Contemporary Engineering Sciences, Vol. 8, 2015, no. 17, 789 - 797
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2015.57196

Development of a User Interface for an Integrated System of Video Monitoring Based on Ontologies

Daniil A. Loktev

Bauman Moscow State Technical University (BMSTU)
5, 2-aya Baumanskaya st.,105005, Moscow, Russian Federation

Alexey A. Loktev

Moscow Technical University of Communications and Informatics (MTUCI)
8a, Aviamotornaya st., 111024, Moscow, Russian Federation

Abstract

We consider the problem of designing and implementing the user interface for control of a multicomponent system of video monitoring, which includes modules of producing high-quality images, database management system, control system of external illumination, the system of visual pattern recognition, system interfaces for developers, users and interacting programs, the system of creation of network templates, which can be scaled up depending on the network structure, the system of adaptation of the developed complex for the needs of the organizations and objectives of the various subject areas. The developed software modules take into account possible influence on the user at different stages of the monitoring operation and the ability to use the enterprise security system within a larger security system. Moreover, the implemented interface can be used in various industries, such as instrumentation, mechanical and biomedical engineering.

Keywords: adaptive user interface, the ontological approach, graphical interface, window container, the assessment of the system's ergonomic, system of the video monitoring

1 Introduction

Here we explore the possibility to develop and realize a user interface, which would assure both the connection between different modules of the complex system and interaction between the system and users [1]. For that several requirements should be met, such as flexibility, modifiability, mobility, multiple using of the elements of screen images or schemes. The intelligent and comprehensive system of the video monitoring should consist of subsystems, each of which performs certain tasks and contributes to solving tasks by other parts of the complex. To combine different applications, procedures and subprograms in a single complex it is necessary to develop a modern multi-directional system of software and user interfaces. Furthermore, the resulting algorithms are not only a means of solving the problems [2] but also independent objects of study, which are determined by the computational complexity, optimality [3], difficulties of the software implementation on the inter-system platforms.

2 Statement of the Problem

An access control system includes readers, controlled blocking devices, software, controllers, etc. One of the main elements of such a system is the procedures and identification algorithms, each of which has its advantages and disadvantages. Environmental conditions, presence of glasses, contact lenses and background noise can affect the object recognition device, which can lead to an incorrect reading of the device information. Therefore, the most interesting is the identification and tracking of the object throughout its route. The technology of face recognition works well with standard video-cameras, which transmit data and are controlled by a personal computer.

To control modules in the unified system of video monitoring it is advisable to use a certain type of graphical user interface that is WIMP-interface («window, icon, menu, pointing device»).

In general, any tool for creating of a graphical interface has to solve two main tasks, i.e., to reduce the development cycle (to exhibit fast performance) and to protect investments in the project (to minimize possible processing) [4]. It is necessary to take into account characteristics of users as the criteria of the interface adaptation [5,6]. For maximum performance, the user interface should be convenient and ergonomic [7]. The ergonomics of the interface is defined as the ability to perform more actions by the user at the same device. To create a user interface one needs to think through dialogues and user tasks, WIMP-interface, the formation of program texts, applied graphic scenes, and the connection of the developed interface with the application.

3 Solution of the Problem

3.1 Ontological Approach

The approach using ontology for the presentation of the interface concepts used in the design allows for more detailed description of the user interface for its further transfer to the software code than other methodologies. In the application of the ontological approach to automation of the professional activity it is necessary to allocate a system of concepts, used by experts of the working group in the design of the user interface.

The structure of the system of the concepts of WIMP-interfaces consists of a set of interface elements [8]. Each element of the interface is defined by its type, a variety of parameters, events and functions. The basic elements of the interface, designed to group items into related groups and presented in the form of rectangular areas on the screen are interface elements, i.e. «window container». Interface elements designed for Input/output data and for calling the commands belong to the type of «element of control». Type «support element» is used to describe the elements of the interface of two previous types. Each element of the interface is connected to the set of events defined by a name and a set of parameters. Events determine the reaction of an interface element and its interaction with a user. To describe the set of possible actions different functions are used. Each function is defined by a name, a set of parameters and a return type.

Any dialogue of a user with the software tool is in accordance with a script which depends on the type of users, their requirements, functions of the application software and is a component of the user interface. System of concepts associated with the script of the dialogue defines a set of possible states of the dialogue and the actions, which are performed in each state. The state of dialogue is defined by an event, which occurs in the interface element. Actions determine the sequence of instructions that are executed in response to an event.

When using the ontological approach the systems of concepts in terms of which the user interface design is implemented are the following:

1) The ontology model of a system of the dialogue concepts describes the term structure of the dialogue concepts in the link form.

2) The ontology model of user tasks describes the problems that can be solved with the help of software.

3) The ontology model of WIMP-interfaces consists of two parts: the interface elements of type "window container" are designed to group the elements of the interface into semantically related groups; the events of the interface elements $Events_i=\{Event_{ij}\}$ give a set of possible responses of the interface element to the events occurring in the course of interaction with the user.

4) The ontology model for the text formation allows for the construction and use of text fragments in the user interface.

5) The ontology model of graphical static and dynamic scenes.

6) The ontology model of a dialogue script describes the initial window, which appears on the screen after starting the software, the structure of responses to events, the types of designs and the types of instructions that are available in the description of reactions to events, the standard functions of the dialogue.

7) The ontology model of the interface connection with the application describes the structure of program interfaces *Interfaces* used to communicate between interface and application software.

3.2 Development of the interface

Based on the proposed techniques the following components of the interface have been developed and implemented:

1. The protected area is divided into sectors and the building is a separate sector.

2. The building is represented in 3D format. The cameras are located on the floors. You can select the desired floor by typing the number in the command line.

3. Each floor displays number of sections and number of cameras. It is also necessary to show the level of access to each room.

4. There is organization of images from the camera in the rooms and on the routes.

5. When receiving images from a single camera recognized and unrecognized objects are allocated differently. The parameters of the object location are displayed (time, place).

6. When there is a picture captured with a single camera or multiple cameras, it works in the recognition mode.

7. For images from a single camera it is possible to get the access to the database.

8. In the most of the windows and modules it is possible to recognize the objects both manually and automatically.

9. There is a log of all accidents, all movements of the recognized (or marked) objects and all actions of the operator.

10. If the object is not recognized, the frames are saved and the object is marked and passed from camera to camera.

The first window is shown of the entire protected area, which includes the building and its surrounding area that is divided into numbered sections. The building refers to an individual segment. It is possible to switch to another window by pressing the button «Go to another window», by inputting the number of the desired window into the command line and by pressing the «Enter», or using the “Back” or “Forward”.

The second window is a view of the building in the normal mode, in the simulation mode (See Fig. 1) or in the transparent mode.

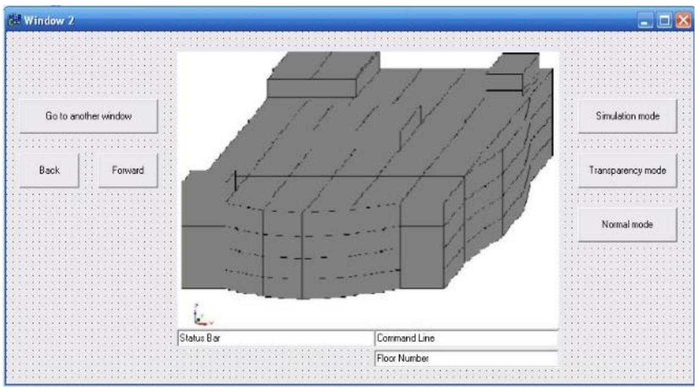

Fig. 1. Window Number 2

The third window shows the plan of the previously selected floor of the building with installed cameras. It is possible to select a video from a single camera or from multiple cameras in the rooms or on the routes.

In the fourth window, a set of cameras in the previously selected room or on the previously selected route are shown. After the transition to a single camera view, the Viola-Jones recognition method using the Haar primitives is applied.

Then four scenarios are possible:

1) A person is recognized as a known object. It is possible to choose any person from the known objects by number (if more than one), to view data about the person and manage his/her access to rooms.

2) A person is recognized as an unknown object. It is possible to choose any person from the unknown objects by number (if more than one), to change the lighting in the room to try to recognize it as a known object or add to the database.

3) There are objects both included and not included in the database. It is possible to choose any person by number, to change the lighting in the room, to view data about the person recognized as a known object (See Fig. 2).

4) The recognition does not occure. In this case it is possible to change the lighting and try to recognize again or see the neighboring cameras.

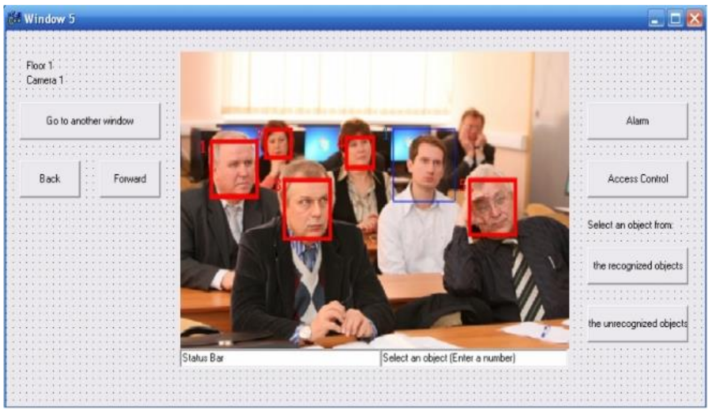

Fig. 2. Window Number 5 – recognized and unrecognized persons together

3.3 Description of the interface using the ontological approach

Describing the implemented interface using the ontological approach it can be noted that the set *Controls* consists of the following subsets [9]: *Controls* = *Base* ∪ *Window controls* ∪ *Windows.*

Base contains a set of supporting elements of the interface. *Base* = *{WIMP-element, Font settings}.*

Window controls contains a set of interface elements, describing the elements of control: *Window controls = {Control button, Input field, Static text, Status bar, Button-down list box}.*

The subset *Windows* contains a set of interface elements, describing the windows. *Windows = {Standard window}.*

WIMP-element = <*Controltype*$_{WIMP\text{-}element}$, *Parameters*$_{WIMP\text{-}element}$, *Events*$_{WIMP\text{-}element}$, *Functions*$_{WIMP\text{-}element}$> – an interface element that describes the common windows and controls properties.

Font settings = <Controltype$_{Font\ settings}$, Parameters$_{Font\ settings}$, Events$_{Font\ settings}$, Functions$_{Font\ settings}$> – an interface element that describes the font settings.

Status bar = <Controltype$_{Status\ bar}$, Parameters$_{Status\ bar}$, Events$_{Status\ bar}$,

Functions$_{\textit{Status bar}}$> – an interface element that describes the controls for displaying information about the current status of the objects.

Button element = <Controltype$_{\textit{Button element}}$, Parameters$_{\textit{Button element}}$, Events$_{\textit{Button element}}$, Functions$_{\textit{Button element}}$> - an interface element that describes the properties that are common for control elements that are used to initiate any action or to change the properties of the objects.

Control button = <Controltype$_{\textit{Control button}}$, Parameters$_{\textit{Control button}}$, Events$_{\textit{Control button}}$, Functions$_{\textit{Control button}}$> - an interface element that describes the properties similar for all buttons designed to run commands or operations.

Button-down list box — <Controltype$_{\textit{Button-down list box}}$, Parameters$_{\textit{Button-down list box}}$, Events$_{\textit{Button-down list box}}$, Functions$_{\textit{Button-down list box}}$> - an interface element that describes the properties similar for all buttons designed to choose variants from a certain set.

Input field = < Controltype$_{\textit{Input field}}$, Parameters$_{\textit{Input field}}$, Events$_{\textit{Input field}}$, Functions$_{\textit{Input field}}$> – an interface element that describes the properties of the area in which the user can enter and edit text.

Static text = <Controltype$_{\textit{Static text}}$, Parameters$_{\textit{Static text}}$, Events$_{\textit{Static text}}$, Functions$_{\textit{Static text}}$> – an interface element that describes the properties of the text used for display only.

Standard Window = <Controltype$_{\textit{Standard Window}}$, Parameters$_{\textit{Standard Window}}$, Events$_{\textit{Standard Window}}$, Functions$_{\textit{Standard Window}}$> – an interface element that describes the properties of the screen, using which is able to get a visual representation of a particular aspect of the problem being solved.

3.4 Calculation of the performance of the interface

The GOMS method [10] allows for estimation of the execution time of a task by the user and for calculation of the performance of the interface application. For the typical user actions the time of their performance can be measured and the static estimates of an elementary action can be obtained. The performance evaluation of the interface includes decomposing into typical components and the calculation of the average time that will be spent by the user to perform this task [11].

M=1.35 s – the time required by the user to prepare mentally for the next step (mental preparation). H=0.4 s – the time needed by the user to move his hands from the keyboard to the manipulator and from the manipulator to the keyboard. P_i – the time required by the user to specify any position on the screen of the monitor. (i depends on the location of the fields and buttons on the form). K= K_s=0.28 s – the time required to press the keyboard keys (K) or the manipulator button (K_s). It depends on the speed of the user's pressing on the buttons and is 0.28 s for the average unexperienced user.

Let us consider an example where a person entering the building and already included in the database should be provided an access to the door #1 of the room #X on the floor #N. His actions in each window can be described as follows:

1) In the first window, the performance is obtained from the following:

$$MH\ MP_1K_s\ MH\ MK\ MK, \qquad (1)$$

where MH – the movement of the hand to the manipulator; MP_1K_s – the cursor movement to the field to enter the desired sector followed by its choice; MH – the movement of the hand from the manipulator to the keyboard; MK – the set of a single value; MK – “Enter”.

2) In the second window: MH MP_2K_s MH MK MK, (2)
where MP_2K_s – the cursor movement to the field to enter the desired sector following by its choice.

3) In the third window: MH MP_3K_s MP_4K_s MH MK MK, (3)
where MP_3K_s – the cursor movement to the field to enter the desired sector and its choice; MP_4K_s – the cursor movement to the field to enter the desired sector and its choice.

4) In the fourth window: MH MP_5K_s MH MK MK, (4)
where MP_5K_s – the cursor movement to the field to enter the desired sector and its choice.

5) In the fifth window:
MH MP_6K_s MP_7K_s MH MK MK MH MP_8K_s MP_9K_s $MP_{10}K_s$, (5)
where MP_6K_s – the cursor movement to the field to enter the desired sector and its choice; MP_7K_s – the cursor movement to the field to enter the desired sector and its choice; MP_8K_s – the cursor movement to the button “Access control” to enter the desired sector and its choice; MP_9K_s – the cursor movement to the button “Allow” to enter the desired sector and its choice; $MP_{10}K_s$ – the cursor movement to the button “Door #1” to enter the desired sector and its choice.

Using the Fitts' law [3] we can calculate the P_i value:

P_i (ms) = a + b*$\log_2$(D/S+1), (6)

where for the one-dimensional example S – the size of the object along the line movement of the cursor, D – the distance from the starting position of the cursor to the object.

For two-dimensional examples, the parameter S is the smallest dimensional value of the object either horizontally or vertically. Constants a and b are the empirical set of the performance parameters of the person. For approximate calculations, we use the following values of the constants in the equation of the Fitts' law (6): a=50, b=150. We obtain: P_1= P_2= P_5 =0.6 s; P_3=0.43 s; P_4=0.44 s; P_6=0.59 s; P_7=0,58 s; P_8=0.58 s; P_9=0.28 s; P_{10}=0.28 s. Thus, the sum of all user actions in all the windows will be equal to 55.78 s. That is, the performance of the interface in this example is about 1 min.

4 Conclusion

In this paper, the structure and main functions of the access control systems were analyzed. We studied the requirements for the creation of user interfaces and the tasks that have to be resolved. Applying the ontological approach to the development of the human-machine interface and the principles of the user interface, the complex of visual object recognition was designed. The performance of the designed interface was calculated based on the GOMS method.

Furthermore, we created the application responsible for the human-machine interface that allows the user to move virtually in the building from camera to camera. To test the application of visual object recognition the video sequence received at the conference, which took place in the Moscow State Construction University, was used.

One of the advantages of the proposed approach for the interface design and development of user interface software is the possibility to configure it for almost all constructions and buildings with consideration of the subject area of the enterprise, location of zones with different access and external uncontrollable factors.

Acknowledgements. The reported study was supported by Russian Science Foundation (research project No. 14-49-00079).

References

[1] O. Danilov, Alternative interfaces [Al'ternativnie interfeisi], Komp'yuternoe obozrenie, 4, 1999, 14 - 17 (in Russian).

[2] J. Bardram, (1998) Collaboration, Coordination, and Computer Support: an activity theoretical approach to the design of Computer Supported Cooperative Work, Ph.D. thesis, University of Aarhus, 1998, 94p.

[3] V. Kaptelinin, Activity Theory: Implications for Human-Computer Interaction, Massachusetts Institute of Technology Cambridge, MA, USA, 1995, 103-116.

[4] A.N. Alfimtsev and V.V. Devyatkov, Smart multi-modal interfaces [Intellektual'nie mul'timodal'nie interfeisi], Kaluga,OOO«Poligraf-Inform», 2011, 238p. (in Russian).

[5] P. Langley, User modeling in adaptive interfaces, *Proc. of the Seventh Intern.Conf on User Modeling*, **407** (1999), 357 - 370. http://dx.doi.org/10.1007/978-3-7091-2490-1_48

[6] A.R. Puerta, Issues in Automatic Generation of User Interfaces in Model- Based Systems, Computer-Aided Design of User Interfaces, Presses Universitaires de Namur, Namur, Belgium, 1996, 323 - 326.

[7] I.L. Artem'eva Multilevel mathematical models of subject areas [Mnogourovnevie matematicheskie modeli predmetnih oblastei]. Iskusstvennii intellekt, T.4, 2006, 85-94 (in Russian).

[8] V.V. Gribova, A.V. Tarasov, The model of the ontology of the subject domain «Graphical user interface» [Model' ontologii predmetnoi oblasti «Graficheskii pol'zovatel'skii interfeis»], Informatika i sistemi upravleniya, 1(9), 2005, 316-321 (in Russian).

[9] D.A. Loktev, A.H. Alfimtsev, A.A. Loktev, The algorithm for object recognition [Algoritm raspoznavaniya ob'ektov]. *Vestnik MGSU*, **5**, (2012), 124-131 (in Russian).

[10] D.A. Loktev, A.A. Loktev, The method of determining the distance to an object by analyzing the blur his image [Metod opredeleniya rasstoyaniya do ob'ekta putem analiza razmitiya ego izobrazheniya]. *Vestnik MGSU*, 6, (2015), 140-151 (in Russian).

[11] D.A. Loktev, A.A. Loktev, Determination of object location by analyzing the image blur, *Contemporary Engineering Sciences*, **8** (2015), no. 11, 467 - 475. http://dx.doi.org/10.12988/ces.2015.52198

Received: July 21, 2015; Published: September 1, 2015