Daniel Clark, 343733502, dcla189

Conceptual Modelling Assignment – <Name, UPI>

Problem Understanding

wegqwegsdfgvb

Identification of Modelling and General Objectives

Modelling Objectives

aowujfboa

General Objectives

We wish to achieve a representative simulation of the real processes in a stadium. We also wish for this to happen in a time-efficient and computationally-efficient manner, and which can be easily adjusted if required.

Defining Input Factors

Number of people entering the stadium [50,000]
Number of sections in the stadium [24]
Number of turnstiles per gate to the stadium [4]

Defining Output Responses

The amount of time that it takes for 50,000 people to get seated within the stadium.

Model Content

Identifying Entities

Drawing Behavioural Paths

Figure 1: Person Behavioural Path

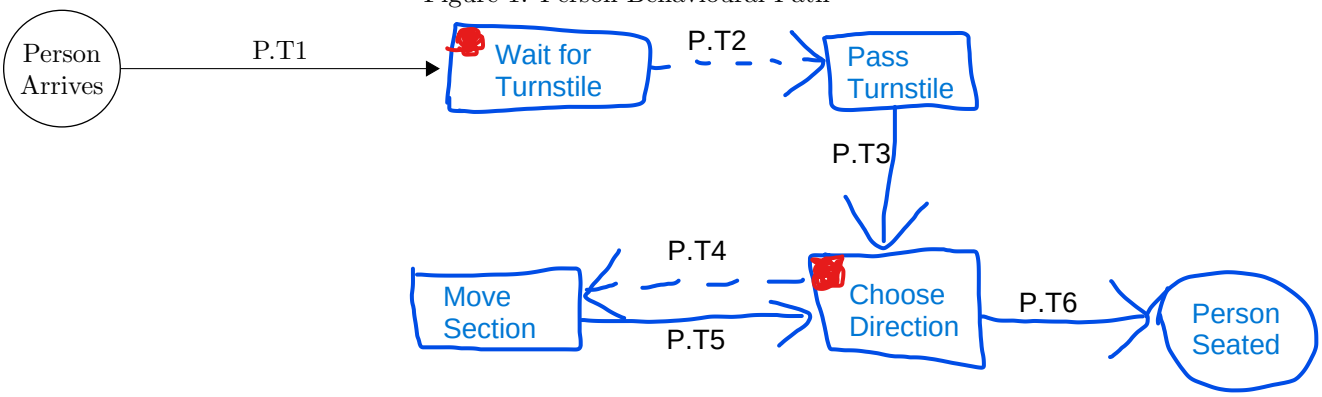
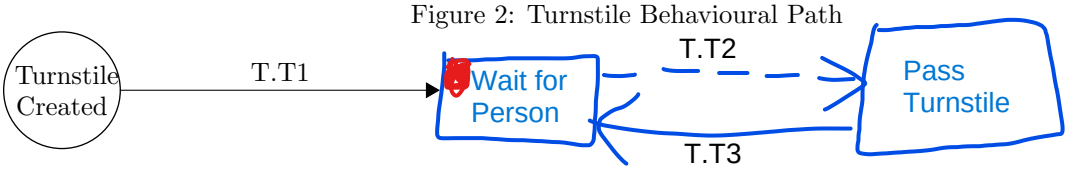


Figure 2: Turnstile Behavioural Path



Model Control – Defining Logic

Logic On Start Wait for Turnstile

Triggered by	Entity Person P

```
1: if (any Turnstile T with T.CurrentAcivity = T.Wait for Person and P.Gate = T.Gate) then
2:     SELECT valid Turnstile T
3:     Wait for Person.End with T ???
4:     TRANSITION P.T2 Wait for Turnstile.End to Pass Turnstile with T
5:     TRANSITION T.T2 Wait for Person.End to 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.CurrentAcivity = P.Wait for Turnstile) then
2:     SELECT valid Person P
3:     Wait for Turnstile.End with P
4:     TRANSITION P.T2 Wait for Turnstile.End to P.Pass Turnstile with T
5:     TRANSITION T.T2 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
                   // Crossing "A" to "X", find best direction from distance
8:                 if (P.Section - "A") < ("X" - P.Section) then
9:                         P.Direction = "up"
10:                else
11:                        P.Direction = "down"
12:                end if
13:        else if (P.CurrentSection = "X") AND (P.Direction = "up") then
                   // Crossing "X" to "A", find best direction from distance
14:                if (P.Section - "A") > ("X" - P.Section) then
15:                        P.Direction = "down"
16:                else
17:                        P.Direction = "up"
18:                end if
19:        else
20:                if  then P.Section > P.CurrentSection then
21:                        P.Direction = "up"
22:                else//
23:                        P.Direction = "down"
24:                end if
25:        end if
26:        if  then P.CurrentSection.N < 100 then
27:                TRANSITION P.T4 Choose Direction.End to Move Section with P
28:        else            with Time = 10
29:                TRANSITION P.T4 Choose Direction.End to Move Section with P
30:        end if          with Time = 9+1000^( 400*10^-6*(N-100) )
31: end if
```

Model Data

Data	Source	Identification	Input	Output
Interarrival Times	Value	Lookup	End Time	Value of time at end of simulation
Time to pass turnstile	Function	Calculate	Person?	Random Uniform Variate
Time taken to move through section	Function	(2-12s) 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 [] Gate [] Section [NA] [] CurrentSection [NA] [] Direction ["up", "down"]
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**		Person P
	Start Event	**Type**	Scheduled
		State Change	1: ?????
	End Event	**Type**	Controlled
		State Changes	1: 2: *// TRANSITION ??? is determined by logic*
Wait for Person	**Participants**		Turnstile T
	Start Event	**Type**	Scheduled
		State Change	1: ?????
	End Event	**Type**	Controlled
		State Changes	1: 2: 3:

Pass Turnstile	**Participants**		Person P, Turnstile T
	Start Event	**Type**	Controlled
		State Changes	1: ???
	End Event	**Type**	Scheduled
		State Changes	1: *// TRANSITION ??? or ??? determined by logic*
Move Section	**Participants**		Person P
	Start Event	**Type**	Controlled
		State Changes	1: ??? 2: ???
	End Event	**Type**	Scheduled
		State Changes	1: ??? 2: ??? 3: ???
Choose Direction	**Participants**		Person P
	Start Event	**Type**	Controlled
		State Change	1: ???
	End Event	**Type**	~~Controlled~~ Scheduled
		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 for U in Turnstiles) + 1 // *Get next ID* 2: 3:
Simulation Finish	**Participant**	None
	Type	Scheduled
	State Changes	1: **for** T ∈ Turnstiles **do** 2: Calculate statistics for T 3: **end for**