



Софтвер за симулацију

Класификација софтвера за симулацију

- Програмски језици опште намене
 - флексибилни и познати
 - погодни за учење принципа и техника дискретне симулације
 - пример: C++, Java, (FORTRAN, Pascal)
- Програмски језици за симулацију
 - GPSS, SIMAN, ...
- Интегрисана симулациона окружења
 - Брза изградња модела
 - уграђене функције
 - графика и анимација
 - пример: Arena, Witness, Simfactory, Anylogic

Главни програм у језику Јава за симулацију система са једним сервером

```
class Sim {

    / Class Sim variables
    public static double Clock, MeanInterArrivalTime, MeanServiceTime,
        SIGMA, LastEventTime, TotalBusy, MaxQueueLength, SumResponseTime;
    public static long   NumberOfCustomers, QueueLength, NumberInService, TotalCustomers, NumberOfDepartures,
        LongService;

    public final static int arrival = 1;
    public final static int departure = 2;

    public static EventList FutureEventList; public static Queue Customers; public static Random stream;

    public static void main(String argv[]) {

        MeanInterArrivalTime = 4.5; MeanServiceTime = 3.2;
        SIGMA                 = 0.6; TotalCustomers   = 1000;
        long seed              = Long.parseLong(argv[0]);
        stream = new Random(seed); // initialize rng stream
        FutureEventList = new EventList();
        Customers = new Queue();

        Initialization();

        // Loop until first "TotalCustomers" have departed
        while(NumberOfDepartures < TotalCustomers) {
            Event evt = (Event)FutureEventList.getMin(); // get imminent event
            FutureEventList.dequeue(); // be rid of it
            Clock = evt.get_time(); // advance in time
            if( evt.get_type() == arrival ) ProcessArrival(evt); else ProcessDeparture(evt); }
        ReportGeneration();
    }
}
```

Методе за иницијализацију и догађај долазак

```
Public static void Initialization()      { Clock = 0.0;
QueueLength = 0;
NumberInService = 0;
LastEventTime = 0.0;
TotalBusy = 0 ;
MaxQueueLength = 0;
SumResponseTime = 0;
NumberOfDepartures = 0;
LongService = 0;

// create first arrival event Event evt =
new Event(arrival, exponential( stream, MeanInterArrivalTime)); FutureEventList.enqueue( evt );
}

Public static void ProcessArrival(Event evt)  { Customers.enqueue(evt);

QueueLength++;
// if the server is idle, fetch the event, do statistics // and put into service
if ( NumberInService == 0) ScheduleDeparture!);

else TotalBusy += (Clock - LastEventTime); // server is busy // adjust max queue length

statistics
if (MaxQueueLength < QueueLength) MaxQueueLength = QueueLength;

// schedule the next arrival
Event next_arrival =
new Event(arrival, Clock+exponential(stream,MeanInterArrivalTime)); FutureEventList.enqueue(
next_arrival );
LastEventTime = Clock;
}
```

GPSS (General Purpose Simulation System)

```
*****
*           Experimental Manufacturing Cell           *
*           Two CNC machines and one Robot          *
*           One arrival area and one finished parts area *
*****
          RMULT      78863
Transit TABLE M1,100,100,20           ;Record lead time
*****
          GENERATE (Exponential(1,0,150)) ;A job arrives
          QUEUE    One                    ;Arrival queue
          SEIZE    Robot                   ;Get the robot
          DEPART   One                     ;Depart the queue
          ADVANCE  8,1                     ;Robot grips the job
          ADVANCE  6                               ;Robot moves to machine 1
          ADVANCE  8,1                     ;Robot place the job
          RELEASE  Robot                    ;Free the robot
          QUEUE    Two                      ;Wait in next queue
          SEIZE    Machine1                 ;Get first machine
          DEPART   Two                      ;Depart the queue
          ADVANCE (Normal(1,60,10))        ;Process time
          RELEASE  Machine1                ;Free machine 1
          QUEUE    Three                    ;Join queue for machine 2
          SEIZE    Robot                    ;Get the robot
          DEPART   Three                    ;Depart the queue
          ADVANCE  8,1                     ;Robot grips part
          ADVANCE  7                               ;Robot moves to machine 2
          ADVANCE  8,1                     ;Robot places the part
          RELEASE  Robot                    ;Free the robot
          QUEUE    Four                     ;Join queue machine 2
          SEIZE    Machine2                 ;Get machine 2
          DEPART   Four                     ;Depart the queue
          ADVANCE (Exponential(1,0,100))   ;Process 2
          RELEASE  Machine2                ;Free machine 2
          QUEUE    Five                      ;Queue for exit station
          SEIZE    Robot                    ;Get the robot
          DEPART   Five                      ;Depart the queue
          ADVANCE  8,1                     ;Robot grips the part
          ADVANCE  5                               ;Robot moves to exit
          ADVANCE  8,1                     ;Robot places the part
          RELEASE  Robot                    ;Free the robot
          TABULATE Transit                 ;Transit time
          TERMINATE 1                       ;Job is completed
*****
```

GPSS извештаји

GPSS World Simulation Report - Robotfms.1.1

Sunday, March 01, 2009 16:21:57

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	162118.227	34	3	0

NAME	VALUE
FIVE	10014.000
FOUR	10012.000
MACHINE1	10010.000
MACHINE2	10013.000
ONE	10007.000
ROBOT	10008.000
THREE	10011.000
TRANSIT	10006.000
TWO	10009.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1004	0	0
	2	QUEUE	1004	0	0
	3	SEIZE	1004	0	0
	4	DEPART	1004	0	0
	5	ADVANCE	1004	0	0
	6	ADVANCE	1004	0	0
	7	ADVANCE	1004	0	0
	8	RELEASE	1004	0	0
	9	QUEUE	1004	0	0
	10	SEIZE	1004	0	0
	11	DEPART	1004	0	0
	12	ADVANCE	1004	0	0
	13	RELEASE	1004	0	0
	14	QUEUE	1004	0	0
	15	SEIZE	1004	1	0
	16	DEPART	1003	0	0
	17	ADVANCE	1003	0	0
	18	ADVANCE	1003	0	0
	19	ADVANCE	1003	0	0
	20	RELEASE	1003	0	0
	21	QUEUE	1003	2	0
	22	SEIZE	1001	0	0
	23	DEPART	1001	0	0
	24	ADVANCE	1001	1	0
	25	RELEASE	1000	0	0
	26	QUEUE	1000	0	0
	27	SEIZE	1000	0	0
	28	DEPART	1000	0	0
	29	ADVANCE	1000	0	0
	30	ADVANCE	1000	0	0
	31	ADVANCE	1000	0	0
	32	RELEASE	1000	0	0
	33	TABULATE	1000	0	0
	34	TERMINATE	1000	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
ROBOT	3008	0.408	22.005	1	1004	0	0	0	0
MACHINE1	1004	0.374	60.390	1	0	0	0	0	0
MACHINE2	1001	0.594	96.146	1	1001	0	0	0	2

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
ONE	3	0	1004	612	0.043	6.942	17.780
TWO	3	0	1004	624	0.095	15.398	40.684
THREE	2	1	1004	645	0.036	5.889	16.469
FOUR	10	2	1003	434	0.733	118.474	208.839
FIVE	3	0	1000	588	0.042	6.858	16.646

TABLE	MEAN	STD.DEV.	RANGE	RETRY FREQUENCY	CUM. %
TRANSIT	376.077	209.029		0	
			100.000 - 200.000	190	19.00
			200.000 - 300.000	276	46.60
			300.000 - 400.000	182	64.80
			400.000 - 500.000	131	77.90
			500.000 - 600.000	79	85.80
			600.000 - 700.000	56	91.40
			700.000 - 800.000	37	95.10
			800.000 - 900.000	22	97.30
			900.000 - 1000.000	11	98.40
			1000.000 - 1100.000	8	99.20
			1100.000 - 1200.000	7	99.90
			1200.000 - 1300.000	1	100.00

CEC XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
1004	0	162030.919	1004	15	16		

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
1005	0	162140.273	1005	0	1		
1001	0	162156.005	1001	24	25		

Интегрисано симулационо окружење (Anylogic)

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a simulation model of an ATM system. The model includes a source, a selectOutput block, a queue, an ATM service block, and a sink. Performance metrics are displayed: "Queue length: 8.182" and "ATM utilization: 0.41". Two histograms are shown: "Waiting time distribution" (yellow bars) and "Time in system distribution" (blue bars). A clock shows the time as 02:19:23. The left sidebar shows a project tree with folders like "bank", "Customer", and "Activity Based Costing Analysis". The bottom panel shows the "Main - Active Object Class" properties, including fields for Name, Agent, Generic, Startup code, and Destroy code. The right sidebar contains a palette with various simulation elements and a "Please note" section.

Please note
To add an element, drag-and-drop it from the palette onto the diagram
Some elements (marked by "pencil" icons) support drawing mode you've used in previous versions. To activate it, double-click on such palette element.
[Hide this information](#)

Model

- Parameter
- Flow Aux Variable
- Stock Variable
- Event
- Dynamic Event
- Plain Variable
- Collection Variable
- Function
- Table Function
- Port
- Connector
- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State
- Environment

Action

- Analysis
- Presentation
- Connectivity
- Enterprise Library
- [More Libraries...](#)

Историја развоја софтвера за симулацију

- 1955-60 Период истраживања (FORTRAN)
- 1961-65 Појава првих симулационих језика (засновани на FORTRAN-у SIMSCRIPT, GASP, ..., SIMULA, GPSS)
- 1966-70 Период формализације (GPSS/360)
- 1971-78 Период експанзије
- 1979-86 Период консолидације и регенерације (SIMAN)
- 1987- Интегрисана окружења

Избор софтвера за симулацију

- Функција моделирања
- Извршно (runtime) окружење
- Layout и анимације
- Функције извештавања
- Подршка испоручиоца софтвера и документација

Прављење модела

- Перспектива моделирања
- Могућност анализе улазних података
- Графичко моделирање
- Програмирање симулације
- Синтакса
- Улазна флексибилност
- Концизност моделирања
- Случајност
- Специјализоване компоненте и шаблони
- Могућност додавања објеката
- Веза са програмским језицима опште намене

Извршно окружење

- Брзина извршавања
- Величина модела; број променљивих и атрибута
- Интерактивни *debugger*
- Статус модела и статистика

Интегрисана окружења

- SIMAN/ARENA
- WITNESS
- SIMFACTORY
- AnyLogic
- FLEXY
-