Daniel Clark, 343733502, dcla189

Conceptual Modelling Assignment – <Name, UPI>

Problem Understanding

A Stadium has requested that we design a conceptual model for the entry of 50,000 people into their stadium. The stadium has 24 evenly-spread sections, and has 4 entry gates spread evenly around the ground. However, the stadium has zero signage, so people entering the stadium will not enter in the most efficient way. People will arrive at a random gate, turn a random direction and use only their knowledge of the alphabet to navigate to their correct section.

People will arrive to the stadium following an exponential distribution. They arrive to each turnstile with a mean interarrival time of 5 seconds, so arrive at a rate of 0.2/s . This means throughout the whole stadium (which has 80 turnstiles) they arrive at a rate of 16 people/second.

The time it takes for a person to progress through a turnstile can be modelled by a triangular distribution, having a minimum of 2s, a mode of 5s, and a maximum of 12s.

The time it takes to move through a section along the concourse is given by a function of the number of people in the section at that time. If there is less than 100 people in the section, it takes a flat 10 seconds, but if there is 100 or more people then it takes 9+1000^(400*10^-6*(N-100)) seconds.

The stadium company wishes for us to determine the time it will take for 50,000 people to be seated in the stadium. This is the major output from our simulation model, but we will also calculate the utilisation of the turnstiles, as this is a figure that the stadium could benefit from knowing.

Identification of Modelling and General Objectives

Modelling Objectives

We wish for our model to be an accurate simulation of a real-world stadium entry process. It should also be adaptible, so that it can easily adjust to a different stadium setup (e.g. improved signage around the stadium, or one of the sections being closed)
We also wish for our model to be an efficient simulation, which is both computationally efficient and time-efficient, to make it easier to solve more simulations for more accurate outputs.

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 gates to the stadium [4]
Number of turnstiles per gate into the stadium [20]

Defining Output Responses

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

Model Content

Identifying Entities

Person - Active
Turnstile - Active
Section - Passive
Gate - Passive

Drawing Behavioural Paths

Figure 1: Person Behavioural Path

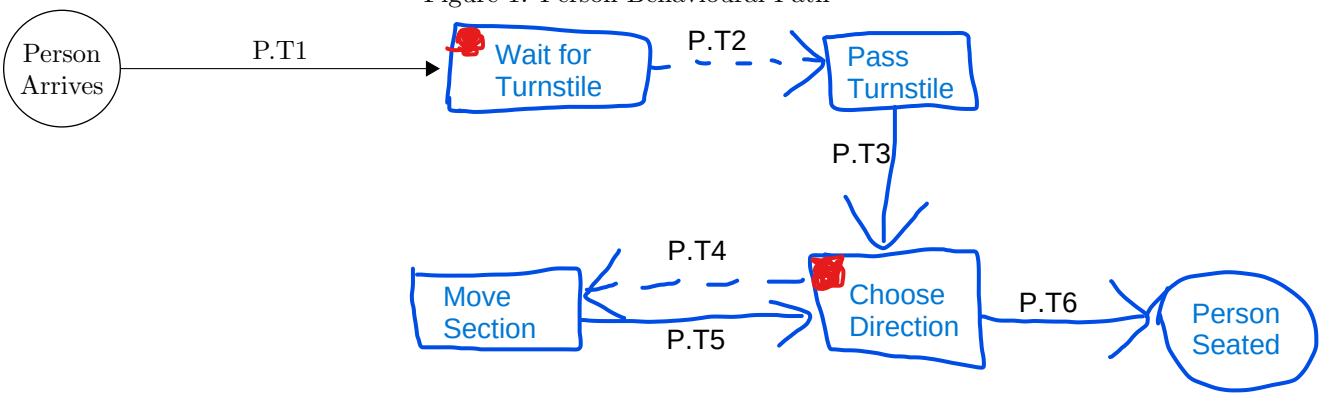
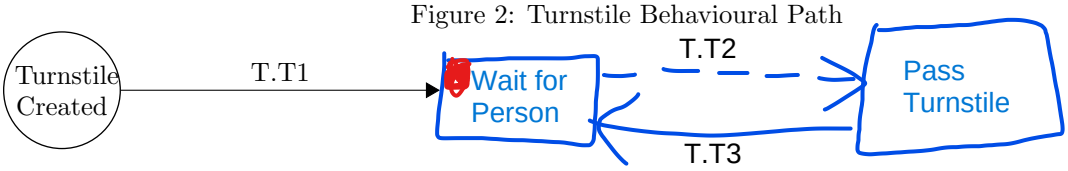


Figure 2: Turnstile Behavioural Path



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 P.Section > P.CurrentSection then
21:                         P.Direction = "up"
22:                 else//
23:                         P.Direction = "down"
24:                 end if
25:         end if
26:
27:
28:
29:
30:
31: end if
```

Model Data

Data	Source	Identification	Input	Output
InterarrivalTime	Distribution	Random Draw	-	Function defined below
TurnstileTime	Distribution	Random Draw	-	Function defined below
SectionTime	Distribution	Random Draw	People in Section N	Function of N (defined below)

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

InterarrivalTime ~ Exp(16) = 16e^(-16x)

TurnstileTime = triangular distribution(min 2s, mode 5s, max 12s)

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 CurrentSection 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] EntrySection [A, G, M or S]

(automatically includes CurrentActivity and CurrentActivityStartTime)

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: TRIGGER on start P.Wait For Turnstile
	End Event	**Type**	Controlled
		State Changes	1: 2: *TRANSITION* P.T2 // from logic
Wait for Person	**Participants**		Turnstile T
	Start Event	**Type**	Scheduled
		State Change	1: TRIGGER on start T.Wait For Person
	End Event	**Type**	Controlled
		State Changes	1: 2: T.WaitTime = T.WaitTime + CurrentTime = T.CurrentActivityStartTime 3: Transition T.T2 // from logic

Pass Turnstile	**Participants**		Person P, Turnstile T
	Start Event	**Type**	Controlled
		State Changes	1: P.CurrentSection = T.Gate.EntrySection SCHEDULE end event at CurrentTime+draw from TurnstileTime
	End Event	**Type**	Scheduled
		State Changes	1: TRANSITION P.T3 Pass Turnstile.END to Choose Direction.START TRANSITION T.T3 Pass Turnstile.END to Wait For Person.START
Move Section	**Participants**		Person P
	Start Event	**Type**	Controlled
		State Changes	1: P.CurrentSection.N -= 1 2: SCHEDULE end event at CurrentTime+draw from SectionTime
	End Event	**Type**	Scheduled
		State Changes	1: if P.Direction="up" then P.CurrentSection += 1 else P.CurrentSection -= 1 2: P.CurrentSection.N += 1 3: TRANSITION P.T5 Move Section.END to Choose Direction.START
Choose Direction	**Participants**		Person P
	Start Event	**Type**	Controlled
		State Change	1: SCHEDULE end event at CurrentTime
	End Event	**Type**	Scheduled
		State Changes	1: 2: Transition P.T4 or P.T6 // as determined by logic.

Model Events

Simulation Start	**Participant**	None
	Type	Scheduled
	State Changes	1: CREATE each Section S; Assign ID, N=0 2: CREATE each Gate G; Assign ID, EntrySection 3: CREATE each Turnstile T; Assign ID, Gate 4: CREATE each Person P; Assign ID, Gate, Section 5: START Person Arrives with P , Turnstile Created with T
Person Arrives	**Participant**	Person (P), Turnstile (T)
	Type	Scheduled
	State Changes	1: P.ID = max(Q.ID for Q in People) + 1 2: P.Section = choose random section 3: SCHEDULE Person Arrival at CurrentTime+draw from InterarrivalTime 4: TRANSITION P.T1 Person Arrives.END to Wait for Turnstile.START

Person Seated	**Participant**	Person P
	Type	Scheduled
	State Changes	1: -
Turnstile Created	**Participant**	Turnstile T
	Type	Scheduled
	State Changes	1: T.ID = max(U.ID for U in Turnstiles) + 1 Assign T.Gate 2: Transition T.T1 3: Turnstile Created.END to Wait For Person.START
Simulation Finish	**Participant**	None
	Type	Scheduled
	State Changes	1: **for** T ∈ Turnstiles **do** 2: Calculate utilisation for T 3: **end for** turnstile_ave_util = mean(utilisations for each T) time_everyone_seated = CurrentTime