Abdullah Al-Faify Student # 5894259

SYSC 5104

Methodologies for Discrete Event Modeling and Simulation

Fall 2011

Assignment # 2 - Report

Modeling and Simulation of a Traffic Roundabout

0. Background: Traffic-related Problem, Roundabouts:

Roundabouts are lane or multilane intersections so organized that traffic can flow in one direction (counter-clockwise) around a central island. There are many kinds of roundabouts. Some of them are very simple while others are large and complicated. The most complicated one is called "The **Magic Roundabout"** in Swindon, England was constructed in 1972 and consists of five mini-roundabouts arranged in a circle shown in figure 1.

Figure 1: The Magic Roundabout" in Swindon, England

Because of the increasing number of vehicles, streets become crowded and the locations where vehicles meet from different directions such as intersections and roundabouts control the flow of traffic, and sometimes these points work as bottleneck for traffic flow. Modeling and simulating roadways using different scenarios –traffic signals and/or signs- becomes essential to avoid paralysis in movements as shown in the following picture.

Part I: Conceptual Model Description:

In this assignment, I will model and test a typical roundabout (Figure 2) with the same configuration that is used in [1][1], where eight single lanes are connected to a single-lane ring. In the ring, vehicles move counter-clockwise. They move into the system through the odd-numbered boundaries. After travelled various distances in the ring, vehicles move out of the system through the even-numbered boundaries, see figure 3.

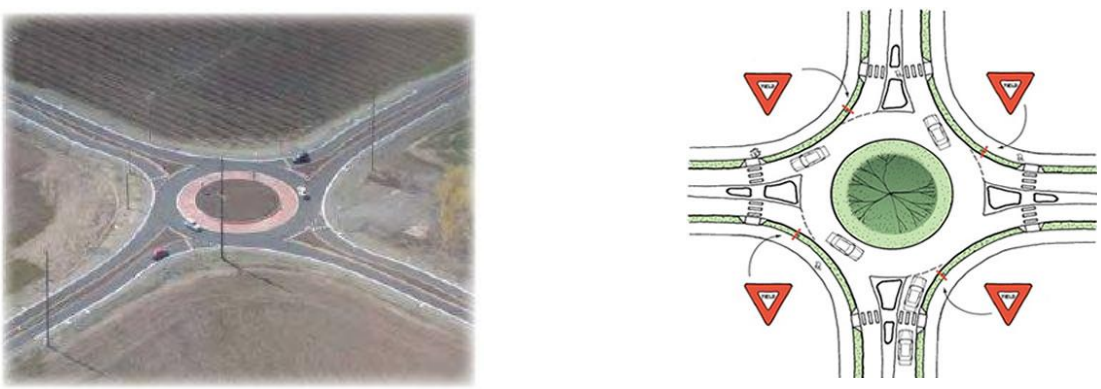

Figure 2: Typical a Roundabout

[1] [1] Cellular Automata for a Traffic Roundabout, by Ding-wei Huang

In [1], the author used four T-shaped intersections to model a typical roundabout. However, I may use a shape that is seen in figure 4 to be closer to the reality. I am, also, planning to show two different animations for the system: one if there is no traffic signs to control of vehicle flow at the roundabout, and the other when there are yield signs.

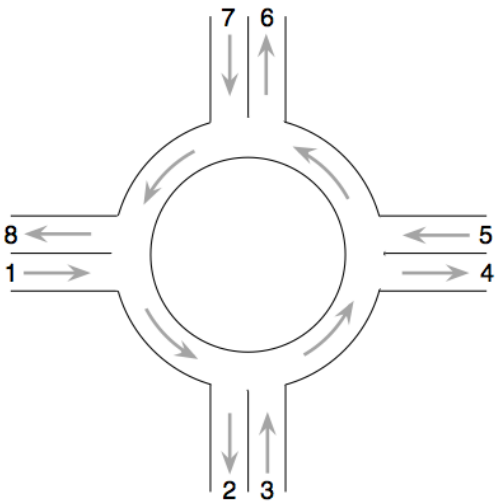

Figure 3: Model Configurations

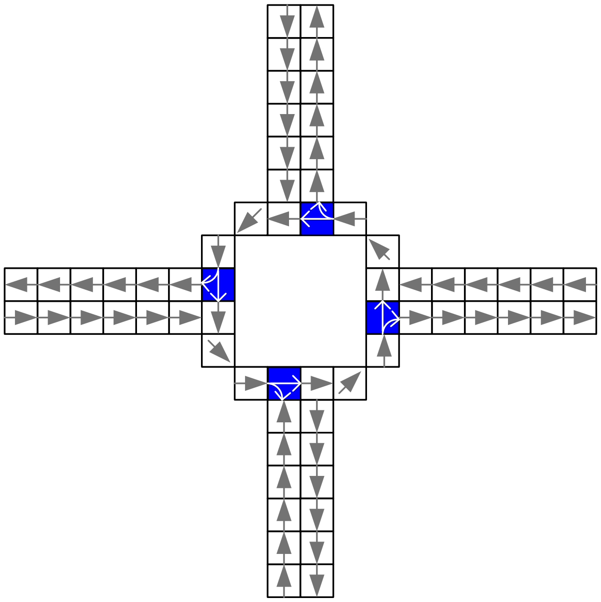

Figure 4: The Model and the Direction Flow for Each Cell

Part II: Cell-DEVS Formal Specification:

i. <u>Cell-DEVS Atomic Model Specification</u>

The model is divided into 25 zones. A cell in each zone behaves differently. However, the following is the formal specification for the Cell-DEVS the model:

$CD = < X, Y, I, S, \theta, N, d, \delta_{int}, \delta_{ext}, \tau, \lambda, D >$

$X = \emptyset$

$Y = \emptyset$

$S = \{ 0, 1 , 2\}$

N = neighborhood = { (-2, 0), (-2,1),
(-1,-2), (-1,-1), (-1,0), (-1,1),
(0,-2), (0,-1), (0,0), (0,1), (0, 2),
(1,-1), (1,0), (1,1), (1,2)
(2,-1), (2,0) }

d = 100 ms

τ:N→S : because there are a lot of rules according to each zone, τ is defined by the rules described in the next part, Cell-DEVS Coupled Model Specification.

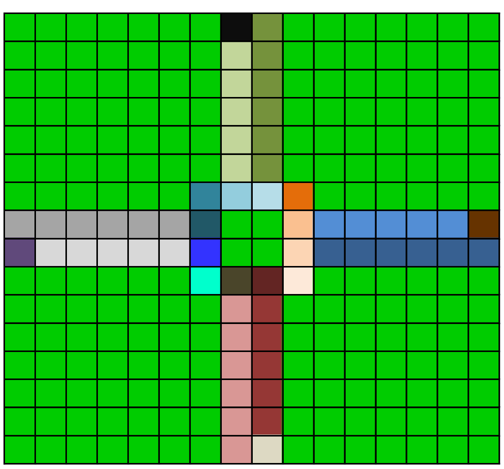

Figure 5: Traffic roundabout model. Each colour represents a zone

		(-2,0)	(-2,1)	
(-1-2)	(-1,-1)	(-1,0)	(-1,1)	
(0,-2)	(0,-1)	(0,0)	(0,1)	(0,2)
	(1,-1)	(1,0)	(1,1)	(1,2)
	(2,-1)	(2,0)		

A cell's neighbourhood

ii. Cell-DEVS Coupled Model Specification

Following is coupled model definition:

```
[top]
components : roundabout

[roundabout]
type : cell
dim : (16,16)
delay : transport
defaultDelayTime : 100
border : nowrapped
neighbors : roundabout(-2,0) roundabout(-2,1)
neighbors : roundabout(-1,-2) roundabout(-1,-1) roundabout(-1,0) roundabout(-1,1)
neighbors : roundabout(0,-2) roundabout(0,-1) roundabout(0,0) roundabout(0,1)
roundabout(0,2)
neighbors : roundabout(1,-1) roundabout(1,0) roundabout(1,1) roundabout(1,2)
neighbors : roundabout(2,-1) roundabout(2,0)
```

The model definition also defines the initial traffic roundabout by specifying the states of the cells of the model as 1's, 0's, or 2's. For example, the following is a definition of a roundabout which can be solved using the Cell-DEVS model:

```
initialvalue : 0
initialrowvalue : 0 2222222002222222
initialrowvalue : 1 2222222002222222
initialrowvalue : 2 2222222112222222
initialrowvalue : 3 2222222002222222
initialrowvalue : 4 2222222102222222
initialrowvalue : 5 2222222002222222
initialrowvalue : 6 2222220000222222
initialrowvalue : 7 0010001220010001
initialrowvalue : 8 0010000220000100
initialrowvalue : 9 2222220000222222
initialrowvalue : 10 2222222002222222
initialrowvalue : 11 2222222002222222
initialrowvalue : 12 2222222002222222
initialrowvalue : 13 2222222102222222
initialrowvalue : 14 2222222002222222
initialrowvalue : 15 2222222012222222


localtransition : roundabout-rule

zone : R-lane-to-roundabout-rule { (7,10)..(7,14) }
zone : R-lane-from-roundabout-rule { (8,10)..(8,15) }
zone : U-lane-to-roundabout-rule { (1,7)..(5,7) }
zone : U-lane-from-roundabout-rule { (0,8)..(5,8) }
zone : L-lane-to-roundabout-rule { (8,1)..(8,5) }
zone : L-lane-from-roundabout-rule { (7,0)..(7,5) }
```

```
zone : D-lane-to-roundabout-rule { (10,8)..(14,8) }
zone : D-lane-from-roundabout-rule { (10,7)..(15,7) }
zone : EU-inside-roundabout-rule { (7,9) }
zone : NE-inside-roundabout-rule { (6,9) }
zone : NR-inside-roundabout-rule { (6,8) }
zone : NL-inside-roundabout-rule { (6,7) }
zone : NW-inside-roundabout-rule { (6,6) }
zone : WU-inside-roundabout-rule { (7,6) }
zone : WD-inside-roundabout-rule { (8,6) }
zone : SW-inside-roundabout-rule { (9,6) }
zone : SL-inside-roundabout-rule { (9,7) }
zone : SR-inside-roundabout-rule { (9,8) }
zone : SE-inside-roundabout-rule { (9,9) }
zone : ED-inside-roundabout-rule { (8,9) }
zone : gen-cars-in-R-lane-to-roundabout-rule { (7,15) }
zone : gen-cars-in-U-lane-to-roundabout-rule { (0,7) }
zone : gen-cars-in-L-lane-to-roundabout-rule { (8,0) }
zone : gen-cars-in-D-lane-to-roundabout-rule { (15,8) }
```

The rules of the model differ from zone to zone. The following figure illustrates the name of each zone follows by rules. When I set the rules inside roundabout zones, I consider a safety distance between cars which represent by empty (0) cell.

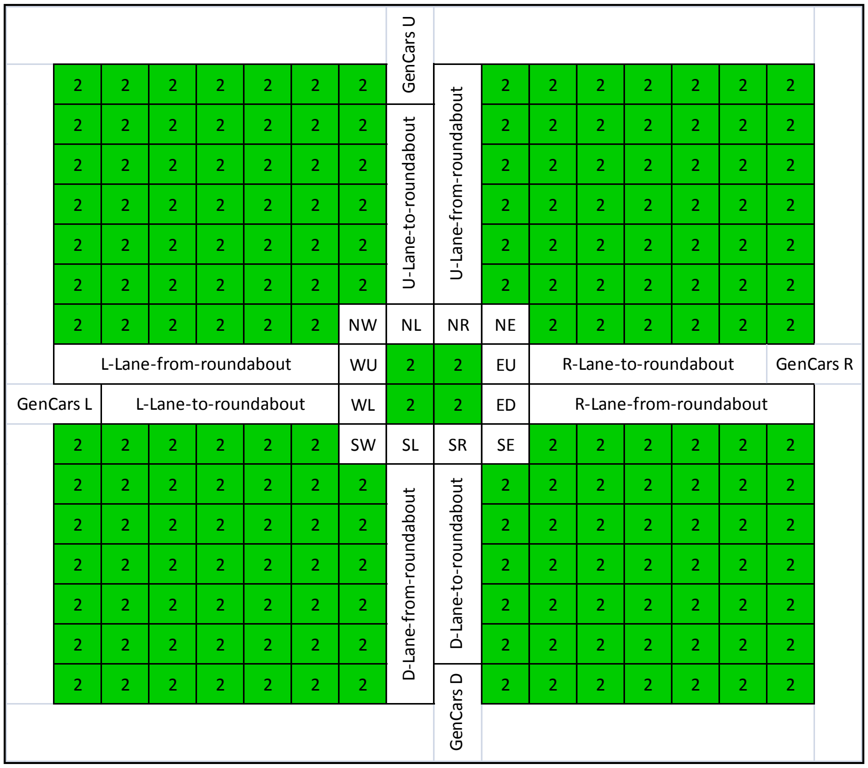

Figure 6: Names of the zones in Traffic roundabout model.

```
[roundabout-rule]
rule : 2 100 { (0,0) = 2 }
rule : 0 100 { t }
[R-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 and (-1,-1) = 2 }
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 and (-1,-1) !=2 and (1,-1) = 0 and (2,-1) = 0
}
rule : 1 100 { (0,0) = 0 and (0,1) = 1 }
rule : 1 100 { (0,0) = 1 and (0,-1) = 1 }
rule : 1 100 { (0,0) = 1 and (1,-1) = 1 }
rule : 1 100 { (0,0) = 1 and (2,-1) = 1 }
rule : 0 100 { t }
[R-lane-from-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,1) = 0 }
rule : 1 100 { (0,0) = 0 and (0,-1) = 1 and (1,-1) = 2 }
rule : 1 100 { (0,0) = 0 and (0,-1) = 1 and (1,-1) != 2 and ( random > 0.5 ) }
rule : 0 100 { t }
[U-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (1,0) = 0 and (1,-1) = 2 }
rule : 0 100 { (0,0) = 1 and (1,0) = 0 and (1,-1) != 2 and (1,1) = 0 and (1,2) = 0 }
rule : 1 100 { (0,0) = 0 and (-1,0) = 1 }
rule : 1 100 { (0,0) = 1 and (1,0) = 1 }
rule : 1 100 { (0,0) = 1 and (1,1) = 1 }
rule : 1 100 { (0,0) = 1 and (1,2) = 1 }
rule : 0 100 { t }
[U-lane-from-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 }
rule : 1 100 { (0,0) = 0 and (1,0) = 1 and (1,1) = 2 }
rule : 1 100 { (0,0) = 0 and (1,0) = 1 and (1,1) != 2 and ( random > 0.5 ) }
rule : 0 100 { t }
[L-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,1) = 0 and (1,1) = 2 }
rule : 0 100 { (0,0) = 1 and (0,1) = 0 and (1,1) != 2 and (-1,1) = 0 and (-2,1) = 0 }
rule : 1 100 { (0,0) = 0 and (0,-1) = 1 }
rule : 1 100 { (0,0) = 1 and (0,1) = 1 }
rule : 1 100 { (0,0) = 1 and (-1,1) = 1 }
rule : 1 100 { (0,0) = 1 and (-2,1) = 1 }
rule : 0 100 { t }
[L-lane-from-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 }
rule : 1 100 { (0,0) = 0 and (0,1) = 1 and (-1,1) = 2 }
rule : 1 100 { (0,0) = 0 and (0,1) = 1 and (-1,1) != 2 and ( random > 0.5 ) }
rule : 0 100 { t }
[D-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 and (-1,1) = 2 }
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 and (-1,1) != 2 and (-1,-1) = 0 and (-1,-2) =
0 }
rule : 1 100 { (0,0) = 0 and (1,0) = 1 }
rule : 1 100 { (0,0) = 1 and (-1,0) = 1 }
rule : 1 100 { (0,0) = 1 and (-1,-1) = 1 }
rule : 1 100 { (0,0) = 1 and (-1,-2) = 1 }
rule : 0 100 { t }
[D-lane-from-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (1,0) = 0 }
rule : 1 100 { (0,0) = 0 and (-1,0) = 1 and (-1,-1) = 2 }
rule : 1 100 { (0,0) = 0 and (-1,0) = 1 and (-1,-1) != 2 and ( random > 0.5 ) }
rule : 0 100 { t }
```

```
[EU-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 }
rule : 0 100 { (1,0) != 1 and (0,0) = 0 and (0,1) = 1 and (2,0) = 1 }
rule : 1 100 { (1,0) = 1 and (0,0) = 0 and ( random <= 0.5 ) }
rule : 1 100 { (1,0) != 1 and (0,0) = 0 and (0,1) = 1 }
rule : 0 100 { t }
[NE-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 }
rule : 1 100 { (1,0) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[NR-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 }
rule : 1 100 { (0,1) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[NL-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 }
rule : 0 100 { (0,1) != 1 and (0,0) = 0 and (-1,0) = 1 and (0,2) = 1 }
rule : 1 100 { (0,1) = 1 and (0,0) = 0 and ( random <= 0.5 ) }
rule : 1 100 { (0,1) != 1 and (0,0) = 0 and (-1,0) = 1 }
rule : 0 100 { t }
[NW-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 }
rule : 1 100 { (0,1) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[WU-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (1,0) = 0 }
rule : 1 100 { (-1,0) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[WD-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (1,0) = 0 }
rule : 0 100 { (-1,0) != 1 and (0,0) = 0 and (0,-1) = 1 and (-2,0) = 1 }
rule : 1 100 { (-1,0) = 1 and (0,0) = 0 and ( random <= 0.5 ) }
rule : 1 100 { (-1,0) != 1 and (0,0) = 0 and (0,-1) = 1 }
rule : 0 100 { t }
[SW-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (1,0) = 0 }
rule : 1 100 { (-1,0) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[SL-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,1) = 0 }
rule : 1 100 { (0,-1) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[SR-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,1) = 0 }
rule : 0 100 { (0,-1) != 1 and (0,0) = 0 and (1,0) = 1 and (0,-2) = 1 }
rule : 1 100 { (0,-1) = 1 and (0,0) = 0 and ( random <= 0.5 ) }
rule : 1 100 { (0,-1) != 1 and (0,0) = 0 and (1,0) = 1 }
rule : 0 100 { t }
[SE-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,1) = 0 }
rule : 1 100 { (0,-1) = 1 and (0,0) = 0 }
rule : 0 100 { t }
[ED-inside-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 }
rule : 1 100 { (1,0) = 1 and (0,0) = 0 }
rule : 0 100 { t }
```

```
##################################Cars-Generator##############
[gen-cars-in-R-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,-1) = 0 }
rule : 1 100 { (0,0) = 0 and (0,-1) = 0 and ( random <= 0.5 ) }
rule : 0 100 { t }
[gen-cars-in-U-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (1,0) = 0 }
rule : 1 100 { (0,0) = 0 and (1,0) = 0 and ( random <= 0.4 ) }
rule : 0 100 { t }
[gen-cars-in-L-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (0,1) = 0 }
rule : 1 100 { (0,0) = 0 and (0,1) = 0 and ( random <= 0.4 ) }
rule : 0 100 { t }
[gen-cars-in-D-lane-to-roundabout-rule]
rule : 0 100 { (0,0) = 1 and (-1,0) = 0 }
rule : 1 100 { (0,0) = 0 and (-1,0) = 0 and ( random <= 0.3 ) }
rule : 0 100 { t }
```

iii. Implementation and Testing

First of all, there is a small problem that doesn't allow the model to work as real situation which is using random numbers to simulate getting cars out of the ring of a roundabout. This problem causes multiple rules will be valid or no one of them will be. However, this problem didn't affect that much on showing the behaviour of the model.

As I mentioned in part I, the model was implemented in two different cases.

- First one, when there are no yield signs at a roundabout (TrafficRoundaboutWithoutYieldSigns.ma). Some rules changed to simulate this situation. The safety distance that is assumed disappeared, and the model's animation shows that there are many accidents at the points where cars that coming to roundabout and cars inside the roundabout meet.
- Second implementation is when there are yield signs that give cars inside a roundabout right to move and control traffic coming toward the ring of a roundabout (TrafficRoundaboutWithYieldSings.ma).

The model behaved nicely as shown in a video file (TrafficRounaboutWithYieldSigns.avi).

- Other implementations were executed with same second case implementation rules but with different initial values. The models behaved similarly.

Part III: Conclusion:

Using Cell-DEVS to model and simulate traffic behavior is a very interesting area, but it has a bit of inability to deal with a stochastic model properly. In my point view, integrate Cell-DEVS with DEVS for modeling traffic behavior is necessary to make a model closer to reality. “It would be interesting to further develop a mean field theory for the critical behavior of congestion” [1]. This small typical roundabout model introduces more complicated and interesting behaviors in such asymmetrical cases that can be demonstrating by CD++ easily and visualize the output using the associated tools.

Part IV: References

[1] Cellular Automata for a Traffic Roundabout, by Ding-wei Huang