

Design and Implementation of an SVG Editor for Power System

Kehe Wu, Bohao Cheng, Jinshui Wu, Yue Yuan

School of Control and Computer Engineering, North China Electric Power University
Beijing, 102206, China
E-mail: wkh@ncepu.edu.cn

Abstract— It is the fundamental requirement for power system graphical editor to be able to draw rapidly, precisely and allocate corresponding configuration parameters for the power components and reserve operation interface. Element of this editor uses the SVG format and defines metadata tags based on the CIM specification carrying metadata in the SVG document which makes the graphics and data can be linked to a unified canonical form. The editor achieves the purpose as follows: editing, storing, and exporting a wiring diagram; importing a wiring diagram from other systems; automatically extracting and storing layers and elements in the importing process; creating and maintaining the user-defined primitive library or layer library and binding associated data. This SVG graphics editor for power system has met the functional requirements of the user and has been applied in several practical projects.

Keywords—SVG; Vector Graphics; electrical wiring diagram; Editor

I. INTRODUCTION

With the rapid development of computer technology and its applications, the scale of grid information system keeps increasing. Data and information associated with the grid are increasing. Compared to traditional forms of text and charts, the graphical presentation has irreplaceable advantages in terms of readability, ease of use, aesthetics, simplicity. In addition, the graphic has a higher information carrying capacity, and in the data display and monitoring of power system, graphics has been widely used^[1]. The visualization of the power grid monitoring and management has become one of the core technology of electric power information system in the graphics-based display platform.

Drawing and displaying electrical wiring diagram is the key fundamental task in the power data acquisition and supervisory control system^[2].

The subject is to realize a graphics editor developed to achieve this goal for power system.

II. DEVELOPMENT OF GRAPHICAL EDITOR FOR POWER SYSTEM

Currently, the wiring diagram of the power system release is a hot issue of the power graphics development system, and editors of most of the manufacturers have developed to satisfy the basic graphics needs, but only the system's ease of use, scalability, flexibility, compatibility, functional comprehensiveness and so on, are different.

In 1994, Han Feng, Sun Shuqi from Huadian Power Institute developed a software named HHCAD for the power plant or substation wiring diagram drawing by using the Turbo Pascal programming language^[3]. The principle of the function of drawing is preparing some basic drawing shapes, as well as the equipment used in a wiring diagram symbols in Turbo Pascal language procedure and then call them in the drawing process. In 2000, Liu Wenxia from North China Electric Power University (Beijing) combined with AutoCAD software to implement a automatic drawing software for a power system wiring diagram by using C language^[4]. This software divides the wiring diagram into a number of "meshes", and the split is based on each node topological relations between them. In 2008, Liu Honggang from Zhejiang University developed a set of time wiring diagram drawing software based on SVG technology and CIM modeling idea on NET platform using C # language^[5]. The software can achieve the drawing of self-defined primitives, and a wiring diagram can be drawn by graphical operating. In 2009, Zhang Fei from North China Electric Power University developed a power system graphical editor based on SVG by using the Java language^[6]. This editor can achieve the drawing of self-defined primitives, and a wiring diagram can be drawn by graphical operating.

Through above description, it can be seen that a graphics editor for the power system has experienced a difficult process, and the function of the development process became increasingly stronger. The design and implementation of the subject-oriented graphics editor of the power system also uses SVG common technical standards.

III. SVG INTRODUCTION AND FEATURES

A. About SVG

SVG is called the Scalable Vector Graphics, It is a graphical format which is based on XML (Extensible Markup Language) for describing two-dimensional vector graphics^[7]. January 14, 2003, SVG 1.1 was established as the W3C standards^[8].

SVG uses plain language of the text to describe the image content, and the structure of the document strictly comply with XML syntax specification, which is a resolution-independent vector graphics format^[9].

B. SVG Features

(1) It is an open standard. SVG has taken purely XML document structure^[10]. (2) High-quality display. (3) The

object is described by the text. SVG plains text to describe vector graphics, which gives SVG image file good readability and smaller volume. What's more, the compressibility is stronger than before ^[11]. (4) It is full support for DOM. (5) interactive and dynamic. SVG provides dynamic interaction ^[12].

IV. OVERALL DESIGN OF THE EDITOR

With analysis based on actual demand, the system should include four functional modules: core graphics module, a resource management module, file management module, electrical parameters configuration module.

The outline design needs to solve one of the most important parts, which is how the graphical interfaces "shell" and the Document Object Model (DOM), and the "core" can be linked up, as shown in Figure 1. That is how the in-memory DOM can render an image on the screen, as well as how the user can modify the operation of the image on the screen that updates to the DOM.

V. ELECTRICAL PRIMITIVE PLUG-IN MODULE

This module is to insert the existing electrical primitive into the canvas. By inserting the electrical diagram into the canvas, together with the basic drawing module, you can complete the work of drawing an electrical wiring diagram.

According to the actual situation of the power system, often each city has its own set of primitive library. So, the respective cities are the first layer classification level of all primitives, and then, each city's diagram elements are grouped again according to the primitive types of equipment, which means the type of equipment is regarded as the second layer classification level.

According to this organization, I design two tables named SymbolGroup and Symbol, which are used to store primitive group information and metafile information, respectively.

VI. PRIMITIVES LIBRARY MANAGEMENT MODULE

This module realizes the establishment and maintenance functional of self-defined primitives. View, grouping, edit, renaming and other operations of the existing primitives can be achieved through this module. At the same time, through this module, the user can also draw a new element, and add it to the library.

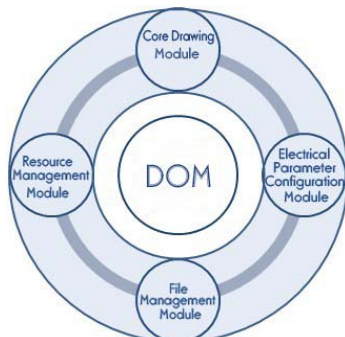


Figure 1. Graphical interface and document object model showing the relationship

VII. LAYER LIBRARY MANAGEMENT MODULE

This module realizes the view and maintenance functions of the layer library. Users can add, delete, change, search and other operations of the layer through this module.

In the actual use of power system, SVG format electrical wiring diagram identifies the layer as a custom tag attribute (such as layer attributes), and adds to each SVG object. When the specified object needs to be moved to a layer, what we need to do is to simply modify the value of the property to its layer, which is simple. As it is to add new properties to function, the original editor modifies little, and it's easy to expand on the basis of existing functionality.

As the scale of a wiring diagram of the power system is not too large, with a volume of general 200kB, after the actual test, the solution in the storage space and processing performance is acceptable.

VIII. THE DESIGN AND IMPLEMENTATION OF THE ORIGINAL IMAGE IMPORT MODULE

The modular system should be able to import SVG format primary connection diagram from other systems SVG, to reduce the workload of the repeated drawing. Meanwhile, in order to be able to use the original primitive material image to edit, the original system should also be able to automatically extract and store the primitives and layers information during the import process.

According to the actual situation of the power system, the cities' existing electrical wiring diagram generally use the tools provided by third-party manufacturers. Due to the different manufacturers of document structure definition, it causes some difference in structure of the electrical wiring diagram of different cities. The strategy pattern is used in the system to solve the problem.

Context is on behalf of the application context, as is shown in Figure 2, it needs to use the algorithm provided by the class of specific strategies. Strategy is an abstract strategy class that defines a common interface to a variety of different algorithms in different ways to implement this interface. Context uses this interface to call different algorithms. Concrete Strategy concrete strategy class, which implements this defined interface of strategy, and provides a specific algorithm.

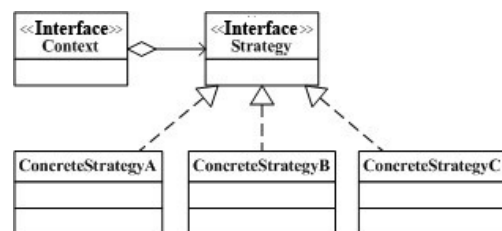


Figure 2. Strategy pattern class diagram

IX. DESIGN AND IMPLEMENTATION OF DATA SOURCE BINDING MODULE

The module user can associate static images and dynamic changes in the data, so that other systems can refresh element status and real-time data points in the real-time display of the image. In addition, the use of these wiring diagram systems may need special interactive operation of certain components, so the module is also defined in the figure earmarked interface for other systems for these operations.

According to the actual application of the power system, the associated data has the following two conditions: (1) the association of equipment and data. I design equipment settings associated data source, in order to refresh dynamic device status (such as configure a switch to remote signaling data source) or to pull equipment from other system information (such as equipment ledger, evaluation, defects). (2) Inserting the measured values. That is directly inserted in figure measurement point, in order to refresh and display these measurement point data (such as telemetry and temperature measurement data displayed in real time on the electrical wiring diagram) and real-time image in the foreground.

For the SVG file structure in both cases, CIM has given the described specifications. The solutions of both application design and implementation based on CIM standards are given in this article's sections 8.1 and 8.2 respectively.

A. Device and data association

The CIM namespace needs to be declared in the list of attributes of the document's root tag using CIM defined label. I write <use> tag to define a switch to store related

information and use metadata tags (<metadata>) in order to switch the associated data source. In the <metadata> Label, I associate equipment identification, from the production of MIS and give the name of the device in the system. Finally, I use <g> label wrapped in electrical components and the metadata together to constitute an associated electrical element of the data source. When the figure is being used by other systems, you can find the appropriate data based on metadata tags for display or processing.

B. Inserting the Measurement values

It shows an SVG document described in the CIM specification, which defines a measured value of the text, and it uses <metadata> label to associate with a measured value of the data source.

```
<?xml version = "1.0" encoding = "utf-8" standalone = "no"?>
```

```
<svg width="1200" height="1000"
xmlns:cge="http://www.cim.com">
```

```
<g id="33000735">
```

```
<text x="845" y="515" font-size="16"
stroke="rgb(255,255,0)"> -000.0 </text>
```

```
<metadata>
```

```
<cge:Meas_Ref ObjectID="251000836_P" ObjectName="2#
Zhubian-high :P"/>
```

```
</Metadata></g></Svg>
```

Compared with section 8.1, we can see that the binding measured values and telemetering binding way are basically the same. The only difference is: telemetering is bound to specific electrical primitives, and the measured value does not depend on any electrical diagram, but draw directly on the canvas.



Figure 3. Editing workspace and properties panel

X. CONCLUSION

A. Conclusion

This article describes the design and implementation process of the SVG editor of a power-oriented system. The primary power system connection diagram can be drawn by the editor and the editing workspace and properties panel which are shown in Figure 3. The SVG editor for power system described in this article has been applied in several practical projects, and these specific studies include the following aspects:

(1) Editor provides a graphical user interface that allows users to draw an electrical wiring diagram quickly and accurately; editor is able to save the SVG files in local disk and the database.

(2) During the import process, layers and symbols are automatically extracted and stored. Meanwhile, in order to take advantage of the original image material for graphic editing, the editor will also be able to extract and store symbols and layer information in the import process.

(3) Drawing and Storage of custom symbols. This editor provides symbol management functions. And it provides functions of editing existing symbols and creating new symbols. In this way, users can easily create and maintain custom symbol library.

(4) Editor provides layer management functions to manage the imported layers. Also, it supports to hide or lock layers.

(5) The editor can associate static images with dynamic data, so that the graphics can be refreshed when they are displayed in other systems.

B. Prospects

In the practical application of the process, the editor can meet the basic demand of drawing, but there are still many deficiencies and improvements left:

(1) Function is still not perfect. For example, in some applications, it is necessary to save topology so that we can do the trend calculation lately.

(2) The level of code reuse is not high enough. I didn't make the similar function in the development process and the public part of the extracted package into separate scalable components, which will undoubtedly increase expansion and maintenance costs in the future.

To overcome the above shortcomings, the next step is the study and improvement of the system:

(1) Mining business features and functional requirements of the power industry, improve the business functions of the editor.

(2) The reconstruction and optimization of the code structure, and improve code reuse of scalability. Try to improve system flexibility and scalability by taking multi-use configuration files into account.

REFERENCES

[1] D. Shi, Y.S. Lu. Study of the Application of SVG in Power System Graphicalized Software, *Relay*, 2004, 32(16), P45-47 (Ch)

[2] Draft IEC61970: Energy Management System Application Program Interface (EMS-API)-Part503: CIM / XML Model Exchange Formats, 2004

[3] F. Han, S.Q. Sun, G.L. Shi, A New Software for Drawing Primary Electric Wiring Diagram, *Hebei Electric Power*, 1994, 4 (1), P54-59 (Ch)

[4] W.X. Liu, Z.Q. Wang, S.X. Ma, Automatic Drawing for the Main-line Diagrams of Power Systems, *MODERN ELECTRIC POWER*, 2000, 17 (3), P39-45 (Ch)

[5] H.G. Liu, SVG-Based Circuit Graphics System Design and Development

[6] F. Zhang, The Design and Implementation of a SVG Editor for Power System, China Electric Education, 2009.3

[7] W3C. Scalable Vector Graphics (SVG) 1.1 Specification. <http://www.w3.org/TR/SVG/>

[8] W3C. Scalable Vector Graphics (SVG) 1.2 Specification [EB / OL]. <http://www.w3.org/TR/2004/WD-SVG12-20041027>

[9] Adobe SVG Viewer. <http://www.adobe.com/svg/viewer/install/main.html>

[10] The World Wide Web Consortium (W3C) Site of Scalable Vector Graphics (SVG) Format. <http://www.w3.org/Graphics/SVG/Overview.html>

[11] Apache Batik SVG Toolkit [EB / OL]. <http://xml-graphics.apache.org/batik/2006>.

[12] C. Baru, A. Behere, C. Cowart. Representation and display of geospatial information: A comparison of ArcXML and SVG. *Web Information Systems Engineering*. 2001. Proceedings of the Second International Conference on, 2001.2 (5) :48-53