International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

Multiagent Systems Simulation

G. Balakayeva, and A. Aktymbayeva

***Abstract*—**In this paper, we consider components of discrete event imitating model, implementing a simulation model by using JAVA and performing an input analysis of the data and an output analysis of the simulation results. Was lead development of imitating model of mass service system with n (n≥1) devices of service. On the basis of the developed process of a multithreading simulated the distributed processes with presence of synchronization. Was developed the algorithm of event-oriented simulation, was received results of system functioning with n devices of service.

***Keywords*—**Imitating modeling, Mass service system, Multi agent system.

I. Introduction

THE method of imitating modeling is one of the most powerful and most effective methods of research of processes and systems of the most various nature and a degree of complexity. The essence of this method consists in a development of the software for simulating process of functioning of system. In order to reception of statistical characteristics of modeled system is necessary carrying out of a plenty of experiments. Results of imitating modeling allow to describe behavior of system, to estimate influence of various parameters of system on its characteristics, to reveal advantages and lacks of offered changes, to predict behavior of system [1].

If attitudes which form model are simple enough for reception of the exact information on questions interesting us it is possible to use mathematical methods. Such decision refers to analytical. However most of the existing systems are very complex, and for them it is impossible to create the real model described analytically. Such models should be studied by means of modeling. At modeling the computer is used for numerical estimations of model and by means of received data we compute its real characteristics.

Imitating modeling can be applied in the most various fields of activity. Below you can see the list of problems at which modeling is especially effective [1]:

- designing and the analysis of industrial systems;
- an estimation of various systems of arms and requirements to its material-technical maintenance;
- definition of requirements to the equipment and to protocols of communication networks;
- definition of requirements to the equipment and to the software to various computer systems;
- designing and the analysis of work of transport systems, for example airports, highways, ports and underground;
- estimation of projects of creation of the various organizations of mass service, for example the centers of processing of orders, institutions of a fast food, hospitals, branches of communication;
- modernization of various processes in business sphere;
- definition of policy in control systems of stocks;
- analysis of financial and economic systems.

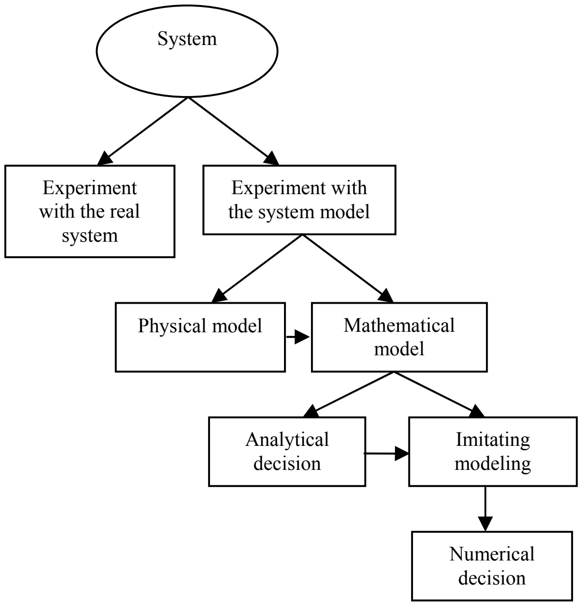

Fig. 1 Ways of the system research

II. System Model

Though modeling is applied to the diversified real systems, all discrete events imitating models include a number of the general components. The logic organization of these components allows to provide programming, debugging and the subsequent change of the program of imitating model [4]. Discrete event imitating model which uses the mechanism of promotion of time from event to event and is written in universal language contains following components:

- condition of the system - set of variables of the condition necessary for the description of system during the certain moment of time;
- clock of modeling time - a variable specifying current value of modeling time;
- the list of events - the list containing time of occurrence of every next type of events;
- statistical counters - the variables intended for storage of the statistical information on the characteristic of system;
- program of initialization - the subprogram establishing imitating model to the initial condition during the moment of time equal 0;
- synchronizing program - the subprogram which finds the next event in the list of events and then translates time of modeling to the period of occurrence of this event;
- the program of processing of events - subprogram updating a condition of the system when there is an event of the certain type (for each type of

events exists the separate program of processing of events);

- library programs - a set of the subprograms applied for generation of casual supervision from distributions of probabilities which have been certain as a part of imitating model;
- the generator of reports - subprogram which reads out estimations (from statistical counters) criteria of an assessment of works and gives out the report on end of modeling;
- the main program - subprogram, which causes synchronizing program to define next event, and then give management to the corresponding event program with the purpose of maintenance of the set updating a condition of system. The main program can supervise also necessity of the termination of modeling and call the generator of reports after its termination.

III. Task Formation

We consider a system where *n* machine services, in order of arrival, a stream of statistically identical jobs, each server with Markovian service times. The distributions of interarrival intervals, service times, operative periods and repair times are given. The only performance measures of interest is the average response time – the time between the arrival of a job and the completion of its service. There are two event types: job arrival and job departure. Intervals between arrivals are exponentially distributed with mean $\frac{1}{\lambda}$.

The basic queueing model is the so-called M/M/1 queueing model. In this model customers arrive according to a Poisson process and the service times of the customers are independent and identically exponentially distributed. The arrival rate is denoted by λ. The service rate is denoted by μ, so the mean service time is 1/μ. The customers are served by a single server in order of arrival (FCFS). We require that

$$\rho = \frac{\lambda}{\mu} < 1, \tag{1}$$

since, otherwise, the queue length will explode. The quantity ρ is the fraction of time the server is servicing customers. Although in reality a queueing system is never as mathematically simple as this, the model contains most of its essential characteristics. The analysis of this system will clearly show the sometimes devastating consequences of randomness both in the arrival process and in the service times.

IV. Analysis of Model and Results

To analyze our system we need information relating to:

- arrival process:
  - how customers arrive e.g. singly or in groups (batch or bulk arrivals)
  - how the arrivals are distributed in time (e.g. what is the probability distribution of time between successive arrivals (the interarrival time distribution))
  - whether there is a finite population of customers or (effectively) an infinite number.

  The simplest arrival process is one where we have completely regular arrivals (i.e. the same constant time interval between successive arrivals). A Poisson stream of arrivals corresponds to arrivals at random. In a Poisson stream successive customers arrive after intervals which independently are exponentially distributed. The Poisson stream is important as it is a convenient mathematical model of many real life queuing systems and is described by a single parameter - the average arrival rate. Other important arrival processes are scheduled arrivals; batch arrivals; and time dependent arrival rates (i.e. the arrival rate varies according to the time of day).
- service mechanism:
  - a description of the resources needed for service to begin
  - how long the service will take (the service time distribution)
  - the number of servers available
  - whether the servers are in series (each server has a separate queue) or in parallel (one queue for all servers)
  - whether preemption is allowed (a server can stop processing a customer to deal with another "emergency" customer)

    Assuming that the service times for customers are independent and do not depend upon the arrival process is common. Another common assumption about service times is that they are exponentially distributed.
- queue characteristics:
  - how, from the set of customers waiting for service, do we choose the one to be served next (e.g. FIFO (first-in first-out) - also known as FCFS (first-come first served); LIFO (last-in first-out); randomly) (this is often called the *queue discipline*)
  - do we have:
    - balking (customers deciding not to join the queue if it is too long)
    - reneging (customers leave the queue if they have waited too long for service)
    - jockeying (customers switch between queues if they think they will get served faster by so doing)
    - a queue of finite capacity or (effectively) of infinite capacity

In terms of the analysis of queuing situations the types of questions in which we are interested are typically concerned with measures of system performance and might include:

- How long does a customer expect to wait in the queue before they are served, and how long will they have to wait before the service is complete?
- What is the average length of the queue?
- What is the probability that the queue will exceed a certain length?
- What is the expected utilisation of the server and the expected time period during which he will be fully occupied.

In order to get answers to the above questions there are *two* basic approaches:

- analytic methods or queuing theory (formula based); and
- simulation (computer based).

The study of multiagent systems (MAS) focuses on systems in which many intelligent agents interact with each other. The agents are considered to be autonomous entities, such as software programs or robots. Their interactions can be either cooperative or selfish. That is, the agents can share a common goal, or they can pursue their own interests.

Topics of research in multi agent systems include [3]:

1. cooperation and coordination,
2. organisation,
3. communication,
4. negotiation,
5. distributed problem solving,
6. multi-agent learning,
7. scientific communities,
8. dependability and fault-tolerance.

V. Future Work and Conclusion

In given article we consider the next points:

- stochastic modeling of production and communication systems,
- exact and approximative analysis of queueing models for mass service system with n ($n \geq 1$) devices of service , and
- implementing a simulation model by using JAVA and performing an input analysis of the data and an output analysis of the simulation results.

Our future goal is to create a distributed simulation system to test various coordination mechanisms. Our assumption is that the system is composed of a group of autonomous agents. Each agent has its own local view of the world and its own goals, but is capable of coordinating these goals with respect to remote agents. In group communication, processes usually communicate in a group basis where a message is sent to a group of processes, rather than just to one process, which is the case in point-to-point communication.

References

[1] Averill M.Law, W. David Kelton. Simulation modeling and analysis. Third edition. McGraw-Hill, Osborne, 2003.
[2] Mitrani I. Simulation techniques for discrete event systems. Cambridge University press, 1982.
[3] Sycara K.P. Multiagent systems. http://www.cs.cmu.edu/~softagents/papers/multiagentsystems.PDF
[4] http://www.dmem.strath.ac.uk/~pball/simulation/simulate.html#Using%20simulation
[5] D. Gross, C.M. Harris, Fundamentals of queueing theory, Wiley, Chichester, 1985.
[6] http://www.aaai.org/AITopics/html/multi.html