

Conceptual Modelling Assignment – <Name, UPI>

Problem Understanding

Identification of Modelling and General Objectives

Modelling Objectives

General Objectives

Defining Input Factors

Defining Output Responses

Model Content

Identifying Entities

Drawing Behavioural Paths

Figure 1: Person Behavioural Path

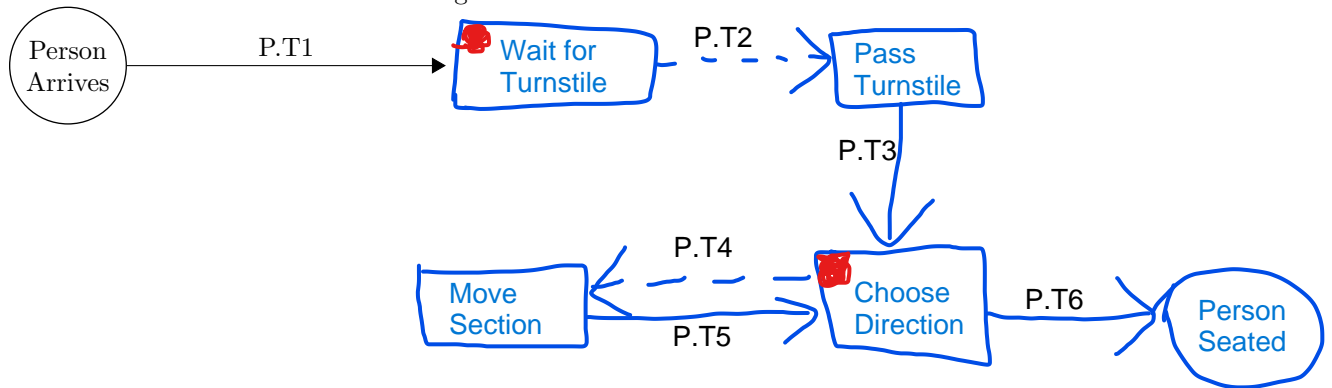


Figure 2: Turnstile Behavioural Path



Model Control – Defining Logic

Logic On Start Wait for Turnstile

Triggered by	Entity Person P
1:	if (any Turnstile T with T.CurrentActivity = T.Wait for Person) then
2:	SELECT valid Turnstile T
3:	Wait for Person.End with T
4:	TRANSITION P.Wait for Turnstile.End to P.Pass Turnstile with T
5:	TRANSITION T.Wait for Person.End to T.Pass Turnstile with P
6:	Pass Turnstile.Start with P and T
7:	end if
8:	

Logic On Start Wait for Person

Triggered by	Entity Turnstile T
1:	if (any Person P with P.CurrentActivity = P.Wait for Turnstile) then
2:	SELECT valid Person P
3:	Wait for Turnstile.End with P
4:	TRANSITION P.Wait for Turnstile.End to P.Pass Turnstile with T
5:	TRANSITION T.Wait for Person.End to T.Pass Turnstile with P
6:	Pass Turnstile.Start with P and T
7:	end if
8:	

On Start Choose Direction

Triggered by	Person P
1: if P.CurrentSection = P.Section then	
2: Choose Direction.End with P	
3: TRANSITION P.T6 Choose Direction.End to Person Seated with P	
4: else	
5: if P.FirstChoice then	
6: if rand()<0.5 then P.Direction = "up" else P.Direction = "down"	
7: else if (P.CurrentSection = "A") AND (P.Direction = "down") then	
8: // Crossing "A" to "X", find best direction from distance	
9: if (P.Section - "A") < ("X" - P.Section) then	
10: P.Direction = "up"	
11: else	
12: P.Direction = "down"	
13: end if	
14: else if (P.CurrentSection = "X") AND (P.Direction = "up") then	
15: // Crossing "X" to "A", find best direction from distance	
16: if (P.Section - "A") > ("X" - P.Section) then	
17: P.Direction = "down"	
18: else	
19: P.Direction = "up"	
20: end if	
21: else	
22: if then P.Section > P.CurrentSection then	
23: P.Direction = "up"	
24: else //	
25: P.Direction = "down"	
26: end if	
27: end if	
28: if then P.CurrentSection has < 100 people then	
29: TRANSITION P.T4 Choose Direction.End to Move Section with P	
30: at time+10	
31: else	
32: TRANSITION P.T4 Choose Direction.End to Move Section with P	
33: at time+9+1000*(400*10^-6*(N-100))	
34: end if	
35: end if	

Model Data

Data	Source	Identification	Input	Output
Time to get seated	Value	Lookup	End Time	Value of time at end of simulation
Time to pass turnstile	Function	Calculate	Person	Random Uniform Variate (2-12s)
Time taken to move through section	Function	Calculate	People in Section N	Function of N (defined below)

$$T = \begin{cases} 60 & N < 100 \\ 59 + 1000^{675 \times 10^{-6}(N-100)} & N \geq 100 \end{cases}$$

Model Entities

Note that default attributes CurrentStart and CurrentActivity are omitted for brevity.

Person	Type	Active
	Attributes – default value or range in []	[] ArrivalTime
		[] ID
		[] Section [NA]
		[] CurrentSection [NA]
Turnstile	Type	Active
	Attributes	[] ID
		[] Gate
Section	Type	Passive
	Attributes	ID [A-X] [] N [0] (number of people in section)
Gate	Type	Passive
	Attributes	[] ID [N, S, E or W]

Model Transitions

Transitions	From Event	To Event
P.T1	Person Arrives	Wait for TurnStile
P.T2	Wait for Turnstile	Pass Turnstile
P.T3	Pass Turnstile	Choose Direction
P.T4	Choose Direction	Move Section
P.T5	Move Section	Choose Direction
P.T6	Choose Direction	Person Seated
T.T1	Turnstile Created	Wait for Person
T.T2	Wait For Person	Pass Turnstile
T.T3	Pass Turnstile	Wait for Person

Model Activities

Wait for Turnstile	Participants		
	Start Event	Type	
		State Change	1:
	End Event	Type	
		State Changes	1: 2: // TRANSITION ??? is determined by logic
	Participants		
	Start Event	Type	
		State Change	1:
	End Event	Type	
		State Changes	1: 2: 3:

	Participants		
	Start Event	Type	
		State Changes	1:
	End Event	Type	
		State Changes	1: // <i>TRANSITION ??? or ??? determined by logic</i>
	Participants		
	Start Event	Type	Controlled
		State Changes	1: 2:
	End Event	Type	
		State Changes	1: 2: 3:
	Participants		
	Start Event	Type	
		State Change	1:
	End Event	Type	Controlled
		State Changes	1: 2: //

Model Events

Simulation Start	Participant	None
	Type	
	State Changes	1: 2: 3: 4: 5:
Person Arrives	Participant	Person (P), Turnstile (T)
	Type	
	State Changes	1: 2: 3: 4:

	Participant	
	Type	
	State Changes	1:
	Participant	
	Type	
	State Changes	1: $T.ID = \max(U.ID \text{ for } U \text{ in Turnstiles}) + 1$ // <i>Get next ID</i> 2: 3:
Simulation Finish	Participant	None
	Type	Scheduled
	State Changes	1: for $T \in \text{Turnstiles}$ do 2: Calculate statistics for T 3: end for