Example HCCM Models

Thomas Adams

2025-02-18

Table of contents

Preface **3**

1 Output Buffering **4**
1.1 Data 4
1.2 Components 5
1.3 Activity Diagrams 7
1.4 Logic 8

2 Health Clinic **9**
2.1 Data 9
2.2 Components 10
2.3 Activity Diagrams 13
2.4 Logic 14

3 Radiology Clinic **15**
3.1 Data 15
3.2 Components 16
3.3 Activity Diagrams 18
3.4 Logic 19

4 Extended Radiology Clinic **20**
4.1 Data 20
4.2 Components 21
4.3 Activity Diagrams 24
4.4 Logic 25

Preface

These are example conceptual models that conform to the HCCM standard.

1 Output Buffering

1.1 Data

Table 1.1: List of Global Variables

Name	Description	Initial Value
NextPacketIdNum	The Id number that will be assigned to the next packet	1
P	The set of all patients	$\emptyset$

Table 1.2: List of Data Modules

Name	Source	Model	Type	Input	Output
InterarrivalTime	Problem Description	Poisson Process	Stochastic	Mean (1 min)	Sample from distribution
PacketSize	Problem Description	Triangular Distribution	Stochastic	Min, Mode, Max	Sample from Distribution
TransmitRate	Problem Description	Constant	Deterministic	-	Value
TransmitDuration	Problem Description	Function	Deterministic	Size, Rate	Value (Size x Rate)
BufferSize	Experiment	Constant	Deterministic	-	Value

1.2 Components

Table 1.3: List of Entities

Entity	Attributes
Packet	ID CurrentActivity CurrentStart Size
Link	ID CurrentActivity CurrentStart

Table 1.4: List of Transitions

Participant	Name	From Event	To Event
Packet	P.1	Packet Arrives	Wait for Link.Start
	P.2	Wait for Link.End	Transmit.Start
	P.3	Wait for Link.End	Packet Leaves
	P.4	Transmit.End	Packet Leaves
Link	L.1	Link Created	Wait for Packet.Start
	L.2	Wait for Packet.End	Transmit.Start
	L.3	Transmit.End	Wait for Packet.Start

Table 1.5: Activities

Activity	Participants	Event	Type	State Change

Wait for Link	Packet (p)	Start	Scheduled	`P.CurrentActivity = "Wait for Link" # default`
				`P.CurrentStart = TIME # default`
				`TRIGGER OnStartWaitForLink WITH p`

		End	Controlled	
Transmit	Packet (p), Link (l)	Start	Controlled	`SCHEDULE Transmit.End at TIME + TransmitDuration (p.Size, Rate)`
		End	Scheduled	`START Packet Leaves WITH p # TRANSITION P.4` `START Wait for Packet WITH l # TRANSITION L.3`
Wait for Packet	Link (l)	Start	Scheduled	`TRIGGER OnStartWaitForPacket WITH l`
		End	Controlled	

Table 1.6: Events

Event	Participants	Type	State Change
Simulation Start	-	Scheduled	`SCHEDULE Link Created at TIME` `SCHEDULE Packet Arrives at TIME + InterarrivalTime()`

Table 1.6: Events

Event	Participants	Type	State Change

Packet Arrives	-	Scheduled	`CREATE Packet p`
			`p.ID = NextPatIDNum`
			`NextPatIDNum = NextPatIDNum + 1`
			`p.Size = PacketSize()`
			`SCHEDULE Packet Arrives at TIME + InterarrivalTime()`
			`START Wait for Transmit WITH p # TRANSITION P.1`

Packet Leaves	Packet (p)	Scheduled	`Calculate statistics for p`

Link Created	-	Scheduled	`CREATE Link l`
			`l.Conversion = Conversion`
			`START Wait for Packet WITH l # TRANSITION L.1`

Simulation Finish	-	Scheduled	`Calculate any required statistics`

1.3 Activity Diagrams

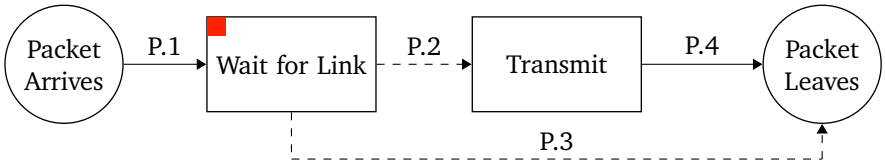

Figure 1.1: Packet Activity Diagram

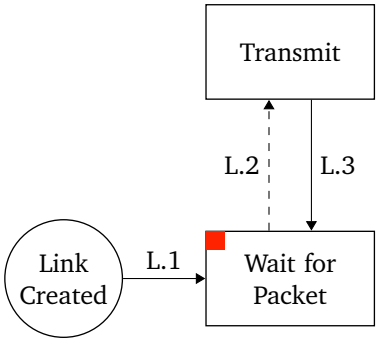

Figure 1.2: Link Activity Diagram

1.4 Logic

Table 1.7: OnStartWaitForLink

Triggered by: Packet p

```
waiting_packets = {p1 FOR p1 IN P IF p1.CurrentActivity = "Wait for Link"}
buffer_used = sum{p1.Size for p1 waiting_packets IF p1 != p}
IF L.CurrentActivity IS "Wait for Packet" THEN
    START Transmit WITH p, L # TRANSITIONS P.2, L.2
ELSE IF buffer_used + p.size > BufferSize THEN
    START Packet Leaves with p # TRANSITION P.3
END IF
```

Table 1.8: OnStartWaitForPacket

Triggered by: Link l

```
packets = {p FOR p IN P IF p.State = "Wait for Link"}
IF packets IS NOT empty THEN
    p_hat = argmin{p.CurrentStart FOR p IN packets}
    START Transmit WITH p_hat, l # TRANSITIONS P.2, L.2
END IF
```

2 Health Clinic

2.1 Data

Table 2.1: List of Global Variables

Name	Description	Initial Value
NextWalkUpIdNum	The Id number that will be assigned to the next walk-up patient	1
NextApptIdNum	The Id number that will be assigned to the next appointment patient	1
NextDoctorIdNum	The Id number that will be assigned to the next doctor	1
W	The set of all walk-up patients	$\emptyset$
A	The set of all appointment patients	$\emptyset$
D	The set of all doctors	$\emptyset$

Table 2.2: List of Data Modules

Name	Source	Model	Type	Input	Output
InterarrivalTime	Experiment	Poisson Process	Stochastic	Mean	Sample from distribution
AppointmentTimes	Experiment	Lookup	Deterministic	Appointment Number	Appointment Time
ConsultationDuration	Clinic Data	Triangular Distribution	Stochastic	Min, Mode, Max	Sample from Distribution

2.2 Components

Table 2.3: List of Entities

Entity	Attributes
Walk-up Patient	ID CurrentActivity CurrentStart ArrivalTime WaitTime
Appointment Patient	ID CurrentActivity CurrentStart AppointmentNumber AppointmentTime WaitTime
Doctor	ID CurrentActivity CurrentStart Role ConsultTime

Table 2.4: List of Transitions

Participant	Name	From Event	To Event
Walk-up Patient	W.1	Walk-up Arrives	Walk-up Wait for Consultation.Start
	W.2	Walk-up Wait for Consultation.End	Consultation.Start
	W.3	Consultation.End	Walk-up Patient Leaves
Appointment Patient	A.1	Appt. Arrives	Appt. Wait for Consultation.Start
	A.2	Appt. Wait for Consultation.End	Consultation.Start
	A.3	Consultation.End	Appt. Patient Leaves
Doctor	D.1	Doctor Created	Wait for Patient.Start
	D.2	Wait for Patient.End	Consultation.Start
	D.3	Consultation.End	Wait for Patient.Start

Table 2.5: Activities

Activity	Participants	Event	Type	State Change
Walk-up Wait for Consultation	Walk-up Patient (w)	Start	Scheduled	`TRIGGER OnStartWalkupWaitForConsultation WITH w`
		End	Controlled	`w.WaitTime = TIME - w.CurrentStart`
Consultation	Walk-up/Appointment Patient (p), Doctor (d)	Start	Controlled	`SCHEDULE Consultation.End at TIME + ConsultationDuration()`

		End	Scheduled	`d.ConsultTime += TIME - d.CurrentStart`
				`START Walk-up/Appt. Patient Leaves WITH p # TRANSITION P.3`
				`START Wait for Patient WITH d # TRANSITION D.3`

Appt. Wait for Consultation	Appt. Patient (a)	Start	Scheduled	`TRIGGER OnStartApptWaitForConsultation WITH a`
		End	Controlled	`a.WaitTime = TIME - a.CurrentStart`
Wait for Patient	Doctor (d)	Start	Scheduled	`TRIGGER OnStartWaitForPatient WITH d`
		End	Controlled	

Table 2.6: Events

Event	Participants	Type	State Change

Simulation Start	-	Scheduled	`SCHEDULE Create Doctor at TIME`
			`SCHEDULE Walk-up Patient Arrives at TIME + InterarrivalTime()`
			`SCHEDULE Appt. Patient Arrives at AppointmentTimes(1)`

Walk-up Patient Arrives	-	Scheduled	`CREATE Walk-up Patient w`
			`w.ID = NextWalkUpIDNum`
			`NextWalkUpIDNum = NextWalkUpIDNum + 1`
			`w.ArrivalTime = TIME`
			`w.WaitTime = 0`
			`SCHEDULE Walk-up Patient Arrives at TIME + InterarrivalTime()`
			`START Walk-up Wait for Consultation WITH w # TRANSITION W.1`

Appt. Patient Arrives	-	Scheduled	`CREATE Appointment Patient a`
			`a.ID = NextApptIDNum`
			`NextWalkUpIDNum = NextWalkUpIDNum + 1`
			`a.ArrivalTime = a.ID`
			`a.AppointmentTime = TIME`
			`SCHEDULE Appt. Patient Arrives at AppointmentTimes(a.ID + 1)`
			`START Appt. Wait for Consultation WITH a # TRANSITION A.1`

Table 2.6: Events

Event	Participants	Type	State Change

Create Doctor	-	Scheduled	`CREATE Doctor d`
			`d.ID = NextDoctorIDNum`
			`NextDoctorIDNum = NextDoctorIDNum + 1`
			`d.ConsultTime = 0`
			`IF d.ID = 1 THEN`
			`  d.Role = "Walk-up"`
			`ELSE`
			`  d.Role = "Appointment"`
			`END IF`
			`IF d.ID < 2 THEN`
			`  SCHEDULE Create Doctor at TIME`
			`END IF`
			`START Wait for Patient WITH d # TRANSITION D.1`

Walk-up Patient Leaves	Walk-up Patient (w)	Scheduled	`Calculate any required statistics for w`
Appt. Patient Leaves	Appointment Patient (a)	Scheduled	`Calculate any required statistics for a`
Simulation Finish	-	Scheduled	`Calculate any required statistics`

2.3 Activity Diagrams

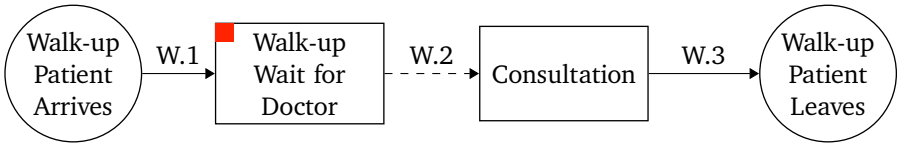

Figure 2.1: Walk-up Patient Activity Diagram

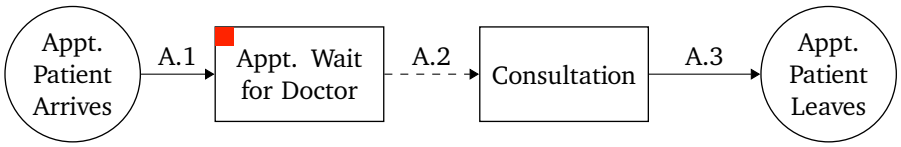

Figure 2.2: Appointment Patient Activity Diagram

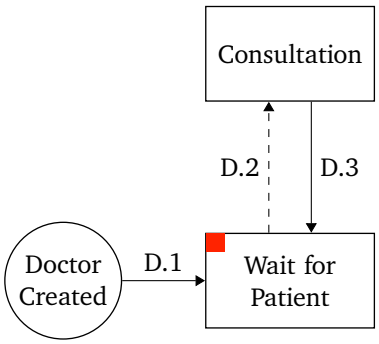

Figure 2.3: Doctor Activity Diagram

2.4 Logic

Table 2.7: OnStartWalkupWaitForConsultation

Triggered by: Walk-up Patient w

```
waiting_walkup_docs = {d FOR d IN D IF d.Role = "Walk-up" AND d.CurrentActivity
     = "Wait for Patient"}
waiting_appt_docs = {d FOR d IN D IF d.Role = "Appointment" AND d.
    CurrentActivity = "Wait for Patient"}
IF waiting_walkup_docs IS NOT empty THEN
    d_hat = argmin{d.CurrentStart FOR d IN waiting_walkup_docs}
    START Consultation WITH w, d_hat # TRANSITIONS W.2, D.2
ELSE IF waiting_appt_docs IS NOT empty THEN
    d_hat = argmin{d.CurrentStart FOR d IN waiting_appt_docs}
    START Consultation WITH w, d_hat # TRANSITIONS W.2, D.2
END IF
```

Table 2.8: OnStartApptWaitForConsultation

Triggered by: Appointment Patient a

```
waiting_appt_docs = {d FOR d IN D IF d.Role = "Appointment" AND d.
    CurrentActivity = "Wait for Patient"}
IF waiting_appt_docs IS NOT empty THEN
    d_hat = argmin{d.CurrentStart FOR d IN waiting_appt_docs}
    START Consultation WITH a, d_hat # TRANSITIONS A.2, D.2
END IF
```

Table 2.9: OnStartWaitForPatient

Triggered by: Doctor d

```
waiting_walkup_pats = {w FOR w IN W IF w.CurrentActivity = "Walk-up Wait for
    Doctor"}
waiting_appt_pats = {a FOR a IN A IF a.CurrentActivity = "Appt. Wait for Doctor
    "}
IF d.Role = "Appointment" and waiting_appt_pats IS NOT empty THEN
    p_hat = argmin{p.CurrentStart FOR p IN waiting_appt_pats}
    START Consultation WITH p_hat, d # TRANSITIONS A.2, D.2
ELSE IF waiting_walkup_pats IS NOT empty THEN
    p_hat = argmin{p.CurrentStart FOR p IN waiting_walkup_pats}
    START Consultation WITH p_hat, d # TRANSITIONS W.2, D.2
END IF
```

3 Radiology Clinic

3.1 Data

Table 3.1: List of Global Variables

Name	Description	Initial Value
NextPatIdNum	The Id number that will be assigned to the next patient	1
NextReceptionistIdNum	The Id number that will be assigned to the next receptionist	1
NextCTMachineIdNum	The Id number that will be assigned to the next CT Machine	1
P	The set of all patients	$\emptyset$
R	The set of all receptionists	$\emptyset$
C	The set of all CT Machines	$\emptyset$

Table 3.2: List of Data Modules

Name	Source	Model	Type	Input	Output
PatientInterarrivalTime	Problem Description	Poisson Process	Stochastic	Mean interarrival time	Sample from distribution
NumReceptionists	Problem Description	Constant	Deterministic	-	Value
NumCTMachines	Problem Description	Constant	Deterministic	-	Value
CheckInTime	Problem Description	Uniform Distribution	Stochastic	Min and max time	Sample from distribution
ScanTime	Problem Description	Log-normal Distribution	Stochastic	Mean and std. dev.	Sample from distribution

3.2 Components

Table 3.3: List of Entities

Entity	Attributes
Patient	ID CurrentActivity CurrentStart
Receptionist	ID CurrentActivity CurrentStart
CT Machine	ID CurrentActivity CurrentStart

Table 3.4: List of Transitions

Participant	Name	From Event	To Event
Patient	P.1	Arrive(P)	Wait for check in.Start
	P.2	Wait for check in.End	Check in.Start
	P.3	Check in.End	Wait for scan.Start
	P.4	Wait for scan.End	Scan.Start
	P.5	Scan.End	Leave(P)
Receptionist	R.1	Arrive(R)	Wait for task(R).Start
	R.2	Wait for task(R).End	Check in.Start
	R.3	Check in.End	Wait for task(R).Start
	R.4	Wait for task(R).End	Leave(R)
CT Machine	CT.1	Arrive(CT)	Wait for task(CT).Start
	CT.2	Wait for task(CT).End	Scan.Start
	CT.3	Scan.End	Wait for task(CT).Start
	CT.4	Wait for task(CT).End	Leave(CT)

Table 3.5: Activities

Activity	Participants	Event	Type	State Change
Wait for Check In	Patient (p)	Start	Scheduled	`TRIGGER OnStartWaitForCheckIn WITH p`
		End	Controlled	
Check In	Patient (p), Receptionist (r)	Start	Controlled	`SCHEDULE Check In.End at TIME + CheckInTime()`
		End	Scheduled	`START Wait for Scan WITH p # TRANSITION P.3` `START Wait for Task (R) WITH r # TRANSITION R.3`
Wait for Scan	Patient (p)	Start	Scheduled	
		End	Controlled	`TRIGGER OnStartWaitForScan WITH p`

Table 3.5: Activities

Activity	Participants	Event	Type	State Change
Scan	Patient (p), CTMachine (c)	Start	Controlled	`SCHEDULE Scan.End at TIME + ScanTime()`
		End	Scheduled	`START Leave (P) WITH p # TRANSITION P.5` `START Wait for Task (CT) WITH c # TRANSITION CT.3`
Wait for Task (R)	Receptionist (r)	Start	Scheduled	`TRIGGER OnStartWaitForTaskR WITH r`
		End	Controlled	
Wait for Task (CT)	CTMachine (c)	Start	Scheduled	`TRIGGER OnStartWaitForTaskCT WITH c`
		End	Controlled	

Table 3.6: Events

Event	Participants	Type	State Change

Simulation Start	-	Scheduled	`SCHEDULE Arrival (R) at TIME`
			`SCHEDULE Arrival (CT) at TIME`
			`SCHEDULE Arrival (P) at TIME + PatientInterArrival()`

Arrival (P)	Patient (p)	Scheduled	`p.ID = NextPatIDNum`
			`NextPatIDNum = NextPatIDNum + 1`
			`SCHEDULE Arrival (P) at TIME + PatientInterArrival()`
			`START Wait for Check In WITH p # TRANSITION P.1`

Leave (P)	Patient (p)	Scheduled	`Calculate statistics for p`

Arrival (R)	Receptionist (r)	Scheduled	`r.ID = NextReceptionistIDNum`
			`NextReceptionistIDNum = NextReceptionistIDNum + 1`
			`IF NextReceptionistIDNum <= NumReceptionists THEN`
			`  SCHEDULE Arrival (R) at TIME`
			`END IF`
			`START Wait for Task (R) WITH r # TRANSITION R.1`

Leave (R)	Receptionist (r)	Scheduled	`Calculate statistics for r`

Arrival (CT)	CT Machine (c)	Scheduled	`c.ID = NextCTMachineIDNum`
			`NextCTMachineIDNum = NextCTMachineIDNum + 1`
			`IF NextCTMachineIDNum <= NumCTMachines THEN`
			`  SCHEDULE Arrival (CT) at TIME`
			`END IF`
			`START Wait for Task (CT) WITH c # TRANSITION CT.1`

Leave (CT)	CT Machine (c)	Scheduled	`Calculate statistics for c`

3.3 Activity Diagrams

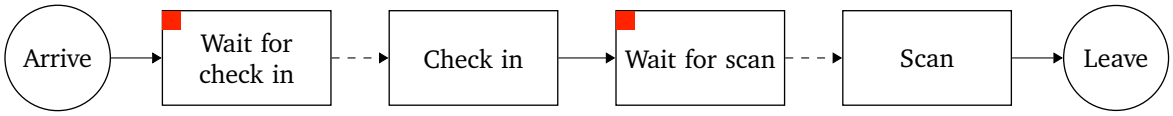

Figure 3.1: Patient Activity Diagram

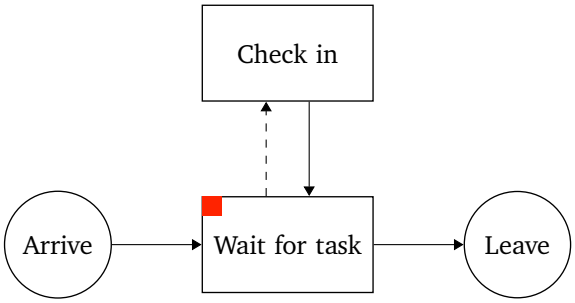

Figure 3.2: Receptionist Activity Diagram

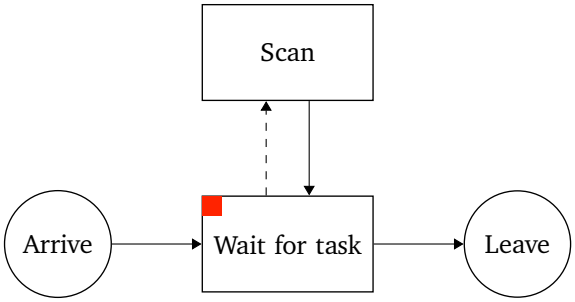

Figure 3.3: CT Activity Diagram

3.4 Logic

Table 3.7: OnStartWaitForCheckIn

Triggered by: Patient p

```
receps = {r FOR r IN R IF r.State = "Wait for task (R)"}
IF receps IS NOT empty THEN
    r_hat = argmin{r.CurrentStart FOR r IN receps}
    START Check In WITH p, r_hat # TRANSITIONS P.2, R.2
END IF
```

Table 3.8: OnStartWaitForScan

Triggered by: Patient p

```
cts = {c FOR c IN C IF c.State = "Wait for task (C)"}
IF cts IS NOT empty THEN
    c_hat = argmin{c.CurrentStart FOR c IN cts}
    START Scan WITH p, c_hat # TRANSITIONS P.4, CT.2
END IF
```

Table 3.9: OnStartWaitForTaskR

Triggered by: Receptionist r

```
patients = {p FOR p IN P IF p.State = "Wait for Check In"}
IF patients IS NOT empty THEN
    p_hat = argmin{p.CurrentStart FOR p IN patients}
    START Check In WITH p_hat, r # TRANSITIONS P.2, R.2
END IF
```

Table 3.10: OnStartWaitForTaskCT

Triggered by: CTMachine c

```
patients = {p FOR p IN P IF p.State = "Wait for Scan"}
IF patients IS NOT empty THEN
    p_hat = argmin{p.CurrentStart FOR p IN patients}
    START Scan WITH p_hat, c # TRANSITIONS P.4, CT.2
END IF
```

4 Extended Radiology Clinic

4.1 Data

Table 4.1: List of Global Variables

Name	Description	Initial Value
NextPatIdNum	The Id number that will be assigned to the next patient	1
NextReceptionistIdNum	The Id number that will be assigned to the next receptionist	1
NextCTMachineIdNum	The Id number that will be assigned to the next CT Machine	1
P	The set of all patients	$\emptyset$
R	The set of all receptionists	$\emptyset$
C	The set of all CT Machines	$\emptyset$

Table 4.2: List of Data Modules

Name	Source	Model	Type	Input	Output
PatientInterarrivalTime	Problem Description	Poisson Process	Stochastic	Mean interarrival time	Sample from distribution
PatientPriority	Problem Description	Discrete Distribution	Stochastic	Priority Probabilities	Sample from distribution
NumReceptionists	Problem Description	Constant	Deterministic	-	Value
NumCTMachines	Problem Description	Constant	Deterministic	-	Value
CheckInTime	Problem Description	Uniform Distribution	Stochastic	Min and max time	Sample from distribution
ScanTime	Problem Description	Log-normal Distribution	Stochastic	Mean and std. dev.	Sample from distribution

4.2 Components

Table 4.3: List of Entities

Entity	Attributes
Patient	ID CurrentActivity CurrentStart Priority[0]
Receptionist	ID CurrentActivity CurrentStart
CT Machine	ID CurrentActivity CurrentStart NeedMaintenance[0]

Table 4.4: List of Transitions

Participant	Name	From Event	To Event
Patient	P.1	Arrive(P)	Wait for check in.Start
	P.2	Wait for check in.End	Check in.Start
	P.3	Check in.End	Wait for scan.Start
	P.4	Wait for scan.End	Scan.Start
	P.5	Scan.End	Leave(P)
	P.6	Arrive(P)	Wait for scan.Start
Receptionist	R.1	Arrive(R)	Wait for task(R).Start
	R.2	Wait for task(R).End	Check in.Start
	R.3	Check in.End	Wait for task(R).Start
	R.4	Wait for task(R).End	Leave(R)
CT Machine	CT.1	Arrive(CT)	Wait for task(CT).Start
	CT.2	Wait for task(CT).End	Scan.Start
	CT.3	Scan.End	Wait for task(CT).Start
	CT.4	Wait for task(CT).End	Leave(CT)
	CT.5	Wait for task(CT).End	Maintenance.Start
	CT.6	Maintenance.End	Wait for task(CT).Start

Table 4.5: Activities

Activity	Participants	Event	Type	State Change
Wait for Check In	Patient (p)	Start	Scheduled	`TRIGGER OnStartWaitForCheckIn WITH p`
		End	Controlled	
Check In	Patient (p), Receptionist (r)	Start	Controlled	`SCHEDULE Check In.End at TIME + CheckInTime()`
		End	Scheduled	`START Wait for Scan WITH p # TRANSITION P.3` `START Wait for Task (R) WITH r # TRANSITION R.3`

Table 4.5: Activities

Activity	Participants	Event	Type	State Change
Wait for Scan	Patient (p)	Start	Scheduled	
		End	Controlled	`TRIGGER OnStartWaitForScan WITH p`
Scan	Patient (p), CTMachine (c)	Start	Controlled	`SCHEDULE Scan.End at TIME + ScanTime()`
		End	Scheduled	`START Leave (P) WITH p # TRANSITION P.5` `START Wait for Task (CT) WITH c # TRANSITION CT .3`
Wait for Task (R)	Receptionist (r)	Start	Scheduled	`TRIGGER OnStartWaitForTaskR WITH r`
		End	Controlled	
Wait for Task (CT)	CTMachine (c)	Start	Scheduled	`TRIGGER OnStartWaitForTaskCT WITH c`
		End	Controlled	
Maintenance	CTMachine (c)	Start	Controlled	`SCHEDULE Maintenance.End at TIME + 30 minutes`
		End	Scheduled	`c.NeedMaintenance = 0` `START Wait for Task (CT) WITH c # TRANSITION CT .6`

Table 4.6: Events

Event	Participants	Type	State Change

Simulation Start	-	Scheduled	`SCHEDULE Arrival (R) at TIME`
			`SCHEDULE Arrival (CT) at TIME`
			`SCHEDULE Arrival (P) at TIME + PatientInterArrival()`

Arrival (P)	Patient (p)	Scheduled	`p.ID = NextPatIDNum`
			`p.Priority = PatientPriority()`
			`NextPatIDNum = NextPatIDNum + 1`
			`SCHEDULE Arrival (P) at TIME + PatientInterArrival()`
			`IF p.Priority <= 2 THEN`
			`  START Wait for Scan WITH p # TRANSITION P.6`
			`ELSE`
			`  START Wait for Check In WITH p # TRANSITION P.1`
			`END IF`

Leave (P)	Patient (p)	Scheduled	`Calculate statistics for p`

Arrival (R)	Receptionist (r)	Scheduled	`r.ID = NextReceptionistIDNum`
			`NextReceptionistIDNum = NextReceptionistIDNum + 1`
			`IF NextReceptionistIDNum <= NumReceptionists THEN`
			`  SCHEDULE Arrival (R) at TIME`
			`END IF`
			`START Wait for Task (R) WITH r # TRANSITION R.1`

Leave (R)	Receptionist (r)	Scheduled	`Calculate statistics for r`

Table 4.6: Events

Event	Participants	Type	State Change

Arrival (CT)	CT Machine (c)	Scheduled	`c.ID = NextCTMachineIDNum`
			`NextCTMachineIDNum = NextCTMachineIDNum + 1`
			`IF NextCTMachineIDNum <= NumCTMachines THEN`
			`    SCHEDULE Arrival (CT) at TIME`
			`END IF`
			`START Wait for Task (CT) WITH c # TRANSITION CT.1`

Leave (CT)	CT Machine (c)	Scheduled	`Calculate statistics for c`
Require Maintenance	CT Machine (c)	Scheduled	`c.NeedMaintenance = 1` `TRIGGER OnRequireMaintenance WITH c`

4.3 Activity Diagrams

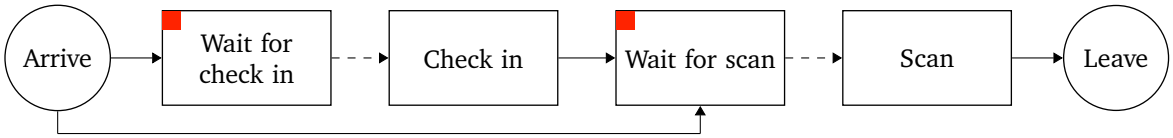

Figure 4.1: Patient Activity Diagram

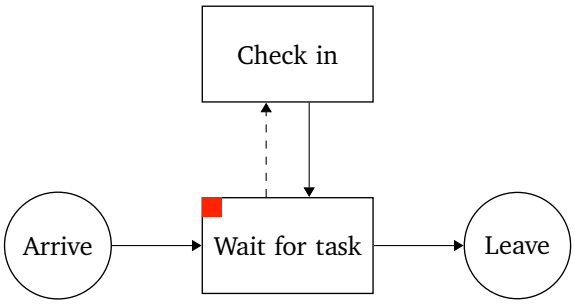

Figure 4.2: Receptionist Activity Diagram

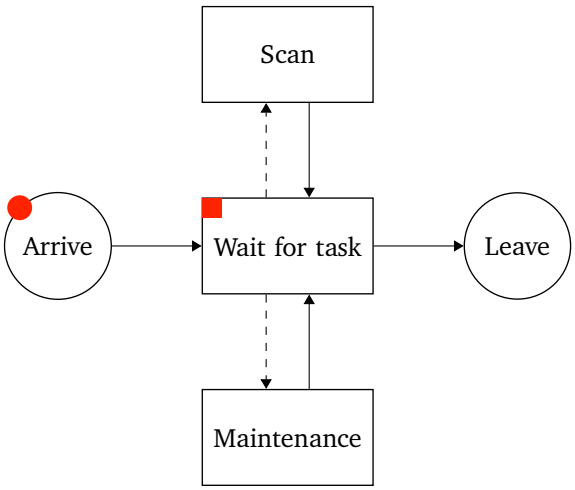

Figure 4.3: CT Activity Diagram

4.4 Logic

Table 4.7: OnStartWaitForCheckIn

Triggered by: Patient p

```
receps = {r FOR r IN R IF r.State = "Wait for task (R)"}
IF receps IS NOT empty THEN
    r_hat = argmin{r.CurrentStart FOR r IN receps}
    START Check In WITH p, r_hat # TRANSITIONS P.2, R.2
END IF
```

Table 4.8: OnStartWaitForScan

Triggered by: Patient p

```
cts = {c FOR c IN C IF c.State = "Wait for task (C)"}
IF cts IS NOT empty THEN
    c_hat = argmin{c.CurrentStart FOR c IN cts}
    START Scan WITH p, c_hat # TRANSITIONS P.4, CT.2
END IF
```

Table 4.9: OnStartWaitForTaskR

Triggered by: Receptionist r

```
patients = {p FOR p IN P IF p.State = "Wait for Check In"}
IF patients IS NOT empty THEN
    p_hat = argmin{p.CurrentStart FOR p IN patients}
    START Check In WITH p_hat, r # TRANSITIONS P.2, R.2
END IF
```

Table 4.10: OnStartWaitForTaskCT

Triggered by: CTMachine c

```
patients = {p FOR p IN P IF p.State = "Wait for Scan"}
IF c.NeedMaintenance = 1 THEN
  START Maintenance WITH c # TRANSITION CT.5
IF patients IS NOT empty THEN
    top_priority = max{p.Priority FOR p in patients}
    top_patients = {p FOR p IN patients IF p.Priority = top_priority}
    p_hat = argmin{p.CurrentStart FOR p IN top_patients}
    START Scan WITH p_hat, c # TRANSITIONS P.4, CT.2
END IF
```

Table 4.11: OnCTMachineArrive

Triggered by: CT Machine c
`SCHEDULE Require Maintenance WITH c at TIME + 8 hours`

Table 4.12: OnRequireMaintenance

Triggered by: CT Machine c

`SCHEDULE Require Maintenance WITH c at TIME + 8 hours`
`IF c.CurrentActivity = "Wait for task (C)" THEN`
`  START Maintenance WITH c # TRANSITION CT.5`
`END IF`