Coloured Petri Net Example

Vincent J. Maccio

Department of Computing and Software
McMaster University
Hamilton, Ontario, Canada
The Bank Courier Problem

- A set of trucks drives around the city collecting cheques and mail
- Locations are hubs, branches, main, processing centre
- Model using a CPN, built up from SADT models
### Data Format

<table>
<thead>
<tr>
<th>Route</th>
<th>Time</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7:45</td>
<td>Start</td>
<td>Boston</td>
</tr>
<tr>
<td></td>
<td>8:20</td>
<td>Branch</td>
<td>B483</td>
</tr>
<tr>
<td></td>
<td>9:10</td>
<td>Branch</td>
<td>B488</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch</th>
<th>Time</th>
<th>Cheques</th>
<th>Mail</th>
<th>ATM</th>
</tr>
</thead>
<tbody>
<tr>
<td>B483</td>
<td>8:20</td>
<td>360</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>12:10</td>
<td>620</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Structured Analysis and Design Technique (SADT)

- Describe complex systems as a series of functions
- A variable can be one of four:
  - Inputs represent data or consumables that are needed by the activity
  - Outputs represent data or products that are produced by the activity
  - Controls represent commands which influence the execution of an activity but are not consumed
  - Mechanisms identify the means, components or tools used to accomplish the activity
The Colours

colour Kind = with ToMain | Hub | Branch | FromMain | Start;
colour Location = string;
colour Stop = record Time:TIME * kind:Kind * loc:Location;
colour Route = list Stop;
colour TruckId = int;
colour RouteSchedule = product TruckId * Route timed;
colour Cheques = int;
colour Mail = bool;
colour ATM = bool;
colour Truck = product TruckId * Cheques * Mail * ATM;
colour Trucks = list Truck

colour Load = product Location * Cheques * Mail * ATM;
colour Loads = list Load;
Colour Example

<table>
<thead>
<tr>
<th>Route</th>
<th>Time</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7:45</td>
<td>Start</td>
<td>Boston</td>
</tr>
<tr>
<td></td>
<td>8:20</td>
<td>Branch</td>
<td>B483</td>
</tr>
<tr>
<td></td>
<td>9:10</td>
<td>Branch</td>
<td>B488</td>
</tr>
</tbody>
</table>

Becomes...

\((4, \begin{cases} \text{time}=745, \text{kind}=\text{Start}, \text{loc}=\text{“Boston”} \\ \text{time}=820, \text{kind}=\text{Start}, \text{loc}=\text{“B483”} \\ \text{time}=910, \text{kind}=\text{Start}, \text{loc}=\text{“B488”} \end{cases})\)@ 745
Vincent J. Maccio Coloured Petri Net Example

Routing Schedule

If rest <> []
then 1` (truck, rest)
else empty

[(c1, m1, a1) = TruckLoad(trucks, truck)
(c2, m2, a2) = BranchLoad(loads, branch)]

Drive to Next Branch for Pickup

@+#Time(hd rest)-time()

UpdateTrucks(trucks,
(truck, c1+c2,
m1 or m2, a1 or a2))

UpdateLoads(loads, branch)

Trucks

I2

C1

Loads

loads

M1

trucks
Conclusion

• An interesting design process for tackling problems
• The project was a success on the small scale
• However, the implications have wider bearing on a larger scale