

# Processing models for conflicting user requests in ubiquitous corporate smart spaces

Dmitriy Levonevskiy\*, Irina Vatamaniuk and Anton Saveliev

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), 39, 14<sup>th</sup> Line V.I., 199178, St. Petersburg, Russia

**Abstract.** This paper considers processing of conflicting user requests in ubiquitous corporate smart spaces. The formulated problem consists in the contradiction between the limitation of available smart space resources to perform the conflicting user requests and necessity to provide the proper quality of service in corporate smart spaces. The principles of constructing the simulation model are described. The experiments were carried out basing on a model of the SPIIRAS digital signage service. Several task management strategies are discussed, an assessment of their effectiveness is given. The research is aimed at improving the quality of service and user experience in human-computer interaction within the corporate smart spaces.

## 1 Introduction

Smart spaces provide the most matter-of-course way of interaction between users and information environment of organizations [1]. The success of smart space implementation depends on three directions in science and technology: ubiquitous computing, ubiquitous communication and multimodal interfaces [2]. The easiness of access to information as well as the convenience of available user interfaces are the distinguishing features for such systems. Nevertheless, there are still a set of problems to solve while developing and deploying ubiquitous multimodal systems of human-computer interaction. These problems are related to information security, reliable and efficient node linking, managing heterogeneous devices, task priorities, assessing quality of service, etc. One of the significant aspects of task management is processing of multiple requests generated simultaneously by different users and securing the proper quality of information perception while interacting with multiple users. This paper discusses available approaches to clarify the problem mentioned above.

## 2 Related work

The main principles of providing the proper quality of multi-user service in multimodal distributed systems are considered in [3]. The authors propose a solution to handle multiple user requests for a multiscreen software-defined IoT digital signage system. The proposed method is based on the dynamic link allocation process which is derived due to the software-defined networking (SDN) architecture of the system. In such system, the functionality of the physical layer is decoupled from the control layer. So, the SDN controller of the digital

signage cloud network estimates the network status and the number of user queries, and automatically updates the rules for query processing and reallocates system resources. Also, the authors propose the original technique for so called invisible communication which provides user with a tool for an easy access to the network services. Such interactivity between stationary terminals and user mobile devices substantially expands user experience.

As a rule, index of perceived quality is the most significant one for media streaming from the point of view of user experience. Its estimation and prediction based on the extended SSIM metric (SSIMplus index) are discussed in [4]. Recent approaches based on various techniques (machine learning, software-defined networking, flow dynamics et al.), can be found in works [5-7]. Estimation of the perceived quality has a great potential to provide the best possible service to each individual user in distributed systems, especially in multiuser ones.

Quality of visual service provided by a system of digital signage is discussed in [8]. This metric is determined based on the level of 2D visual sensitivity, and on the ability to perform 3D binocular fusion by users located at various viewing distances. Its use allows automatic control of visual information and radio resources for multiple users without additional interactions.

Systems of digital signage (DS) find their application in retail services, third-party advertisement, entertainment, wayfinding, education, and transportation. In [9], an architecture for providing quality of service (QoS) and mobility-aware telescreen services in vehicular networks is presented. It allows managing user registrations, service advertisements, network information, and service quality due to the adaptive

\* Corresponding author: [DLewonewski.8781@gmail.com](mailto:DLewonewski.8781@gmail.com)

selection of QoS parameters and a handover decision mechanism for providing seamless imagery effect and enhance user's multimedia perception.

A multi-screen environment provides a new opportunity for digital signage applications as an effective means for advertisements and information [10]. A software-based synchronization is proposed in that work to address the frame-level synchronization between multiple screens. It provides better performance and extends possibilities of a DS system by allowing the use of non-standard screens and screens positioned unconventionally in space.

In [11] QoS and quality of experience (QoE) Aware N-Screen Multicast Service is proposed. In such multiscreen system, users are dynamically combined in multicast groups depending on their devices' capacity and internet traffic. The perceived quality of the multimedia content provided to each group is dynamically adjusted to meet better the QoE requirements according to available resources of user devices. This scheme helps to optimize the use of network resources for specified quality content requirements.

### 3 Model development

A typical application area of distributed multimodal systems is corporate smart spaces that provide ubiquitous human-computer interaction between users and components of the smart space. As an example, we consider the process of interaction between users and MINOS (Multimodal Information and Navigation Cloud System) information service [12]. The information and navigation services of MINOS are implemented via ensemble of webcams, monitors and information screens helping visitors to navigate within the corporate environment of SPIIRAS. The webcams are involved in processes of registration, authentication and recognition of users and allows automating work of checkpoint and reception. The monitors and touchscreens help users to get required information easily.

Users generate requests by means of different modalities (speech, gestures, etc.). They can also control the service using their mobile phones or personal computers (for administrators). Each request needs to be processed, i.e. some task has to be performed and requires allocation of some resource (display and sound channels, user session) during some time. In case when there is only one user working with the system at a certain moment of time, every new request implies suspending any other requests if they require the same resource, so there is no conflict between them. But in case of multiple users, the system may encounter the impossibility of allocating the needed resource for all tasks. Then it is necessary to determine the optimal strategy of request managing that results in the highest possible QoS metrics.

Each request is characterized by a set of properties that include the original user identifier, the session identifier, the active modality and the requested task.

The optimality of a certain strategy depends on a set of parameters (request intensity, resource requirements of tasks, number of users, etc.).

Let us consider the task classification in MINOS:

- by distribution: local (should be processed by only one channel), distributed (should be processed by several channels), global (use all available channels);
- by priority: critical (require immediate suspending all other tasks), regular (have a relative priority in a service discipline), background (are processed only if there are no other tasks and immediately suspended by any non-background tasks);
- by execution procedure: continuous and suspendable;
- by length: with constant or variable length;
- by bindings to other tasks: single, sessional.

The request structure can be described more detailed in JSON format:

```
request: {
  user: {
    id,
    role
  },
  session: {
    id,
    start_timestamp
  },
  modality,
  task: {
    id,
    options,
    resumable,
    resources: [ {
      id,
      requiredAmount,
      requiredTime
    } ]
  }
}
```

The task priority depends on these options. So, to obtain QoS values, we should define:

- source data;
- method of priority designation;
- task management strategy.

Let us consider these aspects in more detail.

Source data include information about the process of each request type generation. The conditions of task generation are not strictly defined and may change depending on the context. However, according to the central limit theorem of the probability theory for event streams, the process of single request entrance can be described using the Poisson distribution, and its parameter ( $\lambda$ ) characterizes the intensity of request emergence. For priority designation, the following principles have to be taken into account.

Firstly, the probability of accurate perception and recognition of multimodal requests is very significant. Requests sent from mobile devices and PC's are recognized the most accurately, because a user has to start the service application, open a session, scan the QR code of the smart space component (for mobile devices) and send a command through the application. These

actions must be performed purposefully, and the probability of their accidental execution is extremely small. In case when speech recognition is applied, the probability of a mistake is higher, for gesture-based control it is even higher [13].

Secondly, credibility and legitimacy of incoming requests are to be considered. Lack of credibility may take place, for example, if the request flow from a certain user does not look like a human-generated one (it can possess a very high frequency of events or be contradictory). This problem can be caused by perception inaccuracy as well as malicious activity (for example, when a bot connects to the service interface). In the latter case the requests are illegitimate. In both cases some requests are denied or assigned the lowest priority.

Thirdly, task processing technique should depend on the possibility of the user to receive the requested data. So, if there is a request from a user assuming some data output on a stationary display, but this user is out of the display’s visibility zone, the task can be delayed or cancelled after a defined time-out.

Fourthly, the task itself has properties like urgency defined automatically by the task type (or the user priority) or manually by the administrator, suspendability, etc.

Task management strategy types are studied well in the theory of queuing systems [14]. Buffering and service disciplines differ in priority presence, rules of task selection, queue length, etc. Any new task can be processed in one of the following way:

- immediate processing (possibly replacing current task(s));
- immediate cancellation;
- adding to the queue.

In the latter case disciplines like FIFO (First In, First Out) and LIFO (Last In, First Out) may be used. To simulate the specified processes, we have implemented a model described in the next Section.

## 4 Experiments

The conducted experiments consist in the simulation of the incoming user request flow and the process of performing the requests. The output of the simulation reflects the efficiency of implemented strategies. While modeling, several assumptions listed below were considered.

At first, this model does not require the background tasks to be taken into account as their execution does not affect processing the tasks with higher priority at all. Critical tasks are also omitted due to their low probability and irregularity.

At this stage of work, we simulated only local tasks to simplify the experiments. In general, processing of a new task consists of the following steps:

1. If there is enough available resource to process the task, it is performed immediately.
2. If the resource is in use by tasks of the same user, these tasks are cancelled, and the new task is performed.
3. In other cases, the system uses one of the strategies described further.

Strategies without priority include:

1. Systems with denials: the task is cancelled if there is no enough resource.
2. Systems with limited waiting time/queue length. Strategies with priority may imply:
  1. Shift tasks with lower priorities.
  2. Suspend tasks with lower priorities.
  3. Cancel tasks with lower priorities.

To assess the quality of task processing several values can be measured:

- percentage of the fulfilled tasks (relative throughput);
- percentage of the fulfilled tasks weighed by their priority;
- average waiting time;
- percentage of resource/channel usage.

The system model used in the experiments is implemented in Python. It takes the following input data:  $N$  types of single requests from different users, request intensities  $\lambda_i, i = 1 \dots N$ , task priorities  $P_i$  depending on the request modalities. The requests consist in user inquiries about help and reference information. The model is able to use strategies mentioned above as well as their combinations. Three priority levels that correspond to the modalities (speech, gestures, mobile application) were used. On appearance of a new request, some action is performed:  $A_H$  (if a new task is of a higher priority),  $A_E$  (equal priorities),  $A_L$  (lower priority). In case when  $A_H = A_L = A_E = A$ , the strategy without priorities is used.

The result of processing of every task is characterized by success and delay (waiting time). Strategies with queues show higher delays, but higher average success level at the same time. Strategies where denials prevail have lower delays and lower success level. To obtain a complex strategy estimate, an effect index for the task  $i$  may be composed. This index depends on both success and delay. In case of denial, we can assume that the task delay  $D_i = \infty$ , and the following estimate of the quality of service may be used to proceed to finite values:

$$E_i = \frac{D_0}{D_0 + D_i}, \tag{1}$$

where  $D_0$  is the delay considered to reduce the effect estimate  $E_i$  twice.

If a task can be suspended with a partial effect, actual time of its processing should also be taken into account. In this case  $E_i$  may be represented as

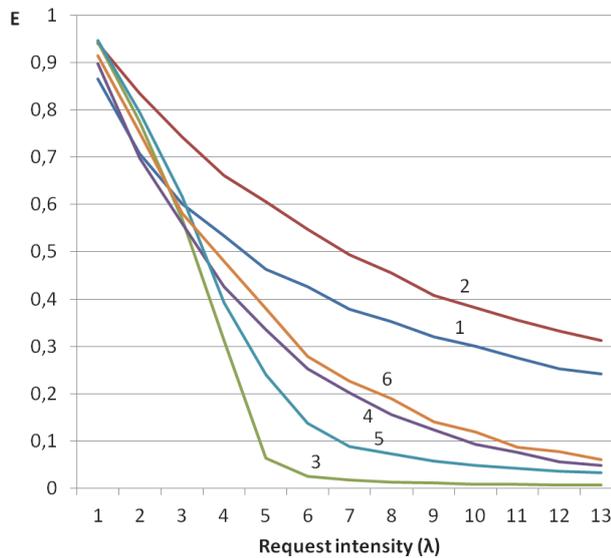
$$E_i = \frac{T_{Fi} D_0}{T_{Pi} (D_0 + D_i)}, \tag{2}$$

where  $T_{Fi}$  is the actual time of the task  $i$  and  $T_{Pi}$  is the planned time. To get an integrated estimate of a certain strategy using task priorities, we can use the following formula:

$$E = \frac{\sum_{i=1}^N P_i E_i}{\sum_{i=1}^N P_i}, \tag{3}$$

where  $P_i$  is the priority of the task  $i$  and  $N$  is the number of tasks.

If the request intensity is low, the tasks have a relatively small number of conflicts, and the value  $E$  is near 1 and does not depend significantly on the strategy. With the growth of intensities, different strategies result in different effects. Figure 1 illustrates the effect dependencies on the request intensity if some common strategies without priorities are applied.



**Fig. 1.** Efficiency of the strategies without priority: 1 – denial, 2 – cancelling conflicting tasks, 3 – FIFO, 4 – LIFO, 5 – FIFO (with maximum delay 1.0), 6 – LIFO (with maximum delay 1.0).

The graph shows that strategies with denials are more efficient than ones with queues if the tasks are treated as if they were of an equal priority. In particular, this can be explained by the effect sensitivity to delays. In the experiments it was assumed that  $D_0 = \bar{T}_p$ , which means a considerably low sensitivity in the terms of human-computer interaction. The interface response time by which the system is perceived by the user as controllable does not exceed several seconds [15]. Thereby, it is evident that any stricter conditions for  $D_0$  will only reduce the estimate for strategies with queues.

It is also noticeable that the limitation of queue length in strategies with queues affect positively on the strategy estimate because this condition leads to cancellations of tasks with high delays that hold the resource but do not make a perceptible impact on the effect estimate. Nevertheless, such strategies concede to the ones with denials.

Then, let us consider the strategies with priorities and compare them to the most efficient strategies without priorities. The conducted experiment included the set of strategies that have at least two different types of actions among  $A_H, A_L, A_E$ , and these actions do not contradict the task priorities. The graphs for the most efficient strategies revealed are shown on Figure 2.

Figure 2 illustrates the effect values for the following strategies:

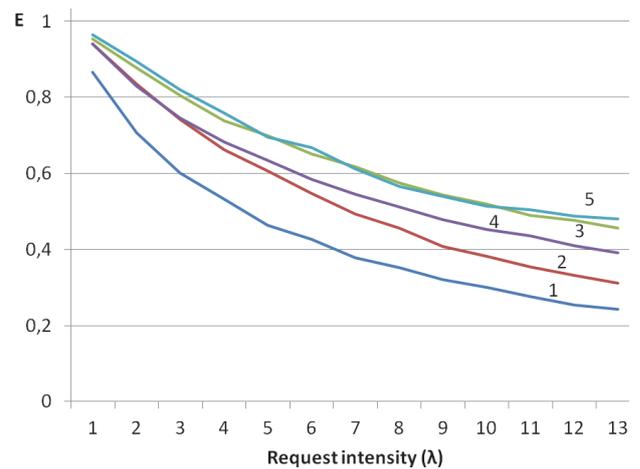
1: denial ( $A = \text{denial}$ );

2: cancel conflicting tasks ( $A = \text{cancellation of other tasks}$ );

3: cancel conflicting tasks if the new task has an equal or a higher priority ( $A_L = \text{denial}, A_H = A_E = \text{cancellation of other tasks}$ );

4: cancel conflicting tasks only if the new task has a higher priority ( $A_L = A_E = \text{denial}, A_H = \text{cancellation of other tasks}$ );

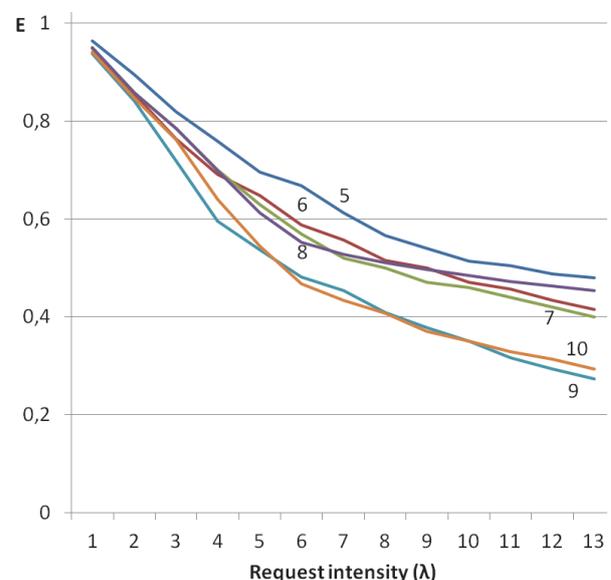
5: add the new task to the FIFO queue only if it has a lower priority, otherwise cancel conflicting tasks ( $A_L = \text{FIFO}, A_E = A_H = \text{cancellation of other tasks}$ ).



**Fig. 2.** Comparison of the strategies without priority (1, 2) and with priority (3, 4, 5).

For the majority of strategies, the simulation shows that adding tasks to the queue results in the lower value of the effect  $E$ . However, using the FIFO queue for processing low-priority tasks leads to a moderate rise of the effect value. This can be explained by the fact that denials for these tasks will cause  $E_i = 0$ , but using queues allows them to retain some probability of execution providing that there are no prevailing tasks. Thus, this strategy results in  $E_i \geq 0$ .

Experiments with the other strategies with priorities show that their efficiency is not so high (Figure 3).



**Fig. 3.** Efficiency of the strategies with priority.

Figure 3 illustrates the strategies:

5: add the new task to the FIFO queue only if it has a lower priority, otherwise cancel conflicting tasks ( $A_L = \text{FIFO}$ ,  $A_E = A_H = \text{cancellation of other tasks}$ ).

6:  $A_L = \text{denial}$ ,  $A_E = \text{cancellation of other tasks}$ ,  $A_H = \text{LIFO}$ ;

7:  $A_L = A_E = \text{cancellation of other tasks}$ ,  $A_H = \text{LIFO}$ ;

8:  $A_L = \text{FIFO}$ ,  $A_E = \text{cancellation of other tasks}$ ,  $A_H = \text{LIFO}$ ;

9:  $A_L = \text{FIFO}$ ,  $A_E = A_H = \text{LIFO}$ ;

10:  $A_L = \text{FIFO}$ ,  $A_E = \text{LIFO}$ ,  $A_H = \text{cancellation of other tasks}$ .

So, it can be seen that under given conditions the 5<sup>th</sup> strategy is optimal. It uses the FIFO queue for low-priority tasks and cancels the conflicting tasks in other cases.

## 5 Conclusion

Choosing the right query strategy allows making the human-computer interaction friendlier, increasing the quality of service and improving user experience of interacting with corporate smart spaces.

The directions of further research include:

1. Analysis of sessional request features, working out the methods of determining their credibility and legitimacy and search of the optimal strategies of their processing. Sessional requests make the event flow not ordinary, but the processes of generating sessions and requests within a certain session may still be considered ordinary, so the models for Poisson event streams can be applied for these processes.

2. Analysis of interaction via several data output channels, homogeneous or using different modalities. These channels may include video and audio channels of stationary smart space nodes or portable user devices, as well as channels of human-robot interaction. The theory of multichannel queuing systems can be the theoretical basis for this study.

3. Analysis of how distinct types of tasks affect the efficiency of the smart space. In particular, this includes classifying tasks of various applied MINOS services besides Digital Signage (navigation, videoconferencing, etc.).

4. In the field of information security, there is a need to study the problem of trust to the user. In particular, the smart space has to be resistant to attacks from malicious users. This can be provided by means of integration with known methods of protection against attacks that are critical for smart spaces (denial of service, spoofing, etc.) [16]. There are also useful methods of protecting the smart space by distinguishing unreliable (deliberately false or mistaken) information [17]. Another approach to protect the smart space from incorrect information is the integrated simultaneous processing of data gathered using different modalities [18].

The presented work was supported by the Russian Science Foundation (grant No. 16-19-00044).

## References

1. X. Zeng, H. Pei, International Conference on Information Computing and Applications 2012, **308**, 628–634 (2012)
2. R. Yusupov, A. Ronzhin, Herald of the Russian Academy of Sciences, **80**, 63–68 (2010)
3. M.A. Hossain et al., Int. J. of Distributed Sensor Networks, **7/12**, 1–14 (2016)
4. Z. Wang, K. Zeng, A. Rehman, H. Yeganeh, S. Wang, Applications of Digital Image Processing XXXVIII, **9599**, 95990Y (2015)
5. Y.T. Lin, E.M.R. Oliveira, S.B. Jemaa, S.E. Elayoubi, 2017 IEEE International Conference on Communications (ICC), 1–6 (2017)
6. H. Nam, K.H. Kim, J.Y. Kim, H. Schulzrinne, Global Communications Conference (GLOBECOM). 1317–1322 (2014)
7. Y. Xu, S. E. Elayoubi, E. Altman, R. El-Azouzi, Y. Yu, IEEE Transactions on Mobile Computing. **15** (11), 2762–2780 (2016)
8. H. Oh, H. Lee, I. Lee, S. Lee, (2015). Dig. Sig. Proc., **45**, 24-35 (2015)
9. G. Sarwar, F. Ullah, H.W. Lee, W. Ryu, S. Lee. Tehnički vjesnik, **24** (1), 177–185 (2017)
10. I. Shin, S. Lee, E. Lee, N.K. Lee, H. Lee, H. ETRI J., **38** (5), 868–878 (2016)
11. G. Sarwar, F. Ullah, S. Lee. J. of Sensors. **6**, 1–11 (2016)
12. I. Vatamaniuk, D. Levonevskiy, A. Saveliev, A. Denisov. 18th International Conference on Speech and Computer (SPECOM-2016), **9811**, 588–595 (2016)
13. H. Liu, L. Wang, Int. J. of Industrial Ergonomics, (to be published)
14. A. Penttinen. *Introduction to Teletraffic Theory* (2003)
15. S. Henty. UI Response Times. URL: <https://medium.com/@slhenty/ui-response-times-acec744f3157>
16. P. Eder-Neuhauser, T. Zseby, J. Fabini, G. Vormayr. Sust. Energy, Grids and Networks, **12**, 10–29 (2017)
17. V. Budkov, I. Vatamaniuk, V. Basov, D. Volf. 18th International Conference on Speech and Computer (SPECOM-2016), **9811**, 419–426 (2016)
18. K. Margiotoudi, S. Kelly, A. Vatakis. Procedia - Social and Behavioral Sciences, **126**, 154–155 (2014)