# Conceptual Model for Jaamsim Implementation

## Model Data

| Data | Source | Identification | Input | Output |
|---|---|---|---|---|
| SectionDistribution | Discrete Uniform Distribution | Parameter | Min = 1, Max = 24 | Sample from Distribution |
| DirectionDistribution | Discrete Distribution | Parameter | Values = [-1, 1], Probabilities = [0.5, 0.5] | Sample from Distribution |
| TurnstileDistribution | Discrete Uniform Distribution | Parameter | Min = 1, Max = 80 | Sample from Distribution |
| InterarrivalTime | Poisson Process | Parameter | Mean = 0.0625s | Sample from Distribution |
| EntryTime | Triangular Distribution | Parameter | Min = 2s, Mode = 5s, Max = 12s | Sample from Distribution |
| TraverseTime | Function | Parameter | 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 []** | SeatSection [0] |
| | | Direction [0] |
| | | Turnstile [N/A] |
| | | Gate [0] |
| | | FirstChoice [0] |
| | | CurrentSection [0] |
| Turnstile | **Type** | Active |
| | **Attributes** | ID [0] |
| | | Gate [0] |
| Stadium | **Type** | Passive |
| | **Attributes** | NumberInSections [0,0,0, ..., 0] (List length 24) |
| | | NumberInStadium [0] |
| | | TurnstileList [] |

## Drawing Behavioural Paths

Figure 1: Person Behavioural Path

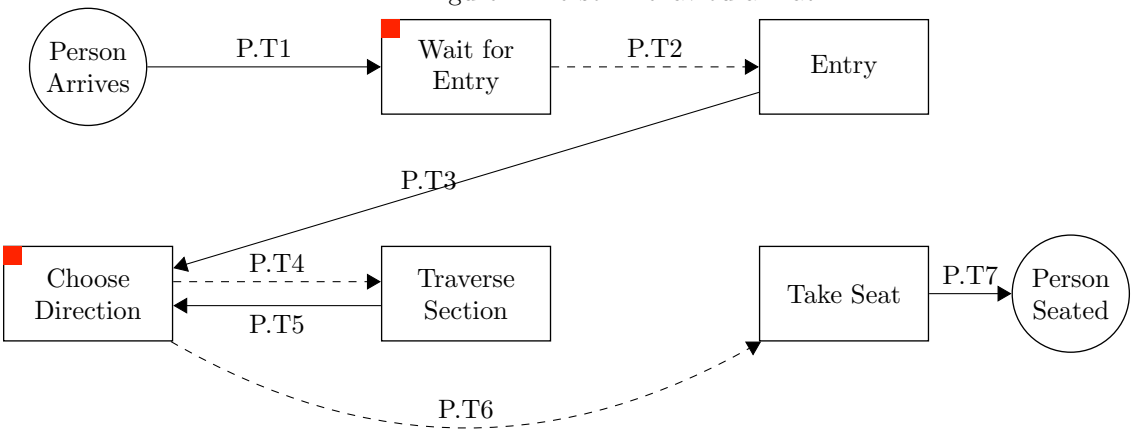
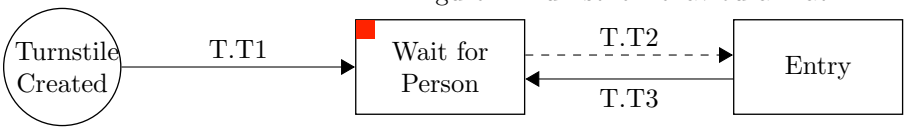



Figure 2: Turnstile Behavioural Path




## Model Transitions

| Transitions | From Event | To Event |
|---|---|---|
| P.T1 | Person Arrives | Wait for Entry.Start |
| P.T2 | Wait for Entry.End | Entry.Start |
| P.T3 | Entry.End | Choose Direction.Start |
| P.T4 | Choose Direction.End | Traverse Section.Start |
| P.T5 | Traverse Section.End | Choose Direction.Start |
| P.T6 | Choose Direction.End | Take Seat.Start |
| P.T7 | Take Seat.End | Person Seated |
| T.T1 | Turnstile Created | Wait for Person.Start |
| T.T2 | Wait for Person.End | Entry.Start |
| T.T3 | Entry.End | Wait for Person.Start |

## Model Events

| | | |
|---|---|---|
| Simulation Start | **Participant** | None |
| | **Type** | Scheduled |
| | **State Changes** | 1: CREATE Stadium S<br>2: S.NumberInStadium = 0<br>3: **for** 80 turnstiles **do**<br>4: CREATE Turnstile T<br>5: Turnstile Created with T<br>6: **end for**<br>7: CREATE Person P<br>8: S.NumberInStadium += 1<br>9: SCHEDULE Person Arrives with P at Current Time + InterarrivalTime |
| Person Arrives | **Participant** | Person (P) |
| | **Type** | Scheduled |
| | **State Changes** | 1: P.SeatSection = SectionDistribution<br>2: P.Direction = DirectionDistribution<br>3: P.Turnstile = Stadium.TurnstileList(TurnstileDistribution)<br>4: P.Gate = P.Turnstile.Gate<br>5: **if** S.NumberInStadium < 10,000 **then**<br>6: CREATE Person Q<br>7: S.NumberInStadium += 1<br>8: SCHEDULE Person Arrives with Q, T at Current Time + InterarrivalTime<br>9: **end if**<br>10: TRANSITION P.T1 from Person Arrives to Wait for Turnstile.Start with P |
| Person Seated | **Participant** | Person (P) |
| | **Type** | Scheduled |
| | **State Changes** | 1: Calculate statistics for P |
| Turnstile Created | **Participant** | Turnstile (T) |
| | **Type** | Scheduled |
| | **State Changes** | 1: T.ID = max(U.ID for U in Turnstiles) + 1<br>// *Get next ID*<br>2: T.Gate = floor((T.ID-1)/20) × 6 + 1<br>3: Stadium.TurnstileList.append(T)<br>4: TRANSITION T.T1 from Turnstile Created to Wait for Person.Start with T |
| Simulation Finish | **Participant** | None |
| | **Type** | Scheduled |
| | **State Changes** | |

## Model Activities

| | | | |
|---|---|---|---|
| Wait for Entry | **Participants** | | Person (P) |
| | **Start Event** | **Type** | Scheduled |
| | | **State Change** | 1: TRIGGER On Start Wait for Entry with P |
| | **End Event** | **Type** | Controlled |
| | | **State Changes** | 1: *// TRANSITION P.T2 is determined by logic* |
| Entry | **Participants** | | Person (P), Turnstile (T) |
| | **Start Event** | **Type** | Controlled |
| | | **State Change** | 1: SCHEDULE End Event at Current Time + EntryTime |
| | **End Event** | **Type** | Scheduled |
| | | **State Changes** | 1: P.CurrentSection = P.Gate<br>2: TRANSITION P.T3 from Entry.End to Choose Direction.Start with P<br>3: TRANSITION T.T3 from Entry.End to Wait for Person.Start with T |
| Choose Direction | **Participants** | | Person (P) |
| | **Start Event** | **Type** | Scheduled |
| | | **State Changes** | 1: TRIGGER On Start Choose Direction with P |
| | **End Event** | **Type** | Controlled |
| | | **State Changes** | 1: *// TRANSITION P.T4 or P.T6 determined by logic* |
| Traverse Section | **Participants** | | Person (P) |
| | **Start Event** | **Type** | Controlled |
| | | **State Changes** | 1: Stadium.NumberInSections(P.CurrentSection) += 1<br>2: SCHEDULE End Event at Current Time + TraverseTime(Stadium.NumberInSections(P.CurrentSection)) |
| | **End Event** | **Type** | Scheduled |
| | | **State Changes** | 1: Stadium.NumberInSections(P.CurrentSection) -= 1<br>2: P.CurrentSection = (P.CurrentSection+P.Direction+23)%24+1<br>3: TRANSITION P.T5 from Traverse Section.End to Choose Direction.Start with P |

| | | | |
|---|---|---|---|
| Take Seat | **Participants** | | Person (P) |
| | **Start Event** | **Type** | Controlled |
| | | **State Change** | 1: SCHEDULE Take Seat.End at Current Time + 30s |
| | **End Event** | **Type** | Scheduled |
| | | **State Changes** | 1: TRANSITION P.T7 from Take Seat.End to Person Seated with P |
| Wait for Person | **Participants** | | Turnstile (T) |
| | **Start Event** | **Type** | Scheduled |
| | | **State Change** | 1: TRIGGER On Start Wait for Person with T |
| | **End Event** | **Type** | Controlled |
| | | **State Changes** | 1: *// TRANSITION T.T2 is determined by logic* |

## Model Control – Defining Logic

**On Start Wait for Entry**

| **Triggered by** | Person P |
|---|---|

```
1: if P.Turnstile.CurrentActivity = Wait for Person then
2:         T = P.Turnstile
3:         Wait for Person.End with T
4:         Wait for Entry.End with P
5:         TRANSITION T.T2 Wait for Person.End to Entry.Start WITH T
6:         TRANSITION P.T2 Wait for Entry.End to Entry.Start WITH P
7:         Entry.Start with P, T
8: end if
```

**On Start Wait for Person**

| **Triggered by** | Turnstile T |
|---|---|

```
1: if any Person P with (P.Turnstile = T) and (P.CurrentActivity = Wait for Entry) then
2:         SELECT P with minimum P.ArrivalTime
3:         Wait for Person.End with T
4:         Wait for Entry.End with P
5:         TRANSITION T.T2 Wait for Person.End to Entry.Start WITH T
6:         TRANSITION P.T2 Wait for Entry.End to Entry.Start WITH P
7:         Entry.Start with P, T
8: end if
```

**On Start Choose Direction**

| **Triggered by** | Person P |
|---|---|

```
1: if P.CurrentSection = P.SeatSection then
2:         Choose Direction.End with P
3:         TRANSITION P.T6 Choose Direction.End to Take Seat with P
4: else
5:         if P.FirstChoice then
6:                 P.FirstChoice = 0
7:         else if (P.CurrentSection = 24) AND (P.Direction = -1) then
   // Crossed "A" to "X", find best direction from distance
8:                 if (P.SeatSection - 1) < (24 - P.SeatSection) then
9:                         P.Direction = 1
10:                else
11:                        P.Direction = -1
12:                end if
13:        else
14:                if P.SeatSection > P.CurrentSection then
15:                        P.Direction = 1'
16:                else// P.SeatSection < P.CurrentSection
17:                        P.Direction = -1
18:                end if
19:        end if
20:        TRANSITION P.T4 Choose Direction.End to Traverse Section with P
21: end if
```