


# Scheduling algorithm for the picture configuration for secondary tasks of a digital human–computer interface in a nuclear power plant

Gang Zhang<sup>1</sup>, Xuegang Zhang<sup>1</sup>, Yu Luan<sup>1</sup>, Jianjun Jiang<sup>2</sup>   
and Hong Hu<sup>2</sup>

## Abstract

Secondary tasks of a digital human–computer interface in a nuclear power plant increase the mental workloads of operators and decrease their accident performance. To reduce the adverse effects of secondary tasks on operators, a picture configuration scheduling algorithm of secondary tasks is proposed. Based on the research background and operator interviews, a scheduling algorithm process is established, and variables and constraint conditions of the scheduling process are defined. Based on the scheduling process and variables definitions, this article proposes a picture feature extraction method, a method for counting identical keywords, an arrangement method of queues in a buffer pool and a picture configuration scheduling algorithm of secondary tasks. The results of simulation experiments demonstrate that the algorithm realizes satisfactory performance in terms of the number of replacements, the average waiting time, and the accuracy.

## Keywords

Digital human–computer interface, a picture configuration scheduling algorithm, buffer pool, constraint conditions

Date received: 27 November 2019; accepted: 16 February 2020

Topic: Robot Manipulation and Control  
Topic Editor: Andrey V Savkin  
Associate Editor: Bin He

## Introduction

An operator must perform his or her not only primary tasks but also secondary tasks of digital human–computer interfaces (HCIs) in a nuclear power plant (Npp) to deal with an accident.<sup>1</sup> The secondary tasks are also known as interface management tasks. Interface management tasks mainly include navigation, configuration, arrangement, interrogation, and automation.<sup>2</sup> An operator must execute secondary tasks to support primary tasks because many parameters and navigations and a substantial amount of information must be configured to correctly deal with an accident.

An operator's cognitive resources must be distributed when an accident is being addressed. If the allocated

cognitive resources outweigh the support capability of an operator, task performance will decline<sup>3</sup> because the cognitive resources of any operator are limited. Then, if

<sup>1</sup>State Key Laboratory of Nuclear Power Safety Monitoring Technology and Equipment, China Nuclear Power Design Company Ltd, Shenzhen, Guangdong Province, China

<sup>2</sup>School of Safety and Environment Engineering, Hunan Institute of Technology, HengYang, HuNan Province, China

### Corresponding author:

Jianjun Jiang, School of Safety and Environment Engineering, Hunan Institute of Technology, HengYang, HuNan Province 421002, China.  
Emails: jjjhnt@126.com; jiangjianjun310126@126.com; 13807474256@126.com



secondary tasks consume additional cognitive resources of an operator, the mental load and work performance of the operator will be affected.

Compared with the traditional operating control platform, a digital HCI provides operators with abundant information and parameters. The information and parameters on any display are not fixed; however, charts and graphs are discontinuous, which will increase the cognitive load of operators, consume their attentional resources, and generate keyhole effects.<sup>4</sup> Then, misreading, misjudgment, and misoperation will easily occur, which will increase the probability of human-factor accidents.

With the rapid development of science and technology, artificial intelligence technology has made great achievements. Intelligent and mechanized machine instead of cumbersome human operation has gradually become true. The flexibility and intelligence of robot control can make up for the security risks and the lack of efficiency and accuracy of manual operation or inspection. If the pictures for secondary tasks of a digital HCI can be intelligently configured by robot technology, operators' cognitive resources and the time of dealing with an event will be decreased, and the accidents caused by human errors will be decreased, so pictures configuration is necessary.

Three core technologies of intelligent system are robot technology, artificial intelligence, and digital technology, respectively, in which, robot technology is the key problem. For robot technology, software control technology is the core of the whole robot control system. To decrease the disturbance from secondary tasks, based on the software control technology of robot this article studies a scheduling algorithm that can be used for picture configuration for secondary tasks. When an operator must obtain parameter information, if the operator need not configure related secondary tasks, he can save time and decrease his cognitive load.

The research achievements regarding secondary tasks are few. Most studies focus on human-machine interfaces (HMIs). In 2011,<sup>5</sup> a visual strategy is used to design an interface between a human and a computer. The design strategy keeps in mind of human beings and on the assumption that the HMI should be as simple as possible. To improve highlighting in an HMI, Anuar and Kim<sup>6</sup> proposed a systematic method for an automatic system of Npps. Bhatti et al.<sup>7</sup> presented a user-centered design strategy that includes operation contexts and relevant interfaces that are suitable for users and standard designs. In 2015,<sup>8</sup> a particle swarm optimization method with weights was proposed for optimizing a complex problem. In 2009,<sup>9</sup> input performance, user comfort, and interface layout were studied. The study shows that input and comfort performances can be improved by optimizing the interface layout. Later, the topological structure and integrated design of the component layout and the shape of the HMI were studied based on a finite element network and a collision detection algorithm.<sup>10,11</sup>

Some scholars studied how the HCI design of warehouse orders affects the perceived load, usability, comfort, and operation performance, and experimental data show that graphic user interfaces can reduce operation time of tasks and human error.<sup>12</sup> In the process of industrial operation, HCI can help operators get familiar with the factory state and deal with unexpected events. Therefore, some scholars put forward the idea of ecological interface design and a dynamic interface design model, which have been applied.<sup>13</sup> Aiming at the diversity in device interaction process, some scholars proposed a multi-objective and multi-mode interaction modeling method based on the interface description language, which could improve the usability of HCI end-user interaction.<sup>14</sup> For the disabled who have difficulty in moving, some scholars studied the HCI based on the gesture interaction mode. The research process used mobile device robot platform, 3D image sensor, identification system based on the support vector machine, and vehicle positioning equipment.<sup>15</sup> Some scholars studied the HMI design for the enterprise online product trading platform. The experimental results show that color plays an important role in awakening customers and that warm and cool colors have different influence on people.<sup>16</sup>

Through simulative experiments, Kantowitz et al. found that interface management tasks reduced the performance of first tasks, and had a direct impact on the reliability for an operator to complete first tasks.<sup>17</sup> Tijerina et al. tested interface management tasks had influence on professional operators of heavy vehicles, namely, interface management task had certain influence on reliability of professional operators.<sup>18</sup> To reduce the adverse impact of the interface management task on the operator, Howard and Kerst proposed that the interface management should be organized into a physical space model that could be easily recognized by the methods of path tracking, backtracking, status identification, and scope limitation.<sup>19</sup> To improve the readability and visibility of the interface management task and reduce the attention resources allocation of operators, Cook and Woods proposed that the characteristics of interface management task could be moved to the data area using analog input device, data control device, and computer monitoring system.<sup>20</sup> The study confirmed that if two tasks are very similar, there is a learning transfer from one secondary task to another secondary task.<sup>21</sup> Under the background of secondary task, to explore the combined effect of anxiety, cognitive load, experience, the researchers designed experiments with secondary tasks, and without secondary tasks, respectively. The experimental situation is set as lower anxiety and higher anxiety. Eleven professionals and 10 novices participated in the experiment; the results show that the anxiety causes performance degradation for the novice and that secondary tasks increase mental load and reduce the rate of response.<sup>22</sup> In a concurrent eye task, some scholars tested whether a manual type secondary task could increase the awareness of eye movement error. The experiment found that the difficulty of a task had no

effect on the awareness of eye movement error, and the participants' ability to monitor eye movement improved with the increase of interference.<sup>23</sup>

In addition to these studies, other achievements regarding the design and evaluation methods of HMI have been realized, such as a virtual environment and a constraint genetic algorithm<sup>24,25</sup> and evaluation methods of HMI.<sup>26–29</sup> Naujoks et al.<sup>30</sup> studied the automation of longitudinal and lateral control during an on-road experiment in everyday traffic. The results demonstrated that driving safety with subjectivity or objectivity was not influenced by the degree of automation. A model for determining the likelihood of a driver's involvement in secondary tasks based on attributes of driving behavior was developed. The model could be applied in crash investigations to resolve legal disputes in traffic accidents.<sup>31</sup>

The descriptions above indicate that secondary tasks give interference for an operator, affect the operators' execution of first task, increase psychological load, and affect the attention resources distribution. To decrease the mental load and distribution of the cognitive resources for operators, based on robot technology, this article proposes a scheduling algorithm for picture configuration of secondary tasks of HCI in an Npp. The article has two main contributions that are listed as follows: (1) the proposed method can be used to automatically configure pictures, which can reduce the time that is spent dealing with an accident and decrease the mental stress of operators, so that the incidence of human-factor accidents can be decreased and (2) the method is established under certain conditions including digital system features and constraint conditions, so the proposed method is more in line with the actual situation.

## Scheduling process and constraint conditions

### Scheduling process

Based on the research background and operator interviews, the process of picture configuration scheduling mainly includes the following: acquiring priority, organizing data, tracking dynamic processes, and using a replacement algorithm. Figure 1 illustrates the process of picture configuration of secondary tasks.

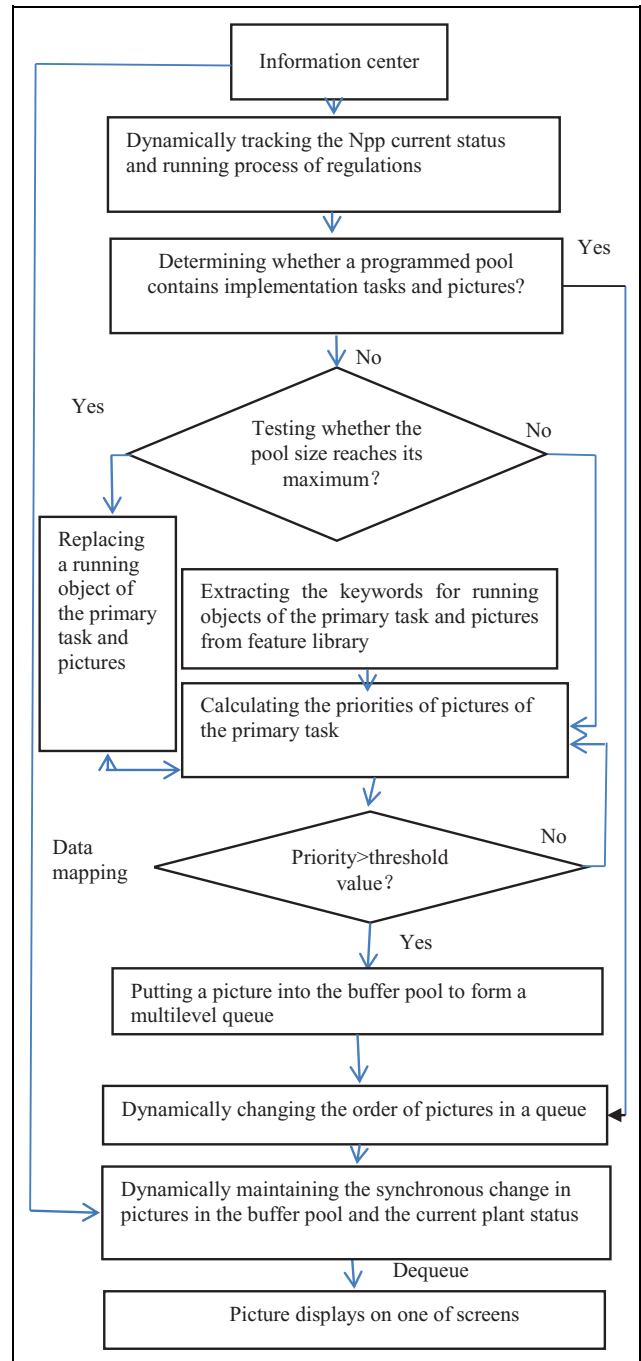
### Constraint conditions of the picture configuration process

**Notations.** Notations are listed below:

- Buffer*: a buffer pool that is used to save related pictures and primary tasks;
- Task<sub>f<sub>i</sub></sub>*

*Task<sub>f<sub>i</sub></sub>*: an implemented object of the *i*th primary task;

*K<sub>time\_long\_task</sub>*: implemented objects that have been recently visited;



**Figure 1.** Process of the picture configuration scheduling algorithm.

- task<sub>s<sub>ij</sub></sub>*: the *j*th picture that is associated with the implemented object of the *i*th primary task;
- size(cur\_task)*: the size of the current implemented objects for the *i*th primary task;
- size(task<sub>s<sub>ij</sub></sub>*): the size of the *j*th picture that is associated with the implemented object of the *i*th primary task;
- size(cur\_sec\_task)*: the sum of all pictures that are related to currently running objects;

*cur\_sec\_task*: all pictures that are related to currently running objects;

*v\_time(task<sub>f</sub>)*: recent visitation time of implementing objects that are related to the *i*th primary task;

*cur\_task*: objects that are being implemented in current primary tasks;

*cur\_f*: the current picture;

*Suff\_size*: the size of the buffer pool;

*F\_t\_size<sub>i</sub>*: the stored size of the implemented object for the *i*th primary task;

*G\_inf\_size<sub>ij</sub>*: the stored size of the *j*th picture that is associated with the implemented object of the *i*th primary task;

*M*: the number of implemented objects of the primary task in the buffer pool;

*N<sub>ij</sub>*: the number of the *j*th picture that is related to the implemented object of the *i*th primary task;

*U\_sum<sub>ij</sub>*: the number of visitations of the *j*th picture that is related to the implemented object of the *i*th primary task;

*S\_f\_sum<sub>i</sub>*: the number of visitations of the implemented object of *i*th primary task;

*Ft<sub>i</sub>*: the visitation frequency of the implemented object of the *i*th primary task;

*mt<sub>i</sub>*: the importance degree of the implemented object of the *i*th primary task;

*Fg<sub>ij</sub>*: the visitation frequency of the *j*th picture that is related to the implemented object of the *i*th primary task;

*gm<sub>ij</sub>*: the importance degree of the *j*th picture that is related to the implemented object of the *i*th primary task;

*pri\_w<sub>ij</sub>*: the priority of the *j*th picture that is related to the implemented object of the *i*th primary task;

*fp<sub>i</sub>*: the weight of the implemented object of the *i*th primary task;

*w<sub>f</sub>*: the threshold value of the implemented object weight of the *i*th primary task;

*k\_w\_f<sub>i</sub>*: extracted keyword vector space of the implemented objects of the *i*th primary task;

*s(k\_w\_f<sub>i</sub>)*: the number of extracted keywords of the implemented objects of the *i*th primary task;

*k\_w\_s<sub>ij</sub>*: the keyword vector space of the *j*th picture that is related to the implemented object of the *i*th primary task;

*s(k\_w\_s<sub>ij</sub>)*: the number of extracted keywords of the *j*th picture that is related to the implemented object of the *i*th primary task;

*sim(k\_w\_f<sub>i</sub>, k\_w\_s<sub>ij</sub>)*: the similarity degree between the implemented object of the *i*th primary task and the *j*th picture that is related to the *i*th primary task;

*vf<sub>ik</sub>*: the extracted *k*th keyword of the implemented objects of the *i*th primary task;

*vs<sub>ijp</sub>*: the *p*th keyword of the *j*th picture that is related to the implemented object of the *i*th primary task;

*pri\_f*: priority threshold value;

*f\_c*: feature library;

*c\_sum*: the number of keywords in the feature library;

*f\_s\_p\_s<sub>ij</sub>*: the number of the identical keywords between the implemented object of the *i*th primary task and the *j*th picture that is related to the *i*th primary task;

*f(cur\_inf)<sub>ij</sub>*: the current status of the plant with the *j*th-newest picture that is related to the implemented object of the *i*th primary task;

*t\_s\_inf<sub>ij</sub>*: the current data or parameters of the *j*th picture that is related to the implemented object of the *i*th primary task;

*flag*: indicator of whether the implemented object of the *i*th primary task is changed;

*change<sub>ij</sub>*: indicator of whether the *j*th picture that is related to the *i*th primary task is changed in the running process.

#### Constraint conditions.

1. The buffer pool size must be greater than or equal to the sum of the sizes of the implemented objects of the primary task and the related pictures, which can be expressed as follows

$$suff\_size \geq \sum_{i=1}^m f\_t\_size_i + \sum_{i=1}^m \sum_{j=1}^{n_{ij}} g\_inf\_size_{ij} \quad (1)$$

2. The visitation frequency of the *j*th picture that is related to the implemented object of the *i*th primary task is as expressed in equation (2)

$$fg_{ij} = \frac{u\_sum_{ij}}{\sum_{i=1}^{n_{ij}} u\_sum_{ij}} \quad (2)$$

Similarly, the visitation frequency of the implemented object of the *i*th primary task is as follows

$$ft_i = \frac{s\_f\_sum_i}{\sum_{i=1}^{n_{ij}} s\_f\_sum_i} \quad (3)$$

3. The weight of the implemented object of the *i*th primary task is defined as

$$fp_i = ft_i \times mt_i \quad (4)$$

4. The buffer pool is initialized to determine which objects of the primary tasks should be added into it. The condition is expressed as follows

$$fp_i \geq w_f \quad (5)$$

5. The similarity degree between the implemented object of the *i*th primary task and the *j*th picture that is related to the *i*th primary task is defined as

$$sim(k\_w\_f_i, k\_w\_s_{ij}) = \frac{f\_s\_p\_s_{ij}}{s(k\_w\_f_i) + s(k\_w\_s_{ij})} \quad (6)$$

6. The priority is calculated via equation (7)

$$pri\_w_{ij} = 0.7 \times sim(k\_w\_f_i, k\_w\_s_{ij}) + 0.2fg_{ij} + 0.1gm_{ij} \quad (7)$$

7. The sum of the sizes of the implemented objects, all pictures that will be added into the buffer pool in the immediate future and all pictures that are currently in the buffer pool must be less than or equal to the buffer pool size, which can be expressed as follows

$$size(cur\_task) + size(cur\_sec\_task) + \sum_{i=1}^m task\_f_i + \sum_{i=1}^m \sum_{j=1}^{N_{ij}} size(task\_s_{ij}) \leq suff\_size \quad (8)$$

For pictures or tasks in the buffer pool:

- ① If  $pri\_w_{ij} < pri\_f$ ,  $task\_s_{ij}$  will be removed from the buffer pool.
- ② If  $pri\_w_{ij} \geq pri\_f$ ,  $Task\_f_i$  is added into the  $i$ th queue of the buffer pool and the queue is reordered.

- (9) The values of “flag” are defined as follows:

- ① If  $flag = 1$ , the current implemented object should be added into the buffer pool and it will be ready for configuring the pictures that are related to the implemented object.
- ② If  $flag = 0$ , pictures that are related to the implemented object continue to be configured.

- (10) The values of  $change_{ij}$  are defined as follows:

- ① If  $change_{ij} = 1$ , the configured pictures should be timely updated to keep pace with the current plant running status.
- ② If  $change_{ij} = 0$ , pictures are not updated.

## Picture configuration scheduling algorithm

### Scheduling process

The scheduling process, which is illustrated in Figure 1, mainly includes determining the priorities of each relevant picture and dynamically arranging the pictures and tasks in a buffer pool. These steps are described in the following.

*Calculating the priority of each picture.* According to the constraint conditions above, equation (7) can be used to calculate the priority of each picture. Equation (7) consists of three parts: (1) the similarity degree between the implemented objects of the primary task and the pictures; (2) the visitation frequency of the pictures; and

(3) the importance degrees of the pictures. The visitation frequency of the pictures can be calculated via equation (2). The importance degree of the pictures can be obtained via operator interviews and expert judgments. The similarity degree can be obtained via equation (6). For equation (6), two steps must be conducted: (1) extracting the picture information keywords that are associated with the implemented objects of the current primary task from a feature library and (2) calculating the number of identical keywords. A feature library is established and improved by domain experts, supervisors, and advanced operators. Extraction of the keywords and calculation of the number of identical keywords can be conducted by following two algorithms, which are presented as follows

- 1) Algorithm for extracting keywords from a feature library

- (1) Algorithm process

The algorithm steps are as follows:

- ① Successively search for the current primary tasks in a feature library.
- ② If the current primary tasks that are being implemented are identified, their keywords will be extracted; otherwise, return to step (1).
- ③ Add the  $i$ th primary task keywords into a vector space ( $k\_w\_f_i$ ).
- ④ Successively search for the current  $j$ th picture keywords from the  $i$ th primary task keyword vector space ( $k\_w\_f_i$ ).
- ⑤ If the current  $j$ th picture is identified, the  $p$ th keyword of the  $j$ th picture will be extracted; otherwise, return to step (4).
- ⑥ Add the  $p$ th keyword into a vector space ( $k\_w\_s_{ij}$ ).

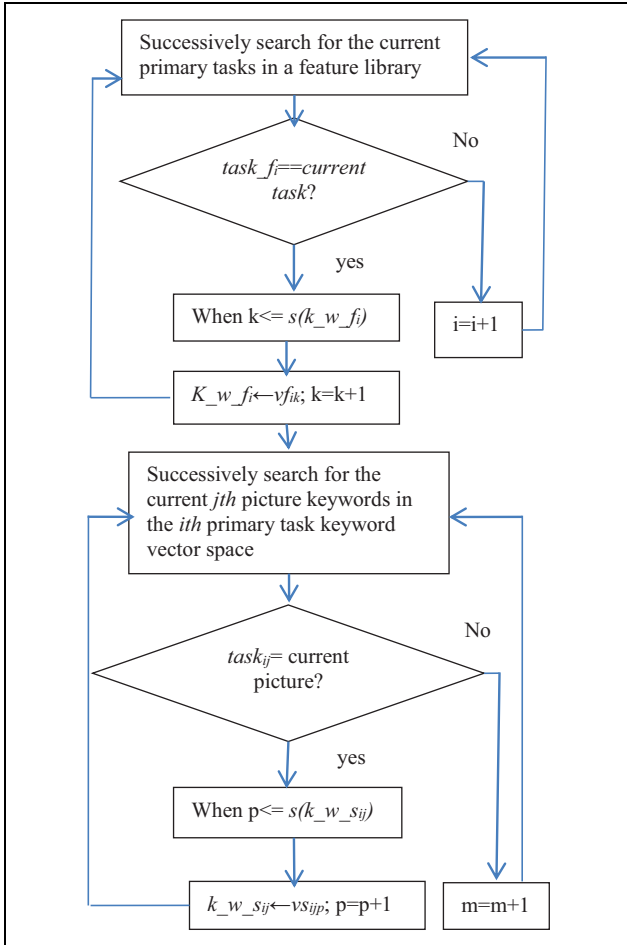
This algorithm process for extracting keywords is illustrated in Figure 2.

- (2) Pseudo code for extracting keywords from a feature library

```

Feature_extract_algorithm()
Begin
i=1;
While(i<=c_sum) begin If(task_fi=cur_task)
While(k<= s(k_w_fi)) begin K_w_fi←vfik;
k=k+1; end;
Else i=i+1; End;m=1;p=1;
While(m<=c_sum) Begin If(task_ij=cur_f)
While(p<= s(k_w_s_ij)) Begin k_w_s_ij←vs_ijp;
p=p+1; End;
Else m=m+1; End;
END.

```



**Figure 2.** Algorithm process for extracting keywords.

2) Algorithm for calculating the number of identical keywords

(1) Algorithm steps

- ① Find the keyword vector space of the implemented objects from the  $i$ th primary task.
- ② Find the keyword vector space of the  $j$ th picture that is related to the implemented object of the  $i$ th primary task.
- ③ Successively search for the  $p$ th keyword of the  $j$ th picture from the current  $i$ th primary task.
- ④ If the current task picture keyword is identified, the count is successively increased.

The algorithm process is illustrated in Figure 3.

(2) Pseudo code for calculating the number of identical keywords

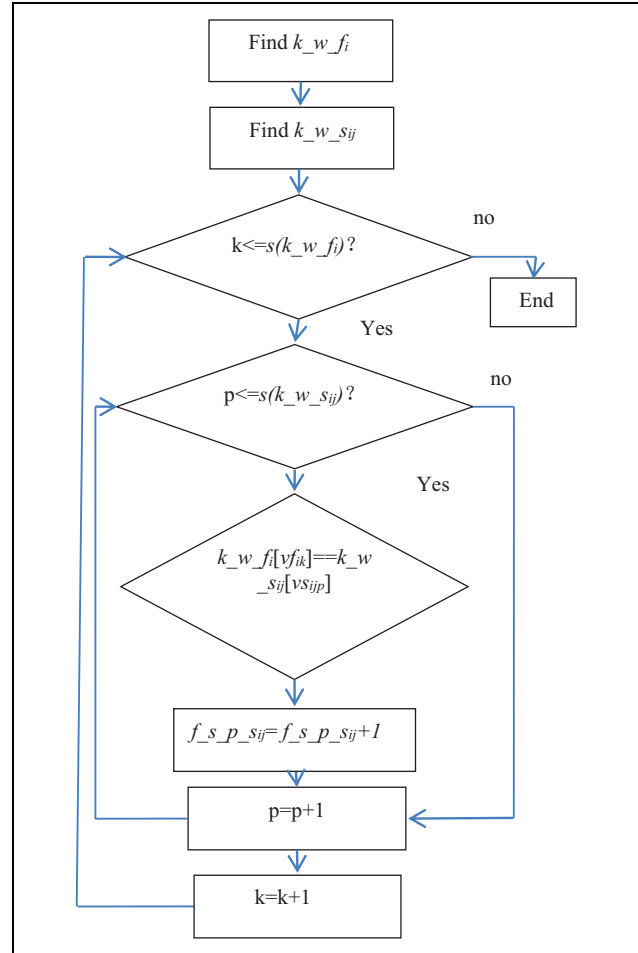
Calculate\_key\_sum()

Begin

$k=1; p=1;$

Locate( $k_w_fi$ );

Locate( $k_w_sij$ );



**Figure 3.** Algorithm process for calculating the number of identical keywords.

```

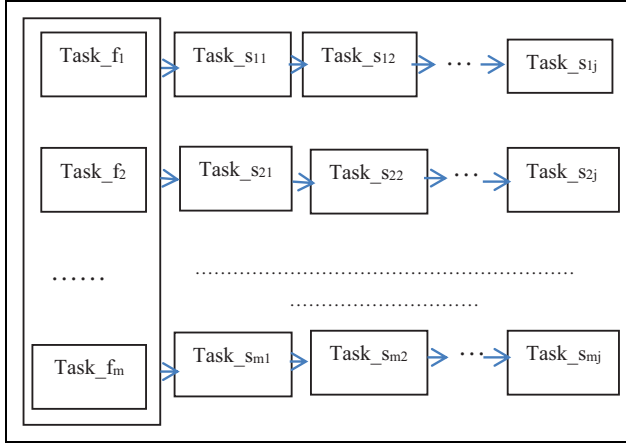
While( $k \leq s(k_w_fi)$ )
  begin
    while( $p \leq s(k_w_sij)$ )
      begin if( $k_w_fi[vfik] = k_w_sij[vsijp]$ )
         $f_s_p_sij = f_s_p_sij + 1; p = p + 1;$ 
      end;
       $k = k + 1;$ 
    end;
  end;
End.

```

*Dynamically establishing the sequences of pictures and primary tasks in multilevel queues of a buffer pool.* Two problems must be solved for picture configuration and primary tasks in a buffer pool: (1) arranging them in order and (2) dealing with the dynamic process when the latest pictures and tasks arrive to the buffer pool. The solutions of the two problems are described in the following sections.

- (1) Arranging the pictures and primary tasks in a buffer pool

The proposed process for arranging the pictures and primary tasks is as follows: ① based on corresponding



**Figure 4.** Queue construction of implemented tasks and pictures.

accidents, the implemented objects of primary tasks for which the weights are greater than or equal to the threshold values are added into the buffer pool, and objects that were implemented earlier are arranged with higher priority in the queue; ② all pictures that are related to the implemented objects of the primary tasks are searched; ③ all relevant pictures are arranged in order of their priorities to build a navigation path; and ④ if the sum of the sizes of all tasks is greater than the buffer pool size, then the pictures that are arranged behind other pictures in the same queue will be removed from the buffer pool. The multilevel queues structure of primary tasks and relevant pictures is illustrated in Figure 4.

- (2) Dealing with the dynamic process when the latest pictures and tasks arrive to a buffer pool

If the sum of the sizes of all implemented objects of primary tasks and pictures in the buffer pool is greater than or equal to the buffer pool size, a few implemented objects and relevant pictures in the buffer pool will be replaced by other objects or related pictures. The replacement process is realized via an algorithm, which has the following algorithm process: ① before the latest pictures and tasks are added into the buffer pool, the sums of the sizes of the buffer pool and tasks, respectively, must be calculated; ② if equation (8) holds, the pictures or tasks will be directly added to the end of a queue of the buffer pool, where the queue structure is illustrated in Figure (4); ③ if equation (8) does not hold, before new pictures or tasks are added into the queues in order, a few pictures or tasks must be removed from the queues, namely the pictures or tasks that have been in the queues for the longest will be replaced by pictures or tasks that should be implemented as early as possible.

The pseudo code of the replacement algorithm is as follows:

```

Rep_task_algorithm()
Begin
  If Eq. (8) then
    Those pictures or tasks are added into the queues;
  else
    Begin
       $K\_time\_long\_task = task\_f_j$ ;
      For  $i = 2$  to  $m$  do
        If ( $v\_time(task\_f_i) > K\_time\_long\_task$ ) then
          Begin
             $K\_time\_long\_task = task\_f_i$ ;
             $v = i$ ;  $i = i + 1$ ;
          end;
         $i = v$ ;
         $task\_f_i \leftrightarrow cur\_task$ ;
         $task\_f_{ij} \leftrightarrow cur\_sec\_task$ ;
        order( $cur\_sec\_task$ );
      end
    End
  End

```

### Picture configuration scheduling algorithm

The process of the picture configuration scheduling algorithm is illustrated in Figure 1.

According to Figure 1, the definitions of the constraint conditions and Scheduling process section, the picture configuration scheduling algorithm of the digital HCI in an Npp is defined as follows:

```

Scheduling_Algorithm_picture_configuration()
Begin
  Initialize
     $W\_f \leftarrow$  an initial value;
     $pri\_f \leftarrow$  an initial value;
  repeat
    For  $i = 1$  to the total quantity of objects to be executed do
      Begin
         $mt_i \leftarrow$  specify a value;
         $fg_{ij} \leftarrow$  according to Eq. (3);
         $fp_i \leftarrow$  according to Eq. (4);
        if (Eq. (5)) then
          add  $task\_f_i$  into a buffer;
        for  $j = 1$  to  $N_{ij}$  do
          begin
            call Feature_extract_algorithm(), which was proposed in this article;
            call Calculate_key_sum(), which was proposed in this article;
             $sim(k\_w\_f_i, k\_w\_s_{ij}) \leftarrow$  according to Eq. (6);
             $pri\_w_{ij} \leftarrow$  according to Eq. (7);
            if (Eq. (1)) then
              continue;
            else
              break;
            end if
          add  $task\_s_{ij}$  into the buffer to form a navigation path;

```

```

    end if; end;
end;
until(Eq. (8) is false)
(2) Function pseudo codes of the running process
Check the current plant status and regulations
For i=1 to m do
    begin
        If(cur_task=task_fi) then
        For j = 1 to Nij do
            Begin
                If(cur_f=task_sij) then
                    if(pri_wij<pri_f) then
                        delete(buffer, task_fij);
                    else
                        re_order(buffer, task_sij);
                    end
                else
                    break;
            End;
        End for
    End for
    End for
    If not exist(buffer, cur_task) then
        If (Eq. (8)) then
            Call Rep_task_algorithm(), which was
            proposed in this article;
        End if
        L1: call Feature_extract_algorithm(), which was pro-
        posed in this article;
        call Calculate_key_sum(), which was proposed in this
        article;
         $sim(k_w_{fi}, k_w_{sij}) \leftarrow$  according to Eq. (6);
         $pri_{wij} \leftarrow$  according to Eq. (7);
    End L1;
    If(pri_wij>pri_f) then
        Add cur_f into buffer;
        re_order(buffer, task_sij);
    else
        goto L1;
    End if;
    For i = 1 to m do
        Begin
            For j=1 to Nij do
                Begin
                    If changeij=1 then
                        Update(task_sij);
                        Mapping(plant_data←task_sij);
                    End if;
                End;
            End;
        End;
    End
End

```

## Performance analysis

### Experimental background

To evaluate the performance of the picture configuration scheduling algorithm, related experiments are conducted

**Table 1.** Task points.

Number	Task description
1	Confirm: Confirm RCV 017VP on RCV 0002BA (BY-pass demineralizers RCV)
2	Confirm REA on AUTO makeup the Boron concentration of the primary system
3	The volume of REA Boron tanks
4	Set RCP 404KU X the value of no load Set point (20% of -4 m)
5	Set RCV 046VP on AUTO
6	Reset CIB signal by RPA 284KG and RPB 284KG
7	Reset SI signal by RPA 060KG and RPB 060KG
8	Confirm the reactor trip by RPA 300TO and RPB 300TO
9	Check that all the CIA values are close
10	Confirm that RIS 061VP and 062VP are open

**Table 2.** Picture numbers.

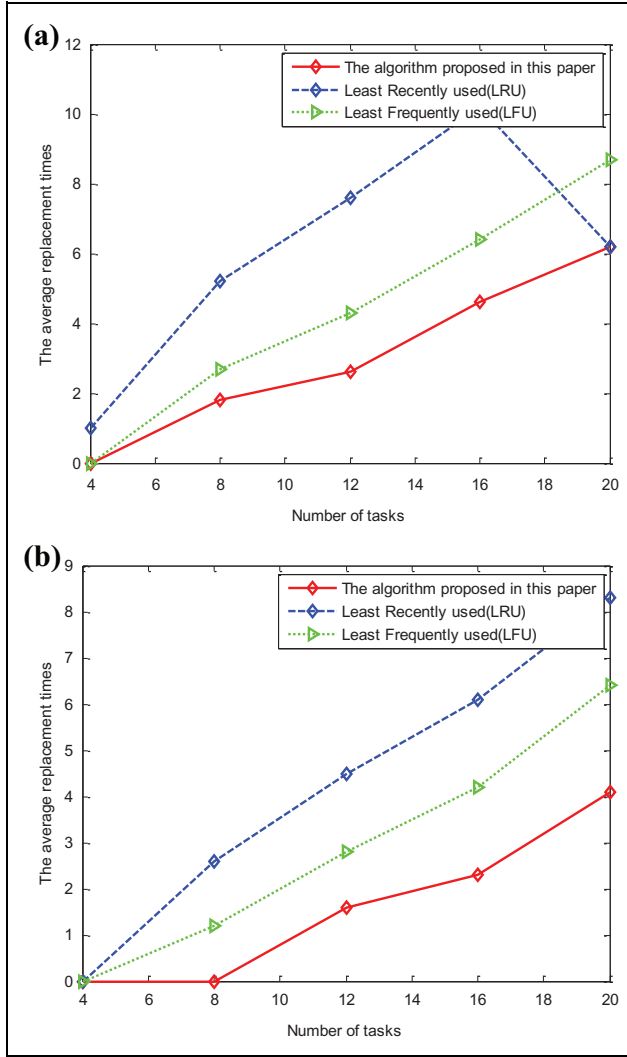
Picture number	
1	RIC003YCD
2	RCV002YCD
3	REA001YCD
4	ECP002YCD
5	TEP003TCD
6	RCV001YED
7	RCP002YCD
8	EPP002YFU
9	RIS100YFU
10	EAS100YFU
11	RGL001YCD
12	EPP001YFU
13	LHP001YCD
14	LHQ001YCD
15	DOS10AYST

by the authors. A steam generator tube rupture (SGTR) accident in an Npp is used for illustration. As task points are more in SGTR accident, 10 task points were selected for the convenience and standard of experimental procedures, experimental participants mainly deal with these task points and the relevant pictures are obtained from DOS regulations of SGTR accidents. The task points are listed in Table 1. Each picture is represented by a number, as presented in Table 2.

### Experiment description

Participants in the experiment must obtain parameters, evaluate the plant status, decide to how to deal with or restore an accident site, and access branches of accident regulations. To compare the time performance including configuring pictures and manual approach, picture configurations are scheduled via the proposed algorithm and participants in the experiment, respectively. Ten students from Hunan Institute of Technology participated



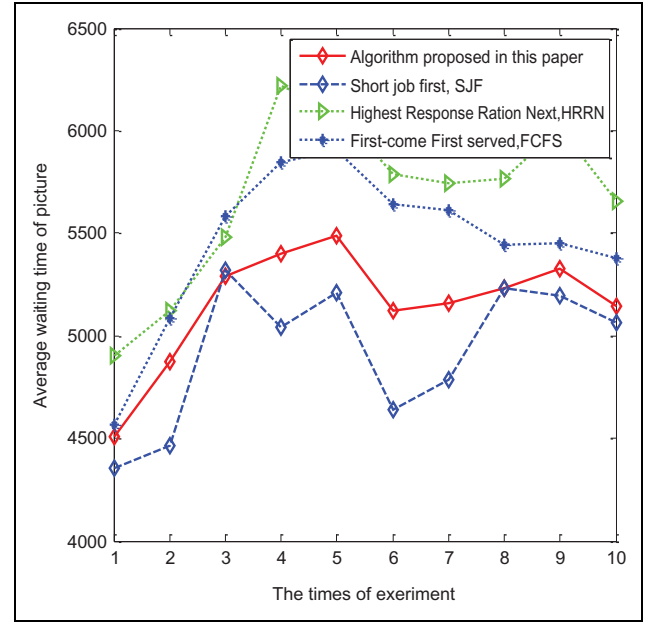


**Figure 5.** Change curves of the numbers of replacements: (a) replacements when the buffer pool size is 3 and (b) replacements when the buffer pool size is 5.

in the simulative experiment; they were divided into five groups and were trained for 2 days. The experiment was conducted 10 times. Each group is required to do two trials.

In the experiment, some parameters have dynamic values, such as  $N_{ij}$ ,  $U_{sum_{ij}}$ ,  $S_{f_{sum}_i}$ ,  $Ft_i$ ,  $Fg_{ij}$ ,  $mt_i$ , and  $gm_{ij}$ . The dynamic values may be obtained during the simulative experiment according to related tasks. The initial values of a few parameters must be specified directly. Two initial values are set as  $w_f = 0.5$  and  $pri_f = 0.5$ . Most parameter values are obtained or dynamically changed according to the running process of the SGTR accident. The experimental process is based on Figure 1.

The simulation platform that is used for the experiment is Windows 7, with an i7-6700 CPU, 8 G RAM, and disk space of 500 GB. The experimental results are the mean values of all experimental data.



**Figure 6.** Picture average waiting time (ms).

### Performance analysis

The performance of the picture configuration scheduling algorithm is analyzed from several perspectives according to the experimental data.

- (1) The change curves of the numbers of replacements, which are plotted in Figure 5.

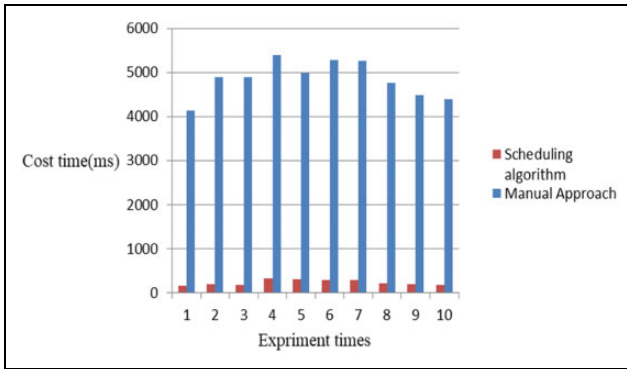
Replacement is viewed as a process, namely ① lower correlation pictures with current task are removed from buffer pool; ② more correlation pictures with current running task will get into the buffer pool. According to Figure 5, the numbers of replacements of (a) and (b) are 0 when the number of tasks is 4, which is the optimal case. Fewer replacements correspond to less time being spent on picture configuration.

Comparing with the least recently used and least frequently used methods, the algorithm proposed in this article conducts few replacements, which indicates the algorithm proposed has good performance on replacements. It is shown in Figure 5 that the number of replacements will increase with the number of tasks, which accords with the actual scenario, as the size of a buffer pool is fixed and the probabilities that relevant pictures are not in the buffer pool increase with the number of tasks.

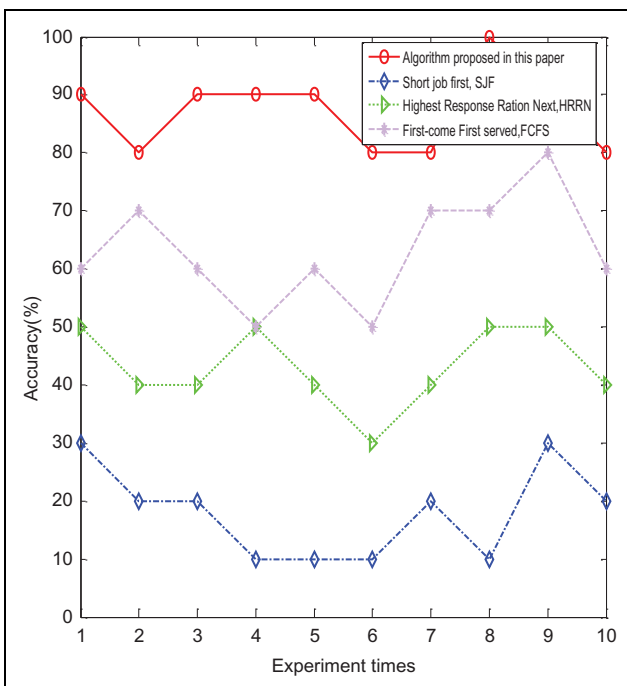
- (2) Picture average waiting time, which is plotted in Figure 6.

Waiting time is viewed as an interval, namely, it is after picture is get into the buffer pool, until is automatically configured on a screen.

According to Figure 6, the picture average waiting time in the experiments with the algorithm that is proposed in



**Figure 7.** Picture configuration times of the manual approach and the scheduling algorithm.



**Figure 8.** Accuracy of the picture configuration scheduling algorithm.

this article is approximately 5200 ms. Comparing with the highest response ratio next (HRRN) and first come first served (FCFS) methods, the algorithm that is proposed in this article has a shorter waiting time; however, comparing with the shortest job first (SJF) method, it has a longer waiting time. Shorter waiting time means that time cost of picture configure is less. By and large, the algorithm performance on average waiting time is good.

### (3) Time cost analysis

According to Figure 7, the time cost of the scheduling algorithm is far less than the time cost of the manual approach for picture configuration; hence, the scheduling algorithm outperforms the manual approach. If time cost of

picture configuration is decreased, then psychology pressure of operators is decreased, and then accident safety can be improved.

### (4) Accuracy of picture configuration, which is plotted in Figure 8.

According to Figure 8, the accuracy of the picture configuration scheduling algorithm proposed in this article is approximately 85%; hence, it is reliable. Comparing with the SJF, HRRN, and FCFS methods, the algorithm that is proposed in this article is more accurate.

## Conclusions

This article discusses how secondary tasks in a digital HCI increase the mental loads of operators and analyzes the advantages that pictures were intelligently configured by robot technology. In this article, based on robot technology, a picture configuration scheduling algorithm of secondary tasks is obtained. All relevant variables of the scheduling algorithm are defined. Mathematical expressions for several constraint conditions are established. In addition, several algorithms for extracting information features, counting identical keywords, and configuring pictures of secondary tasks were proposed. The simulative experiment analysis results demonstrate that the picture configuration scheduling algorithm realizes satisfactory performance.

Most of the data obtained via the simulation experiments reflect the algorithms' performances for picture configuration, such as correctness, number of replacements, and waiting time. However, the participants are students; hence, the time that is spent on configuring pictures manually might exhibit small deviations. However, the deviations have little effect on the performance of the scheduling algorithm, as the time difference between the manual approach and the scheduling algorithm is very large. Thus, the small deviations have no readily observable effects on the difference in time cost between the manual approach and the scheduling algorithm. In the future, the constraint conditions will be further improved according to feedbacks in application process; the algorithm will be extended to other fields.


## Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported in part by Hunan Provincial Natural Science Foundation of China (2019JJ40066, 2017JJ4019), the Social and Science Fund of Hunan Province of China (XSP18YBZ035), the Scientific Research Foundation of Hunan Institute of Technology of China (HQ19009), The key laboratory of Hunan Province (2019TP1020), China.

**ORCID iD**

Jianjun Jiang  <https://orcid.org/0000-0001-5856-2055>

**References**

1. Li Z, Da-xin Y, and Yi-qun W. The effect of information display on human reliability in a digital control room [in Chinese]. *China Saf Sci J* 2010; 20(9): 81–84.
2. Linfeng L. *A monitoring transferring law research for operators in a digital nuclear power plant* [in Chinese]. Master Thesis, University of South China, 2013.
3. Wickens C. *Processing resources and attention. Multiple task performance*. London: Taylor & Francis Press, 1991.
4. Seong PH. *Reliability and risk issues in large scale safety-critical digital control systems*. New York: Springer Press, 2009.
5. Yan R. The Research of human-computer interface design based on vision communication. *Procedia Eng* 2011; 15: 3114–3118.
6. Anuar N and Kim J. A direct methodology to establish design requirements for human–system interface (HSI) of automatic systems in nuclear power plants. *Ann Nucl Energy* 2014; 63: 326–338.
7. Bhatti G, Bremond R, Jessel JP, et al. Design and evaluation of a user-centered interface to model scenario on driving simulators [Special Issue on Road Safety and Simulation]. *Transport Res C Emer Technol* 2015; 50: 3–12.
8. Futrell BJ, Ozelkan EC, and Brentrup D. Optimizing complex building design for annual daylighting performance and evaluation of optimization algorithms. *Energ Buildings* 2015; 92(1): 234–245.
9. Haynes S. Effects of positioning optimization in an alternative computer workstation for people with and without low back pain. *Int J Ind Ergonom* 2009; 39(5): 719–727.
10. Xia L, Zhu J, Zhang W, et al. An implicit model for the integrated optimization of component layout and structure topology. *Comput Method Appl Mech Eng* 2013; 257: 87–102.
11. Sørensen SE, Hansen MR, Ebbesen MK, et al. Non-linear optimization of track layouts in loop-sorting-systems. *Automat Constr* 2013; 31: 19–30.
12. Kim S, Nussbaum MA, and Gabbard JL. Influences of augmented reality head-worn display type and user interface design on performance and usability in simulated warehouse order picking. *Appl Ergon* 2019; 74: 186–193.
13. Lindscheid C, Sakthithasan P, and Engell S. An ecological interface design based visualization of the energy balance of chemical reactors. *IFAC Papers OnLine* 2019; 51: 308–314.
14. Gaouar L, Benamar A, Le Goar O, et al. HCIDL: human–computer interface description language for multi-target, multimodal, plastic user interfaces. *Future Comput Inform J* 2018; 3: 110–130.
15. Ding LR, Lin RZ, and Lin ZY. Service robot system with integration of wearable Myo armband for specialized hand gesture human–computer interfaces for people with disabilities with mobility problems. *Couput Electr Eng* 2018; 69(7): 815–827.
16. Cheng FF, Wu CS, and Leiner B. The influence of user interface design on consumer perceptions: a cross-cultural comparison. *Comput Hum Behav* 2019; 101: 394–401.
17. Kantowitz B, Hanowski R, and Tijeina L. Simulator evaluation of heavy-vehicle workload:  $\Pi$ : complex secondary tasks. In: *1996 Proceedings of the Human Factors Society-40th Annual Meeting*, Santa Monca, CA, USA, August 1996, pp. 1002–1006. Human Factors Society.
18. Tijerina L, Kigr S, Rockweel T, et al. Workload assessment of in-cab test message system and cellular phone use by heavy vehicle drivers the road. In: *1995 proceeding of the Human Factors Society-39th Annual Meeting*, Santa Moncia, CA, USA, September 1995, pp. 1015–1019. Washington, DC: Human Factors Society.
19. Howard J and Kerst S. Memory and perception of cartographic information for familiar and unfamiliar environments. *Hum Factors* 1981; 23: 495–504.
20. Cook R and Woods D. Adapting to new technology in the operating room. *Hum Factors* 1995; 38: 593–613.
21. TiborPetzoldt SB and Krems JK. Learning effects in the lane change task (LCT) – realistic secondary tasks and transfer of learning. *Appl Ergon* 2014; 45(3): 639–646.
22. Nibbeling N, Oudejans RRD, and Daanen HAM. Effects of anxiety, a cognitive secondary task, and expertise on gaze behavior and performance in a far aiming task. *Psychol Sport Exerc* 2012; 13(4): 427–435.
23. Robinson MM and Irwin DE. Conscious error perception: the impact of response interference from a secondary task. *Atten Percept Psychophys* 2017; 79: 863–877.
24. Avola D, Spezialetti M, and Placidi G. Design of an efficient framework for fast prototyping of customized human–computer interfaces and virtual environments for rehabilitation. *Comput Meth Prog Biomed* 2013; 110(3): 490–502.
25. Troiano L and Birtolo C. Genetic algorithms supporting generative design of user interfaces: examples. *Inform Sci* 2014; 259(20): 433–451.
26. Ramakrisnan P, Jaafar A, Hanis F, et al. Evaluation of user interface design for leaning management system (LMS): investigating student’s eye tracking pattern and experiences. *Procedia Soc Behav Sci* 2012; 67(10): 527–537.
27. Browne K and Anand C. An empirical evaluation of user interface for mobile video game. *Entertain Comput* 2012; 3(1): 1–10.
28. Chun-yan X, Sheng-yuan Y, Qing-fen L, et al. Experimental study on human–machine interface evaluation of main control room in nuclear power plants. *Chin Saf Sci J* 2008; 18(8): 109–114 (In Chinese).
29. Wei Z and Wei H. Method of software interface evaluation based on eyetracking technology [in Chinese]. *Electro Mech Eng* 2013; 29(4): 62–64.
30. Naujoks F, Purucker C, and Neukum A. Secondary task engagement and vehicle automation – comparing the effects of different automation levels in an on-road experiment. *Transport Res F Traf Psychol Behav* 2016; 38(3): 67–82.
31. Ye M, Osman OA, Ishak S, et al. Detection of driver engagement in secondary tasks from observed naturalistic driving behavior. *Accid Anal Prev* 2017; 106(9): 385–391.