td>
<td>1000</td>
<td>90651000</td>
<td>90.651000</td>
<td>0</td>
<td>517</td>
</tr>
</tbody>
</table>

Simulation steps executed: 4000
Model time: 100321
90% [96.703-4.0159, 96.703+4.0159]

- $\lambda = \frac{1}{100}$
- $\mu = \frac{1}{50}$
- $\rho = 0.5$
- $L = 1$
- $S = 100$
<table>
<thead>
<tr>
<th>Name</th>
<th>Avg</th>
<th>90% Half Length</th>
<th>95% Half Length</th>
<th>99% Half Length</th>
<th>Std</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flow_time</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_id</td>
<td>1000.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>max_id</td>
<td>1845.200000</td>
<td>320.566238</td>
<td>395.592378</td>
<td>568.379854</td>
<td>553.038435</td>
<td>1211</td>
<td>3088</td>
</tr>
<tr>
<td>min_id</td>
<td>0.400000</td>
<td>0.299328</td>
<td>0.369383</td>
<td>0.530723</td>
<td>0.516398</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sum_id</td>
<td>424854.700000</td>
<td>47563.02127</td>
<td>58694.792140</td>
<td>84331.597902</td>
<td>82055.362490</td>
<td>293109</td>
<td>540346</td>
</tr>
<tr>
<td>avg_id</td>
<td>424.854700</td>
<td>47.563021</td>
<td>58.694792</td>
<td>84.331598</td>
<td>82.055362</td>
<td>293.109</td>
<td>540.346</td>
</tr>
<tr>
<td><strong>List_length.dc_example.net.queue_1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_id</td>
<td>2002.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>2002</td>
<td>2002</td>
</tr>
<tr>
<td>max_id</td>
<td>19.800000</td>
<td>3.040301</td>
<td>3.751860</td>
<td>5.390604</td>
<td>5.245104</td>
<td>15</td>
<td>33</td>
</tr>
<tr>
<td>min_id</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>avg_id</td>
<td>3.526256</td>
<td>0.491455</td>
<td>0.606477</td>
<td>0.871374</td>
<td>0.847855</td>
<td>2.214570</td>
<td>4.693034</td>
</tr>
<tr>
<td><strong>Marking_size.example.net.busy_1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_id</td>
<td>2002.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>2002</td>
<td>2002</td>
</tr>
<tr>
<td>max_id</td>
<td>1.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>min_id</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>avg_id</td>
<td>0.821571</td>
<td>0.014726</td>
<td>0.018172</td>
<td>0.026109</td>
<td>0.025405</td>
<td>0.792455</td>
<td>0.867871</td>
</tr>
<tr>
<td><strong>Marking_size.example.net.free_1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>count_id</td>
<td>2002.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>2002</td>
<td>2002</td>
</tr>
<tr>
<td>max_id</td>
<td>1.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>min_id</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>avg_id</td>
<td>0.178429</td>
<td>0.014726</td>
<td>0.018172</td>
<td>0.026109</td>
<td>0.025405</td>
<td>0.132129</td>
<td>0.207545</td>
</tr>
<tr>
<td>Name</td>
<td>Avrg</td>
<td>90% Half Length</td>
<td>95% Half Length</td>
<td>99% Half Length</td>
<td>StD</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>count_id</td>
<td>1000.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1000</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>max_id</td>
<td>4899.700000</td>
<td>1252.193365</td>
<td>1545.259897</td>
<td>2220.201002</td>
<td>2160.274471</td>
<td>2216</td>
<td>9339</td>
</tr>
<tr>
<td>min_id</td>
<td>3.300000</td>
<td>2.186834</td>
<td>2.69864</td>
<td>3.877365</td>
<td>3.772709</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>sum_id</td>
<td>1709043.500000</td>
<td>507550.790075</td>
<td>826339.272858</td>
<td>899912.748359</td>
<td>875622.763150</td>
<td>985796</td>
<td>3450629</td>
</tr>
<tr>
<td>avrg_id</td>
<td>1709.043500</td>
<td>507.550790</td>
<td>826.339273</td>
<td>899.912748</td>
<td>875.622763</td>
<td>958.796000</td>
<td>3460.629000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Flow_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Avrg</td>
</tr>
<tr>
<td>count_id</td>
<td>1000.000000</td>
</tr>
<tr>
<td>max_id</td>
<td>4899.700000</td>
</tr>
<tr>
<td>min_id</td>
<td>3.300000</td>
</tr>
<tr>
<td>sum_id</td>
<td>1709043.500000</td>
</tr>
<tr>
<td>avrg_id</td>
<td>1709.043500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List_length_dc_example_net`queue_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>count_id</td>
</tr>
<tr>
<td>max_id</td>
</tr>
<tr>
<td>min_id</td>
</tr>
<tr>
<td>avrg_id</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marking_size_example_net`busy_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>count_id</td>
</tr>
<tr>
<td>max_id</td>
</tr>
<tr>
<td>min_id</td>
</tr>
<tr>
<td>avrg_id</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marking_size_example_net`free_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>count_id</td>
</tr>
<tr>
<td>max_id</td>
</tr>
<tr>
<td>min_id</td>
</tr>
<tr>
<td>avrg_id</td>
</tr>
</tbody>
</table>

Note deviations.
Why?
Conclusion analysis

- Analysis is typically model-driven to allow e.g. what-if questions.
- Models of both operational processes and/or the information systems can be analyzed.
- Types of analysis:
  - validation (interactive simulation/gaming)
  - verification (state-space analysis, place and transition invariants, siphons, traps, etc.)
  - performance analysis (simulation)