Agents Interaction and Queueing System Model of Real Time Control of Students Service Center Load Balancing

Malika Abdrakhmanova(✉), Galimkair Mutanov, Zhanl Mamykova, and Ualsher Tukeyev

Al-Farabi Kazakh National University, Al-Farabi Avenue 71, Almaty, Kazakhstan
malika.berikovna@mail.ru, zhmamykova@gmail.com, ualsher.tukeyev@gmail.com

Abstract. The problem of effective organization of Students service center (SSC) activities is considered. In this paper is proposed combine agents interaction and queuing system model for creation real time control of SSC load balancing. The developed combined model allows to minimize the number of required personnel resources and their idle time and to create adaptive, modular, well scalable system.

Keywords: Electronic university · Students Service Center · Queueing system Load balancing · Multi-agent modeling

1 Introduction

Informatization of society involves radical social changes therefore high-quality expansion of the informatization demands comprehensive accounting of a human factor, relevant there is a problem of adaptation of the person to life in the conditions of the new information environment. Transformation of university activities with creating favorable conditions for consumers of its services through elimination of administrative barriers and preservation of valuable free time is possible when following to experience of creation of Public Service Centers and provide services by the principle of "single window" in the uniform Students Service Center (SSC).

The practical importance of SSC is directed to creation of socially important conditions for high-quality stay of students in the territory of a campus, having provided them conditions for receiving consultation on the organization of educational process, receiving high-quality socially significant services in one place.

Main activities of SSC: zones of providing administrative and consulting services in educational process; access points for "narrow experts" consultations: financier, lawyer; uniform center of the youth organizations; center of providing medical services; access point to electronic services of the university; zone of providing administrative services of the state character.

For effective management of SSC activities it is necessary to predict future load in various zones of service. Significant for consideration for the purpose of load

N. T. Nguyen et al. (Eds.): ICCCI 2018, LNAI 11055, pp. 349–359, 2018.
https://doi.org/10.1007/978-3-319-98443-8_32

optimization is zone of administrative and consulting services in educational process. The list of the functions realized in this zone by employees of registrar office and student's department is very extensive, also level of demand on these services raised. Due to the reorganization of their activities the new view on their services is created. In these departments in addition to consulting support of students, functions on administration of educational process are provided. This fact allows to divide these zones into sectors with the reduced number of functions. Functions of these sectors can be crossed on some questions, for this purpose registrar office and student's department employees are given with the corresponding functions of role policy of access, that allowing to serve as necessary any application with the crossed requests in these categories of service.

Now we are faced by a problem of modeling of registrar office and student's department zones activities. As SSC is the socially oriented system having signs of queueing systems, parameters of efficiency of its functions can be determined in terms of the queueing theory. Considering the fact that the documents generated in one departments are widely used in work of others, and functions are crossed thus between the specified departments and between sectors in departments, it is possible to consider in the system the possibility of transfer of requests to serve in other sectors at emergence of load balancing needs.

The field of social service including service of students, assumes constant need for service expansion, addition of new types of service. In view of need of fast and effective scaling of the system, the best approach to modeling of interaction between sectors for the purpose of load balancing will be multiagent approach.

2 Related Works

The research and development of queueing systems models have gained distribution in connection with tasks from the most various spheres of services. The detailed analysis of various models of queueing systems, their mathematical characteristics and opportunities of application are presented by Adan and Resing [1].

The social queueing systems represent the greatest difficulties for simulation owing to difficulty of prediction not only characteristics of an arrival steam of requests for service, but also prediction characteristics of service process. The reason for that is the impossibility of creation of process exact protocols in which it is difficult to provide requests in strictly predetermined template. The research of behavior of social queueing systems has found reflection in works of Yuan and Hwarng [2], where they have conducted a research of dynamic behavior of queueing system under impact of social interactions.

The complexity of prediction of social queueing system load requires to analyse of possibility of load balancing in the presence of several parallel operating queues for the purpose of maximizing use of resources and minimization of wait time. Jiang and Li in their work [3] consider 2 approaches to distribution of tasks, the value of this work is the analysis of talent-based allocation approach, following which tasks are distributed according to skills of agents. The algorithm of load balancing in parallel systems of customer service is offered by Down and Lewis [4].

In social simulation and modeling have been increasingly applied recent years agent-based approach. This approach allows to model the systems which are difficult to formalization by standard mathematical tools, it allows to build adaptive, modular, well scalable systems [5]. Researches in the field of agent-based approach represent an assessment of various theories of agent-based modeling of social systems, main applied fields of agent-based approach and offer various rules of agents interaction [5–7]. The important direction of agents interaction is cooperation for achievement of a common goal, various models of cooperation of agents were considered in works [8, 9].

Within the article considered the model of load balancing control of students service center in which the queueing theory and agent-based approach are combined. The control system is presented as multiagent system, where agents interact for providing effective service by redistribution of load, each agent achieves the local objectives proceeding from rules of the queueing theory.

3 Description of the Offered Approach

3.1 Modeling of the Students Service Center as a Queueing System

Students service center is an example of queueing system with several queues (zones) with non-uniform arrivals of model (M/M/c): (GD/∞/∞), on Kendall's classification [1]. This is the model with exponential interarrival times with mean $1/\lambda$, exponential service times with mean $1/\mu$ and c parallel identical servers. Customers are served in order of arrival. We suppose that the occupation rate per server,

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

is smaller than one.

An important quantity is the probability that a job has to wait. Denote this probability by ΠW. It is usually referred to as the delay probability [1].

$$\Pi_w = \frac{(c\rho)^c}{c!}\left((1-\rho)\sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!}\right)^{-1} \tag{2}$$

From the equilibrium probabilities we directly obtain for the mean queue length

$$L = \Pi_W \cdot \frac{\rho}{1-\rho} \tag{3}$$

The mean waiting time,

$$W = \Pi_W \cdot \frac{1}{1-\rho} \cdot \frac{1}{c\mu} \tag{4}$$

In general, arrival stream of queueing system can be non-uniform when the requests of several categories differing from each other in laws of distribution either intervals of

receipt, or service time, or serving priorities come to system. Very often in the analysis of such queues initial non-uniform load comes down to equivalent uniform. This process includes the following transformations of initial parameters:

1. intensity of the integrated stream

$$\lambda_{gen} = \sum_{k=1}^{H} \lambda_k, \tag{5}$$

where H is the number of requests categories

2. average holding time of requests of the integrated stream

$$t_{av} = \frac{1}{\lambda_{gen}} \sum_{k=1}^{H} \lambda_k \cdot t_k \tag{6}$$

3.2 Description of the Method of SSC Input Streams Balancing Dynamic Control with Use of Multiagent Approach

The primary activity of SSC is represented by functions on the organization of educational process. Considering the fact that within the academic year the intensity of addresses on many functions considerably changes, it is important to organize work of SSC with minimization of employees' idle time and customers waiting time in queue. In traditional approach to the organization of educational process, the main functions are distributed between registrar office and student's department. Functions of these departments can be subdivided into 4 categories:

1. functions of registrar office on consultation of students about the organization of educational process – f_1;
2. functions of registrar office on administration of educational process – f_2;
3. functions of student's department on consultation of students about the organization of educational process – f_3;
4. functions of student's department on administration of educational process – f_4.

Social systems are characterized by unpredictable behavior, therefore it is impossible to calculate ideal parameters of effective work of sectors as a queueing systems with a big share of reliability and to base long-term work of system on these parameters. For balancing of changing load in sectors, the possibility of the decentralized redistribution of an arrival requests flow with use of multiagent approach is considered. Use of multiagent approach will allow to develop module system, which can be scaled easily: to add new sectors with new sets of functions and new nature of interaction with other sectors.

Agents in system will be parallel program modules which accept the sector requests flow as an entrance, count parameters of work of the sector according to the queueing theory and interact with other agents for load balancing. Agents aim not only to support effective work of the sector, but also collaboration with other agents for maintenance of reliable work of all system. In terms of multiagent approach there is a set of different

definitions of the term the agent, we represent the definition, the most suitable for our purposes: an agent is an autonomous entity, which acts upon an environment and directs its activities toward achieving some specified intentions [10].

The system consists of 4 interacting agents $\{A_i\}_{i=1}^4$, which are carrying out calculation of service parameters for 4 sectors. Agents interact with the environment, accepting the requests which arrived on service and using environment resources to serve the requests. Resources of the environment are described quantitatively and qualitatively (quantity – $\{c_i\}_{i=1}^4$ and skills – $\{r_i\}_{i=1}^4$ of employees).

Skills (roles) of employees are defined by types of requests which they can process. Generally, all employees are universal and can process any type of requests, but for reduction of the general time of requests implementation due to mechanization of often performed operations, all types of requests are classified on 4 categories and are transferred only to the relevant sector to which certain employees are assigned.

Each agent is characterized by the skills and a state. Skills $\{R_i\}_{i=1}^4$ of agents represent calculation of working parameters for each sector according to the queueing theory. For the considered sectors the queueing system types describing them are identical and their parameters pay off by the same rules, but generally, sectors can belong to various types of queueing systems.

The agent has information only on parameters of its sector, it obtains information on parameters of other sectors by means of exchange of messages with other agents. For the considered system the most important parameters of effective work are an average waiting time of requests in queue and queue length, in this work queue length we accept as the balancing parameter. Each sector has reference parameters of service calculated from a condition that waiting time of a request in queue shouldn't exceed 10 min for the sectors serving students and not to exceed 15 min for the sectors serving departments of educational process support. We will designate reference queue length for the sector i through L_i.

Depending on parameters of system the state of the agent is defined. The set of states S of agent is described by 5 names: standard loading – s_1, high loading – s_2, $S^1 = \{s_1, s_2\} \subseteq S$, passive state – s_p, state of transfer – s_t, state of reception – s_r, $S^2 = \{s_p, s_t, s_r\} \subseteq S$. Standard loading is described by the following ratio:

$$L_i^c \leq 1.2 \cdot Li, \tag{7}$$

where L_i^c – the current expected queue length calculated taking into account the current intensity of an input stream. Loading level, higher than the standard, referred as high loading.

The state of transfer is the state when an agent carrying out redirection of requests, state of reception is the state when an agent receives requests, the state when agent isn't occupied with redistribution will be considered as passive. The current state of the agent is described by a combination of states from 2 sets:

$$s^1 \times s^2 = \{s_1, s_2\} \times \{s_p, s_t, s_r\}$$

Agents who can communicate and react to messages from other agents we will call as friendly. In the considered system we will consider as friendly the agents who are in same zone (friendliness of type 1 – fr_1, cost of unloading is 10%) and also the agents serving same type of clients (friendliness of type 2 – fr_2, cost of unloading is 15%). The cost of unloading represents a share of requests serving time growth in the new sector. Requests serving time increases owing to low mechanization of actions according to requests from other sector.

We will consider data exchange between sectors of different types in various zones inadmissible as it will have the high cost of unloading because of functions low similarity in diagonal sectors.

Interaction of agents is described by a set of rules for sending messages and reactions to messages of each agent.

Rules of Interaction of Agents:
Each agent accepts a flow of requests on an entrance. The agent A_i can transfer each request in one of directions: D_{ii} – to serve in the sector; D_{ij}, $i,j = \overline{1,4}$ – to transfer to agent with friendliness of type 1 or type 2.

Serving of a request has the cost T_j – serving time of the request in system. Serving a request in other sectors increases this cost. Serving of the request in a section with friendliness of type 1 has cost $T_{j,norm} = T_j \cdot 10\%$; 10% – expenses of lack of mechanization; Serving of the request in a section with friendliness of type 2 has cost $T_{j,norm} = T_j \cdot 15\%$.

As the parameter presenting the state of the current agent to other agents we will accept the value based on queue length L_i. As this parameter has different values for the different queues, in order to avoid cases of constant redistribution in favor of systems with high value of the parameter, it is necessary to carry out normalization. The idea of normalization is in leading the normalized values to sizes comparable among themselves, so that they could be compared directly. The most widespread way of normalization which we will also use, it:

$$L_{i,norm} = \frac{L_i^c}{L_i}. \tag{8}$$

Considering existence of cost for transfer a request to other agents, as decision-making value will be used $L_{i,s} = L_{i,norm} \cdot 20\%$, where 20% - the indirect expenses concerning a sector overload additional categories of functions.

Agents Communication Algorithm:

(1) at a starting time an agent is in the state $s_1 \times s_p$; counters of the redirected requests are zero;
(2) at receipt of a request in state $s_1 \times s_p$, put the request in queue to the current sector, execute recalculation of system states; if the state $s_2 \times s_p$ is reached, pass to step 3;
(3) send the message of type 1 to friendly agents whose marker of an overload isn't established;

– *message of type 1* – a message with information on overload state parameter;

(4) if receives the message of type 1 from the friendly agent, calculate the system state, send the agent message of type 2; compare the state to the parameter of the initiator of communication; if

$$L_{j,norm} > L_{i,s} \tag{9}$$

where $i \neq j$, A_i – current agent, A_j – initiator agent, pass to step 5;

– *message of type 2* – a message with information on the state parameter;

(5) pass into the state $s^1 \times s_r, s^1 \in S^1$, register the initiator;
(6) if receives the message of type 2 in state $s_2 \times s^2, s^2 \in S^2$, check the condition (9), if it is satisfied at least for 1 agent, pass into the state $s_2 \times s_t$, register recipient agent (−s), who will receive requests; pass to step 7; if the condition (9) isn't satisfied for any agent, send all friendly agents the message of type 3;

– *message of type 3* – a message about need of unloading;

(7) in state $s_2 \times s_t$, check the list of recipient agents, if the list is not empty, pass to step 8; if the list is empty, calculate the state of type s^1 and change the state of type s^2 to s_p;
(8) redirect the last request in the queue and information on its type to the last recipient agent in the list; register the redistribution; delete the recipient from the list; pass to step 8;
(9) if a request from an agent arrives in state $s^1 \times s_p, s^1 \in S^1$, reject the request and send to the system administrator the message of type 6;

– *message of type 6* – an error message with error details;

(10) if a request from an initiator agent arrives in state $s^1 \times s_r, s^1 \in S^1$, accept the request, delete the initiator from initiators list; check the list of initiators: if the list is empty, calculate the state of type s^1 and change the state of type s^2 to s_p;
(11) if a request arrives in state $s^1 \times s_r, s^1 \in S^1$ from non-initiator agent, reject the request, send to the system administrator the message of type 6;
(12) if in state $s^1 \times s_r, s^1 \in S^1$ is idle time more than 1 m, change the condition to $s^1 \times s_p, s^1 \in S^1$;
(13) if the transferred request was rejected by a recipient agent, put the request in queue to the current sector; send to the system administrator the message of type 6;
(14) if obtaining the message of type 3 from friendly agent, check the system state; if the state is $s_2 \times s_p$, then register the sender; check the list of agents whose marker of an overload is established; if not for all of friendly agents established the marker, then pass to step 3; if the marker established for all friendly agents, then send to the initiator agent message of type 5;

– *message of type 5* – a message about impossibility of unloading;

(15) if changing the state from $s_2 \times s_t$ to $s^1 \times s_p$, check the list of the senders of request for unloading; if the list is not empty, send all a message of type 4; compare the state to the parameter of the agent from the list; if condition (9) is satisfied, pass to step 5;

– *message of type 4* – a message with state parameter value;

(16) if the message of type 5 arrives from an agent, register agent as a marked and register the time of receipt of the message;
(17) if after marking of the agent there passed 30 min, then remove marking;
(18) if the message of type 4 arrives from an agent, check the condition (9): if it is satisfied, pass into the state $s_2 \times s_t$, register recipient agent; pass to step 7.

4 Experimental Results of the Developed Model and Algorithm

For calculation of SSC queues parameters the basic data revealed during observation at traditional service in departments have been defined. One hour of the working day is accepted as a unit of time. On the basis of average values of λ_i (clients/hour) and μ_i (clients/hour), the necessary number of service channels c_i for each queue and the corresponding average queue length L_i (clients) and average waiting time W_i (hours) for the condition $W_i \leq 10$ min for consulting sectors and $W_i \leq 15$ min for administration sectors has been counted (Table 1). At the same time λ_i values are not constant and can change. Output parameters of queues are calculated according to average values of arrival stream parameters and don't consider peak loading.

Table 1. Parameters of SSC queues

No. of queue	c_i	λ_i	μ_i	ρ_i	L_i	W_i
1	12	61	5.6	0.9	6.5	0.1
2	4	13	4.2	0.77	1.8	0.14
3	8	59	8	0.92	9	0.15
4	4	20	6	0.83	3.3	0.16

We will check the work of algorithm and efficiency of service process on the basis of simulation.

The queueing systems of SSC can't provide effective service at essential changes of input parameters. The simulation (Fig. 1) represents the result of system functioning at increase in entrance load in two of 4 queues by 20%. The lengths of overloaded queues grow infinitely.

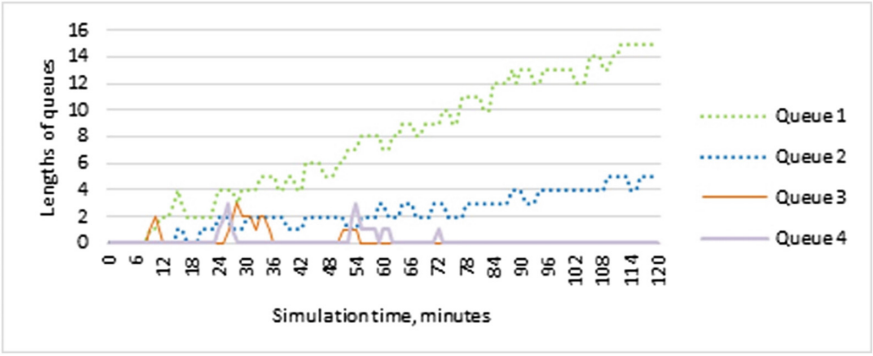

Fig. 1. Simulation of service process with constant overloading in some queues

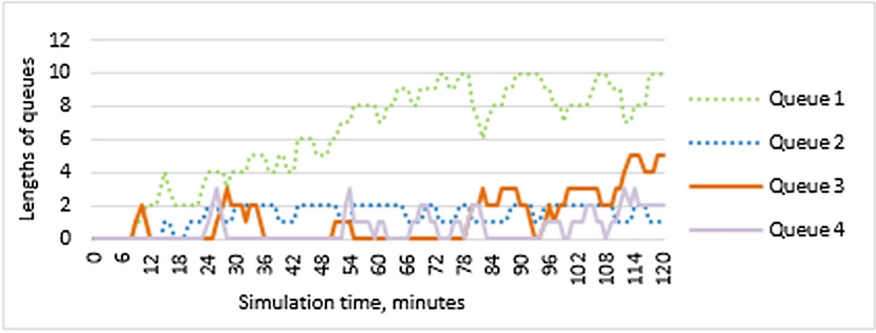

Fig. 2. The simulation result of load balancing according to the agents interaction algorithm

We can see in Fig. 2 that using the load balancing algorithm we can avoid infinitely growing queue length.

If there are periodic overloads in queues, despite gradual stabilization of length of queue, waiting time of the request in queue can go beyond admissible limits. Distribution of loading at a periodic overload in queues (Fig. 3) gives the chance to keep waiting time of requests in admissible limits.

Charts demonstrate that the load balancing using the proposed algorithm can become a solution of the problem of daily overloads in systems with parallel queues.

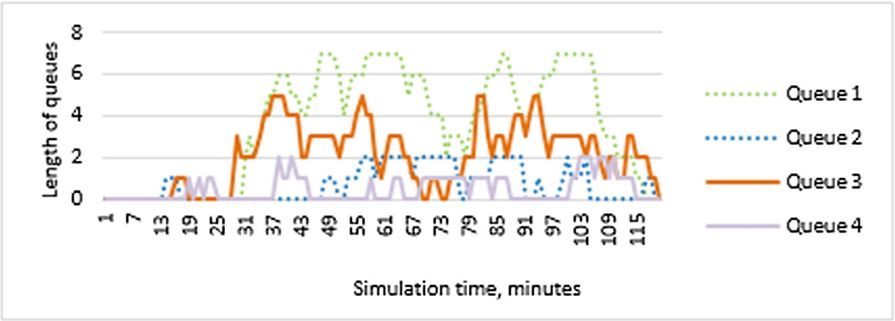

Fig. 3. The simulation result of load balancing according to the agents interaction algorithm in queues with periodic loading

5 Conclusion and Future Works

Thus, by means of methods of the queueing theory and agent-based load balancing many problems of planning, assessment and optimization of quality of SSC functioning can be solved. Agent-based approach allow to develop module system, which can be scaled easily: to add new sectors with new sets of functions and new nature of interaction with other sectors. Now, after optimization of service process of educational process supporting zones, future work consists in optimization of further service routes of requests from these zones, thereby, optimization of processes in queueing networks. It will be also important to present results of work of the algorithm with use of real entrance data on system parameters.

References

1. Adan, I., Resing, J.: Queueing Systems. Eindhoven University of Technology (2015)
2. Yuan, X., Hwarng, H.B.: Managing a service system with social interactions: stability and chaos. Comput. Industr. Eng. **63**(4), 1178–1188 (2012). https://doi.org/10.1016/j.cie.2012.06.022
3. Jiang, Y., Li, Z.: Locality-sensitive task allocation and load balancing in networked multiagent systems: talent versus centrality. J. Parallel Distrib. Comput. **71**(6), 822–836 (2011). https://doi.org/10.1016/j.jpdc.2011.01.006
4. Down, D.G., Lewis, M.E.: Dynamic load balancing in parallel queueing systems: stability and optimal control. Eur. J. Oper. Res. **168**(2), 509–519 (2006). https://doi.org/10.1016/j.ejor.2004.04.041
5. Li, X., Mao, W., Zeng, D., Wang, F.-Y.: Agent-based social simulation and modeling in social computing. In: Yang, C.C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P.S., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K., Mao, W., Zhan, J. (eds.) ISI 2008. LNCS, vol. 5075, pp. 401–412. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69304-8_41

6. Klügl, F., Timpf, S.: Approaching interactions in agent-based modelling with an affordance perspective. In: El Fallah-Seghrouchni, A., Ricci, A., Son, T.C. (eds.) EMAS 2017. LNCS (LNAI), vol. 10738, pp. 21–37. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91899-0_2
7. Szymanezyk, O., Dickinson, P., Duckett, T.: Towards agent-based crowd simulation in airports using games technology. In: O'Shea, J., Nguyen, N.T., Crockett, K., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2011. LNCS (LNAI), vol. 6682, pp. 524–533. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22000-5_54
8. Luo, J., Shi, Z., Wang, M., Huang, H.: Multi-agent cooperation: a description logic view. In: Lukose, D., Shi, Z. (eds.) PRIMA 2005. LNCS (LNAI), vol. 4078, pp. 365–379. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03339-1_29
9. Monostori, L., Valckenaers, P., Dolgui, A., Panetto, H., Brdys, M., et al.: Cooperative control in production and logistics. Ann. Rev. Control **39**(1), 12–29 (2015). https://doi.org/10.1016/j.arcontrol.2015.03.001. Elsevier
10. Seel, N.M. (ed.): Encyclopedia of the Sciences of Learning. Springer, Boston (2012)