Research on Complex Multi-window HMI Construction Technology in Dispatching Automation System

Lin Zhao, Lingtao Zhang, Pai Sun
China NARI Technology Co., Ltd. Beijing 100192, China

Bo Zhuang, Guopeng Wang, Yan Geng
State Grid JiBei Electirc Power Co., Ltd. Beijing, 100053, China

***Abstract*—The integrated intelligent multi-window alarm interactive interface has been widely used in dispatching automation system in recent years. Integrated intelligent alarm picture is a multi-window interface which is composed of power-flow diagram, station maps, tables, text and other elements. The interface can represent a lot of information, but the process of building such interface is very complicated. In order to meet the demand of quickly generating such multi-window interface, it is necessary to break the limitations of custom programming method. To make great progress in the traditional HMI editor, we propose a new HMI editor to support constructing multi-window interface. This paper proposes a new dynamic layout implementation technique, which receive and process script language and parameter passing from graphic components, and gives an integrated monitoring interface instance to illustrate the mentioned technology. Picture editors or third-part developers will be able to build their own integrated intelligent alarms multi-window interface by using the provided picture editing tool, and that allows quickly response to the requirement of HMI.**

Keywords-multi-window interface; integrated intelligent alarm; HMI; picture editor; automatic generation

I. INTRODUCTION

There are two main kinds of monitoring pictures in dispatching automation system according to the graphical display, one is vector diagram which can be step-less zoom, such as power flow diagram, station map, interval graph, and the other is operation interface which consists of table, tree, text boxes and other UI components. The first type picture is produced by the picture editing tools for dispatching and monitoring personnel, and the second type picture is usually realized by programming. In foreign systems, the non-programming techniques for generating the second category of pictures is researched and achieved some results. For example the ViewstarICS system is developed by Cegelec and the OASYS system is developed by Valmet, which supports generating simple picture based UI controls by a picture editing tool. CC-2000 and D5000 systems are realized that a simple spreadsheet component can be embedded into the picture by picture editor in China.

In recent years, multi-window display screen represented by the application of integrated intelligent alarm began to be widely used [1-4] in China, which is the use of combination of the first kind and second kind pictures, such as station map, power flow diagram, and other basic graphical components, and trees, tables, and other UI controls, in the past this type of picture can only be achieved by the development of customized programming. EMS / SCADA systems does not have the ability to edit and generate comprehensive picture. This makes the realization of such man-machine interface is time-consuming, and also makes developers and users of dispatching automation platform are difficult to accomplish similar interface according to their needs.

The proposed approach has been greatly improved compared with the original picture editor and picture browser, so that you can edit the picture which is similar to integrated intelligent multi-window display alarm interface by editing tools, and the method shorten such interface realization cycle rapidly. The graphical editor and third-party developer can also independently make such a comprehensive display interface by the picture editor after they get some basic training.

II. OVERALL PROGRAM

Integrated Intelligent Alarm is composed of several pictures, each of which corresponds to one type of alarm. These pictures are composed of several sub-windows, and sub-window display content can be considered a graphical component, which divided into two categories. One is the picture display module used to display the power flow diagram, station map, and other interval graphics which related to the alarm[5-7]; , and the other is a custom component that is used to display the alarm itself and associated with the alarm some information.

The display module is consist of station map, power flow diagram, interval graphics, custom components which are generated by a specific editor tool[8,9], and the display components and custom components should be bonded on each sub-window, and the event handler is defined for each component. In the browser, the processor can call an interface which provided by external components based on the received event, and the processor is actually a script, which used the script language based on the actual needs of special design specialized languages, including variable definition, assignment, event content acquisition interface methods and components, and the use of standard SQL statements. After screen browser starts, when it receives the power grid alarm event, and will open the corresponding alarm picture according to the type of event and pass the event to the picture of the various sub-windows, and sub-window mobilize its internal binding component of the event processor for rendering components, then complete the entire display screen.

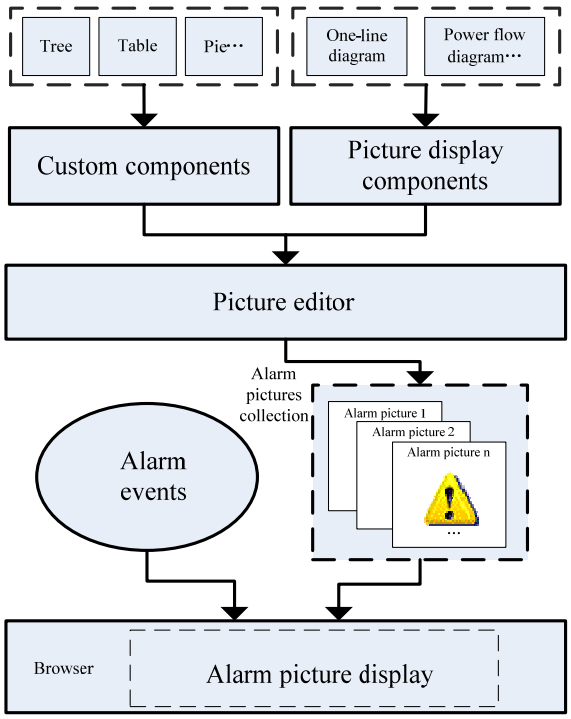

Figure 1. Intelligent multi-window alarm schematics

III. KEY TECHNOLOGY

A. *Picture Display Component*

Picture component is key part to draw all kinds of monitoring picture in dispatching automation system, and which also can be superimposed on some specific effect[10,11]. The main application of picture component in Integrated intelligent alarm is to draw station maps, power flow diagram and interval graphics, and also highlight some of the original equipment in the picture[10].

Picture component provides a set of interfaces, which will be called in the script by the picture editor, and decide the final picture status.

TABLE I. INTERFACE OF PICTURE DISPLAY COMPONENTS

Function	Description
void open(String picname, int zoom)	Open specified picture that name is picname，and zoom is Scaling parameters
void open(String picname)	Open specified picture which is adaptive sized
void locate(long id)	The specified component display in the center of the viewport
void visualization(long id,int type)	Superimposed on the figure a graph on a specific kind of display device id associated to this graphic, type is the specific effect, plastic figures, 1 flashes, 2 represents the effect can hang a sign with the actual needs continues to expand
Void display()	Picture display components show

B. *Custom Component*

Custom component is composed of graph, pie, bar graphs and other elements and tables, trees, text boxes and other UI controls. UI controls and conventional picture editor has been not well edited, and only some simple tables can be embedded in the picture, but the complex components cannot embedded in the picture.

The main reason is that it is difficult to find a effective method which combined the graphical representation of dispatching automation interface with the data of UI controls. This designed custom components editing tool is capable to edit and set tree component, tables and other common UI components by manually, which meets the requirement of power grid monitoring, as shown in Figure 2.

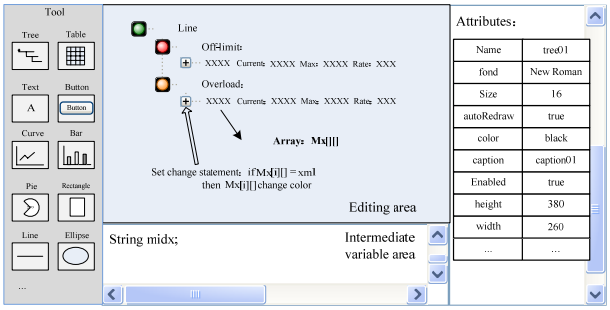

Figure 2. Custom component editor interface

Process is as following: first to drag a component into the editing area, the optional components include trees, tables, text boxes, etc. On the right side window, the basic properties of the components can be defined, including background color, foreground color, font, borders and other properties. In addition to the basic attributes, the specific properties is needed to define, and different components have different proprietary attributes. The tree component is the most complex component in the editor, so we illustrate the customization process by building one complex tree component.

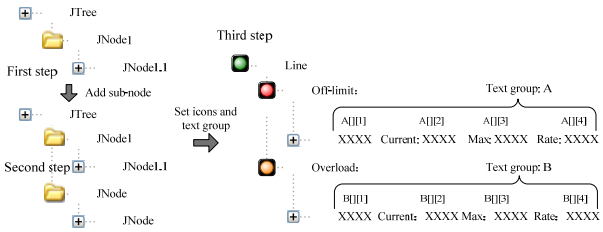

Figure 3. Definition process tree components

Above tree component is defined as following, the first step is to drag the tree into the editor, as shown in Figure.3 in

the tree component, including the root node of the tree component which the default text is the "JTree", a child node which the default text is "JNode1", and the second child node which the default text is "JNode1.1".

Because of the actual needs, a child node is added in the tree component, which the default text is "JNode2", and the lower level of the node as child nodes with default text "JNode2.1", as shown the second step in Figure 3.

The next step is to define texts added to the tree component, as shown in Figure. 3 in the third step, which the root "JTree" will be replaced by the text "Line", and "JNode1" will be replaced by "off-limit", "JNode1.1" will be replaced by text group "A", which is composed of static text and dynamic data, as following: "XXXX current value: XXXX limits: XXXX load factor: XXXX", where "XXXX" is dynamic data, and dynamic data means that data needs reading from the database or getting from the alarm event. The text of "JNode2" node is replaced with the word "overloaded", and the text of "JNode2.1" is replaced with text group B.

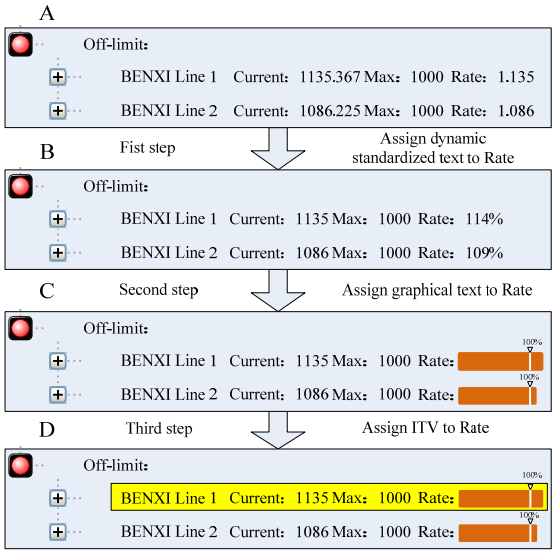

Figure 4. Tree component properties setup process

At this step, the tree that is defined above is shown as "A" in Figure.4. "off-limit" node display content can achieve the effect of "A" in Fig.4. However, such a tree can display data, but it is not intuitive. In order to make the tree has a better display of results, three methods are presented to convert general text to dynamic graphics, which are dynamic standardized text, text graphic, text conditions change.

1) Dynamic standardized Text: which is to control the floating point decimal places and whether to convert floating point into percentage.

2) Graphical Text: which replace text in tree with graphic components, and should apply to percentage. It convert percentage to a filled rectangles, circles and other shapes, and filled area is corresponding to the value of percentage, and the filled direction can be defined as needed.

3) Text condition transformation: change the text display style according to the conditional expression. When the condition statement is true, the text changes. The form of text transformation includes foreground color changing, background colors changing, blinking and text directly converted into another specific graphics.

Intermediate variables which is replaced by "ITV" in this article can be specifically identified when an alarm event occurs. The value is obtained from the alarm event by event processor module, then is passed to the custom component. Custom components can define one or more intermediate variables. Above tree is created, which is needed to define xm1 as ITV, replacing the "off-limit" line name. In the browser, when "off-limit" alarm event is received, the corresponding alarm screen would be shown, and the event handler gets the "off-limit" line name from the event tree body and assigns value to the intermediate variable.

Attaching text transformation conditions to the text group "A", the statement is as following:

```
if (A[][1] == xm1)
   then A.backColor   change   YELLOW
```

The meaning of the statement is that if the first column of text group "A" is equal to the intermediate variable, then the background color of the first column of text group "A" turns yellow. Thereby it is highlighted in the browser that current "off-limit" line occurred, and the show style is shown as "D" in Fig.4.

The above method is applicable to all types of UI controls as general methods, and integrated use of the above method enables components to achieve the desired display effect, and other UI components are more simple compared with tree component.

The components that defined above can be saved into files, and a set of custom components form custom collection. Custom component provides two interfaces, invoked by the picture editor as required in the script, then the display of custom components is completed. Custom components are used in conjunction with the subsequent introduction of a multi-window layout manager.

Function	Description
void setParameter（String param,String value）	For the custom components from intermediate variable assignment, the first parameter is the intermediate variable name, the second intermediate variable values.
void diaplay();	Custom Components display

C. *Multi-window Layout Manager and Component Binding*

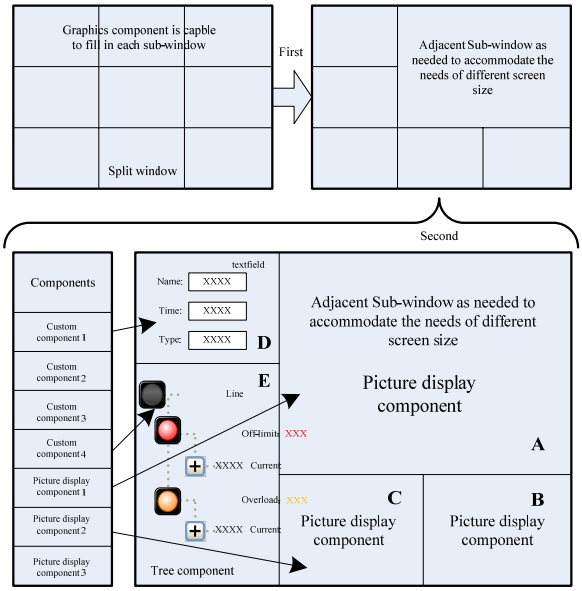

Figure 5. The process of building a multi-window interface

Multi-window layout manager is similar to drawing the table function in word. Layout step to define multi-window is showed in Figure 5 (a), and the window is divided into a plurality of sub-window by the ranks of dividing line. Sun-window supports merging and splitting, and splitting line is dragged by the mouse to resize the sub-window, and adjacent sub-windows make adaptive change. Sub-window can be loaded graphics display component and custom components.

The following integrated alarm display interface is as an example to describe the multi-window layout manager, and the basic layout is shown as Figure. 5(b). The display area in the upper right binds screen display components, and is for displaying the power flow diagram, and the two sub-screen display components that will display one-line diagram related to the line are bonded in the bottom right window. A custom component for displaying alarm briefing is bonded in the upper left window, and the second custom components is bonded in the bottom left side window. A tree is used to display more "off-limit", "overloaded" lines. At this point, the basic content on the line limit alarm integrated display screen is complete.

D. *Event Handler Definition*

Event handlers are connector among alarm events, the picture components and custom components, which is actually a script that is parsed by browser program and calls the interface provided by above components according to actual semantics. For "off-limit" alarm event, variable is included in the event body as showed in table II.

TABLE II. ALARM EVENT

Event variable	Description
EventID	Event type
desc	Alarm description
keyID	Line ID
keyDescr	Line description
st	Source station
zst	Destination station
value	Current value
time	Alarm time
stPicture	Source station name
zstPicture	Destination station name
mainPicture	Power flow diagram

The script attached to the window "A" is composed of variable definition, assignment, getting the variables in the event the body and interface calling of self-display component. It is as following.

```
String picName=Event.get("mainPicutre"); //Get the power flow diagram
String lnID    =Event.get("keyID");          //Get line ID
open(picName,200);                  //open picture , zoom out 200%
locate(lnID);                       //Locate    the line
visualization(lnID,2);              //attach a tag on the line
display();                          //display
```

Event handler script of custom components in E sub-window in Figure.5 (tree component) is described as following.

```
String ln_descr = Event.get("Descr");//get line description
setParameter（"xm1",keyDescr）//assign to the intermediate variables
display();                          //display the components
```

Event handler is customized for each component by using the above method.

IV. CONCLUSION

This paper studies a new technology that generate the multi-window integrated intelligent alarm HMI in dispatching automation system, and discusses the principle of generation, and designs the custom components, picture display components and the multi-window layout manager. It makes the picture editor who would be familiar with the database structure and SQL statements is able to draw more complex integrated alarm interfaces, and enhances supporting capabilities of dispatching automation platform. Interface generation technology is also suitable for generating other types of screens in dispatching automation browsers. The integrated alarm interface based on this technology has been operated stably and reliably in provincial power grid dispatching center.

REFERENCES

[1] LIU Si-cong, ZHU Zi-kun. Application of Intelligent Alarm System in Regional Dispatching System. Guangxi Electric Power, 2012, 35(6): 31-33.

[2] CHEN Jian-min, HAN Xue-jun, LUO Jing-nian, et al. Intelligent Alarm System Based on Protection Principle. East China Electric Power, 2012, 40（6）: 984-986.

[3] Liu Daowei, Li Xuchen, Zhao Hong, et al. Design and implementation of on-line stability and energy-saving comprehensive monitoring and early warning sys-tem[J]. Power System Technology, 2011, 35(1): 213-217(in Chinese).

[4] Yang Honggeng, Ming Jiao, Dai Haibo. Realization of intelligent alarm system for district power network[J]. Proceedings of the CSU-EPSA, 2011, 23(2): 105-109(in Chinese).

[5] Xiaodong Zhang, Quansheng Zhu, et al. Study on the Coordinated Multi-View Technology of EMS. Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific. 27-29 March 2012 : 1 – 4.

[6] ZHAO Lin, WANG Lili, LIU Yan, et al. Research and analysis on visualization technology for power grid real-time monitoring[J]. Power System Technology, 2014, 38(2): 538-543.

[7] SHEN Guohui, SHE Dongxiang， SUN Pai, et al. Research and application of power system visualization technology[J]. Power System Technology, 2009, 33(17): 31-36.

[8] ZHAO Lin， XIE Qiaoyun, LIN Jinghuai, et al. Key technologies for visualization function of Fujian energy management system[J]. Power System Technology, 2010, 34(S2): 365-370.

[9] GUO Chongjun, HONG Feng, CHEN Jinfu, et al. Application of visualization technology in electric power system[J]. Water Resources and Power, 2011, 29(2): 146-149.

[10] LIU Junyong, CHEN Jinhai, SHEN Xiaodong, LI Chengxin, Online Visual dispatch and early warning system of power grid[J]. Electric Power Automation Equipment, 2008, 28（1）: 1-5.