SIMULATION TRACES

Programació d’esdeveniments

Interacció de processos

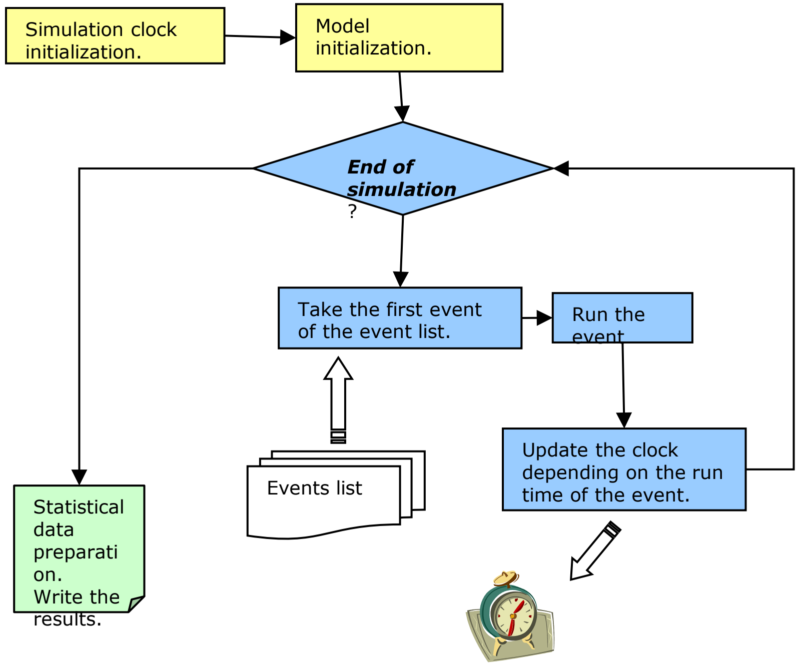
Exploració d’activitats

Event Scheduling trace

We suppose this data set (priority for the arrivals)

Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?
0	0	1	-	0	0	0	0

Eleme nt	Arrival time	Service time
1	1	1
2	2	2
3	3	1
4	4	2
5	5	1



Simulation clock initialization.
Model initialization.
End of simulation ?
Take the first event of the event list.
Run the event
Events list
Update the clock depending on the run time of the event.
Statistical data preparati on. Write the results.

Event Scheduling trace

Element	Arrival time	Service time
1	1	1
2	2	2
3	3	1
4	4	2
5	5	1

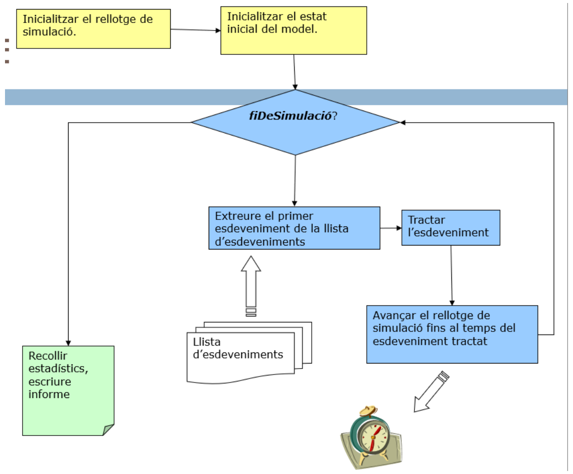
Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?	Num Exits	Num Arr
0	0	1	-	0	0	0	0	0	0
1	1	2	2	1	0	1	0	0	1
2	2	3	2	1	1	1	0	0	2
3	2	3	4	1	0	0	1	1	2
4	3	4	4	1	1	1	0	1	3
5	4	5	4	1	2	1	0	1	4
6	4	5	5	1	1	0	1	2	4
7	5	-	5	1	2	1	0	2	5
8	5	-	7	1	1	0	1	3	5
9	7	-	8	1	0	0	1	4	5
10	8	-	9	0	0	0	1	5	5

Event Scheduling trace

We suppose this data set (priority for the arrivals)

Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?
0	0	1	-	0	0	0	0

Element	Arrival time	Service time
1	1	1
2	2	2
3	2.5	2
4	3	1
5	6	2


Inicialitzar el rellotge de simulació.
Inicialitzar el estat inicial del model.
fiDeSimulació?
Extreure el primer esdeveniment de la llista d’esdeveniments
Tractar l’esdeveniment
Llista d’esdeveniments
Avançar el rellotge de simulació fins al temps del esdeveniment tractat
Recollir estadístics, escriure informe

Event Scheduling trace

Element	Arrival time	Service time
1	1	1
2	2	2
3	2.5	2
4	3	1
5	6	2

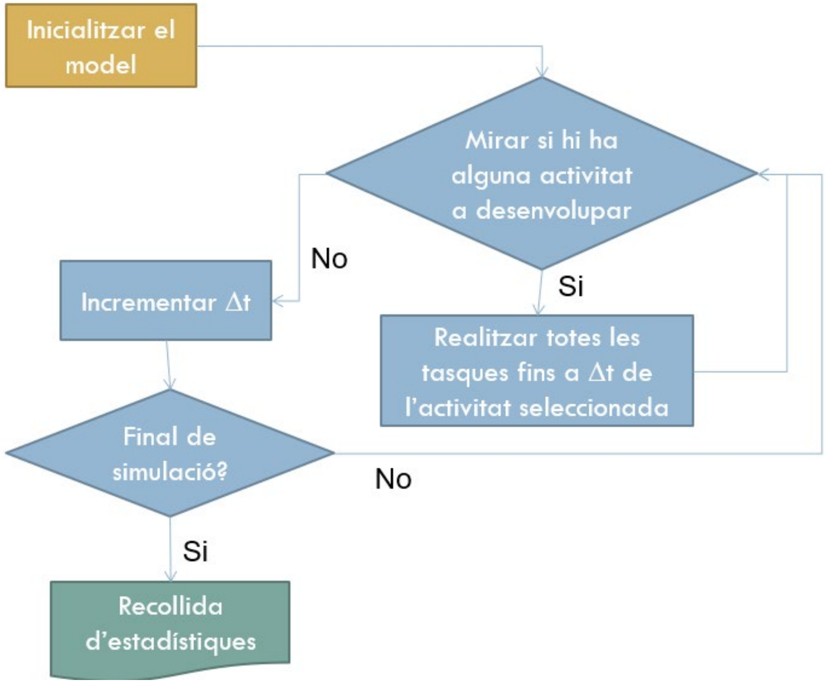
Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?	Num Exits	Num Arr
0	0	1	-	0	0	0	0	0	0
1	1	2	2	1	0	1	0	0	1
2	2	2,5	2	1	1	1	0	0	2
3	2	2,5	4	1	0	0	1	1	2
4	2,5	3	4	1	1	1	0	1	3
5	3	6	4	1	2	1	0	1	4
6	4	6	6	1	1	0	1	2	4
7	6	-	6	1	2	1	0	2	5
8	6	-	7	1	1	0	1	3	5
9	7	-	9	1	0	0	1	4	5
10	9	-	-	0	0	0	1	5	5

Activity scanning trace

- We suppose a delta = 1. We simulate until time = 6

Id	Time	Event time	Next Arrival	Next Exit	Server state	Queue long	Is Arrival ?	Is Exit?
0	0	0	1	-	0	0	0	0

Element	Arrival time	Service time
1	1	1
2	2	2
3	2.5	2
4	3	1
5	6	2



Inicialitzar el model

Mirar si hi ha alguna activitat a desenvolupar

No

Si

Incrementar Δt

Realitzar totes les tasques fins a Δt de l'activitat seleccionada

Final de simulació?

No

Si

Recollida d'estadístiques

Activity scanning trace

Element	Arrival time	Service time
1	*1*	1
2	*2*	2
3	*2.5*	2
4	*3*	1
5	*6*	2

Id	Time	Event time	Next Arrival	Next Exit	Server state	Queue long	Is Arrival?	Is Exit?
0	0	0	1	0	0	0	0	0
1	1	1	2	2	1	0	1	0
2	2	2	2.5	2	1	1	1	0
3	2	2	2.5	4	1	0	0	1
4	3	2.5	3	4	1	1	1	0
5	3	3	6	4	1	2	1	0
6	4	4	6	6	1	1	0	1
7	5							
8	6	6	-	6	1	2	1	0
9	6	6	-	7	1	1	0	1
10	7	7	-	9	1	0	0	1
11	8							
12	9	9	-	-	0	0	0	1

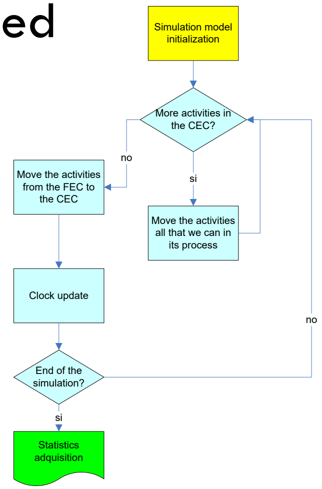
Process interaction trace

Steep	Time	BEC	CEC	FEC
1	Start		-	-
2	0		-	(1,Out,1,2)

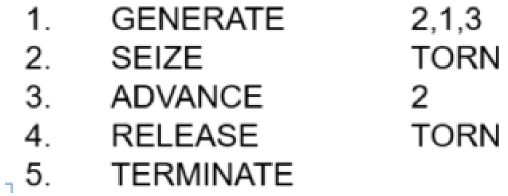
- Generation times:
  - (2,3,6)
- Only 3 entities are generated

- FEC contains:
  - ID of XACT
  - ID of current block.
  - ID of destination block.
  - Time to move

```
1.    GENERATE     2,1,3
2.    SEIZE        TORN
3.    ADVANCE      2
4.    RELEASE      TORN
5.    TERMINATE
```


Simulation model initialization
More activities in the CEC?
no
si
Move the activities from the FEC to the CEC
Move the activities all that we can in its process
Clock update
no
End of the simulation?
si
Statistics adquisition

Example (Event chains)



```
1.   GENERATE    2,1,3
2.   SEIZE       TORN
3.   ADVANCE     2
4.   RELEASE     TORN
5.   TERMINATE
```

Steep	Time	BEC	CEC	FEC	Comentaris
1	Start		-	-	
2	0		-	(1,Out,1,2)	The first XACT..
3	2		(1,Out,1,Now)	-	Xact from FEC to CEC.
4	2		-	(2,Out,1,3) (1,3,4,4)	Moving the Xact 1 as far as we can, stops on 3 (ADVANCE). Generation of the second XACT.
5	3		(2,Out,1,Now)	(1,3,4,4)	Xact from FEC to CEC.
6	3	(2,2,3,Now)		(1,3,4,4) (3,Out,1,6)	Moving the XACT 2 as far as we can, stops on 2 (SEIZE. Generating the third XACT.

Example (Event chains)

1.	GENERATE	2,1,3
2.	SEIZE	TORN
3.	ADVANCE	2
4.	RELEASE	TORN
5.	TERMINATE	

Steep	Time	BEC	CEC	FEC	Comentaris
7	4		(1,3,4,Now) (2,2,3,Now)	(3,Out,1,6)	Xact from FEC and BEC to CEC.
8	4		-	(3,Out,1,6) (2,3,4,6)	Move the Xact 1 as far as we can, leave's the System. We move the Xact 2 as far as we can, block 3 (ADVANCE).
9	6		(2,3,4,Now) (3,Out,1,Now)	-	Xact from FEC to CEC.
10	6		-	(3,3,4,8)	Move the Xact 2 as far as we can, leave the System. We move the Xact 3 as far as we can, block 3 (ADVANCE).
11	8		(3,3,4,Now)	-	Xact from FEC to CEC.
12	8		-	-	Move the Xact 3 as far as we can, leave the System.

Process interaction trace

Steep	Time	CEC	FEC
1	Start	-	-
2	0	-	(1,Out,1,2)

```
1.    GENERATE      6,4
2.    SEIZE         TORN
3.    ADVANCE       2
4.    RELEASE       TORN
5.    TERMINATE
```

- Generation times:
  - (2,4,6)
- Only 3 entities are generated

- FEC contains:
  - ID of XACT
  - ID of current block.
  - ID of destination block.
  - Time to move

Example (Event chains)

```
1.    GENERATE      6,4
2.    SEIZE         TORN
3.    ADVANCE       2
4.    RELEASE       TORN
5.    TERMINATE
```

Steep	Time	CEC	FEC	Comments
1	Inici	-	-	
2	0	-	(1, Out,1,2)	The first XACT..
3	2	(1,Out,1, Now)	-	Xact from FEC to CEC.
4	2	-	(2, Out,1,4) (1,3,4,4)	Moving the Xact 1 as far as we can, stops on 3 (ADVANCE). Generation of the second XACT.
5	4	(2, Out,1, Now) (1,3,4,Now)		Xact from FEC to CEC.
6	4		(2,3,4,6) (3, Out,1,6)	Moving the XACT 1 leaves the System. Moving the XACT 2 as far as we can, stops on 3 (ADVANCE). Generating the third XACT.

Example (Event chains)

1.	GENERATE	6,4
2.	SEIZE	TORN
3.	ADVANCE	2
4.	RELEASE	TORN
5.	TERMINATE	

Steep	Time	CEC	FEC	Comments
7	6	(2,3,4, Now) (3, Out,1,Now)		Xact from FEC to CEC.
8	6	-	(3,3,4,8)	Moving the XACT 2 as far as we can, leaves the System. Moving the XACT 3 as far as we can, Arrive to 3 (ADVANCE).
9	8	(3,3,4, Now)		XACT from FEC to CEC.
10	8	-	-	Moving XACT 3 as far as we can, leaves the System.

Process interaction trace

Steep	Time	CEC	FEC
1	Start	-	-
2	0	-	(1,Out,1,3)

- Generation times:
  - (3,6,9)
- Only 3 entities are generated

- FEC contains:
  - ID of XACT
  - ID of current block.
  - ID of destination block.
  - Time to move

```
1.    GENERATE     6,4
2.    SEIZE        TORN
3.    ADVANCE      2
4.    RELEASE      TORN
5.    TERMINATE
```

Example (Event chains)

```
1.    GENERATE      6,4
2.    SEIZE         TORN
3.    ADVANCE       2
4.    RELEASE       TORN
5.    TERMINATE
```

Steep	Time	CEC	FEC	Comments
1	Inici	-	-	
2	0	-	(1, Out,1,3)	The first XACT..
3	3	(1,Out,1, Now)	-	Xact from FEC to CEC.
4	3	-	(2, Out,1,6) (1,3,4,5)	Moving the Xact 1 as far as we can, stops on 3 (ADVANCE). Generation of the second XACT.
5	5	(1,3,4,Now)	(2, Out,1,6)	Xact from FEC to CEC. Moving the XACT 1 leaves the System.
6	6	(2, Out,1Now)	-	Xact from FEC to CEC.
7	6		(2,3,4,8) (3, Out,1,9)	Moving the XACT 2 as far as we can, stops on 3 (ADVANCE). Generating the third XACT.

Example (Event chains)

```
1.    GENERATE     6,4
2.    SEIZE        TORN
3.    ADVANCE      2
4.    RELEASE      TORN
5.    TERMINATE
```

Steep	Time	CEC	FEC	Comments
8	8	(2,3,4, Now)	(3, Out,1,9)	Xact from FEC to CEC. Moving the XACT 2 as far as we can, leaves the System.
9	9	(3, Out,1,Now)		XACT from FEC to CEC.
10	9		(3,3,4, 11)	Moving the XACT 3 as far as we can, Arrive to 3 (ADVANCE).
11	11	(3,3,4, Now) -	-	Moving XACT 3 as far as we can, leaves the System.

Two servers

- □ A queue system is composed by two identical servers and a single queue shared by the two servers. To simplify the modelling process, we consider that the clients arrive to the system in the times specified on the next table. Notice that the clients can arrive in groups. The service time for each one of the servers follows a constant distribution of **one unit of time**.
- □ We will execute first the departures. Use **Activity Scanning** approach.

Time	Number of clients
1	1
2	2
5	3
6	1
7	1

Two servers

Time	Number of clients
1	1
2	2
5	3
6	1
7	1

Id	Time	Event time	Next Arrival	Next Exit Server 1	Next Exit Server 2	Server 1	Server 2	Queue	Event
0	0	0	1	-	-	0	0	0	-
1	1	1	2	2	-	1	0	0	A

First the departures

Two servers

Time	Number of clients
1	1
2	2
5	3
6	1
7	1

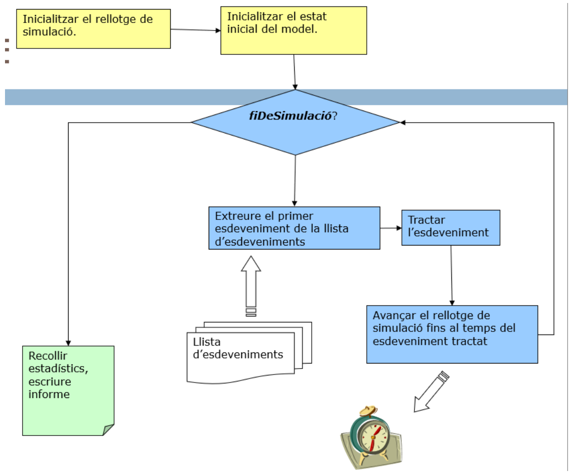
Id	Time	Event time	Next Arrival	Next Exit Server 1	Next Exit Server 2	Server 1	Server 2	Queue	Event
0	0	0	1	-	-	0	0	0	-
1	1	1	2	2	-	1	0	0	A
2	2	2	2	-	-	0	0	0	D1
3	2	2	5	3	3	1	1	0	A
4	3	3	5	-	3	0	1	0	D1
5	3	3	5	-	-	0	0	0	D2
6	4	-	5	-	-	0	0	0	-
7	5	5	6	6	6	1	1	1	A
8	6	6	6	7	6	1	1	0	D1
9	6	6	6	7	-	1	0	0	D2
10	6	6	7	7	7	1	1	0	A
11	7	7	7	-	7	1	1	0	D1
12	7	7	7	-	-	1	1	0	D2
13	7	7	-	8	-	1	0	0	A
14	8	8	-	-	-	0	0	0	D1

Event Scheduling trace

We suppose this data set (priority for the arrivals).
FILO QUEUE! (we do not want to know the lifespan of entities, but what to do in case we want?)

Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?
0	0	1	-	0	0	0	0

Element	Arrival time	Service time
1	1	1
2	1,5	2
3	2	5
4	3	2
5	5	1


Inicialitzar el rellotge de simulació.
Inicialitzar el estat inicial del model.
fiDeSimulació?
Extreure el primer esdeveniment de la llista d'esdeveniments
Tractar l'esdeveniment
Llista d'esdeveniments
Avançar el rellotge de simulació fins al temps del esdeveniment tractat
Recollir estadístics, escriure informe

#	Arrival time	Service time
1	1	1
2	1,5	2
3	2	5
4	3	2
5	5	1

Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?	Num Exits	Num Arr	Time client enters in queue	Av time waiting (queue)	Av time waiting (system)
0	0	1	-	0	0	0	0	0	0	{}	{}	{}
1	1	1,5	2	1	0	1	0	0	1	{}	{}	{}
2	1,5	2	2	1	1	1	0	0	2	{1.5}	{}	{}
3	2	3	2	1	2	1	0	0	3	{1.5, 2}	{}	{}
4	2	3	4	1	1	0	1	1	3	{1.5}	{0}	{1}
5	3	5	4	1	2	1	0	1	4	{1.5,,3}	{0}	{1}
6	4	5	9	1	1	0	1	2	4	{1.5}	{0,1}	{1, 2}
7	5	-	9	1	2	1	0	2	5	{1.5, 5}	{0,1}	{1, 2}
8	9	-	11	1	1	0	1	3	5	{1.5}	{0,1,4}	{1, 2, 6}
9	11	-	12	1	0	0	1	4	5	{}	{0, 1, 4, 9.5}	{1, 2, 6, 6}
10	12	-	13	0	0	0	1	5	5	{}	{0, 1, 4, , 9.5} = 3,6	{1, 2, 6, 6, 10,5 } = 5,1