

Application of Highcharts in the Analysis of the Multi-source Track Inspection Data

WANG Yijun

Institute for Railway Information
Engineering, Central South University
Changsha, China

WANG Jun,

School of Software, Central South University
Changsha, China
E-mail: object@aliyun.com

Abstract—In the railway permanent way department, the daily work of maintenance, inspection and repair depends on a large amount of inspection data which comes from various ways, such as dynamic/static monitoring devices, mapping on site, design standards or railway rules. These data needs to be imported from various interfaces. After calculation processing, these data will be provided to the railway permanent way system in way of “Data Visualization” for developing a maintenance plan or repair effect analysis. Consequently, the Management System of Railway Quality Information is built in the B/S mode. It is based on the use of third-party tools for JavaScript libraries, Highcharts. This system implements dynamic data graphics and operator information exchange, and the data sharing between branches becomes easier.

Keywords—component: Data Visualization; Web; Track Inspection; Highcharts; JavaScript

I. INTRODUCTION

Under repeated loads of train wheels, track components deteriorate gradually, which may be manifested in track geometry changes. The changes will cause track geometry parameters to deviate from their design values, and thus result in track irregularities^[1,2]. Track inspection data contain different dynamic or static inspection data. Nowadays, these data are basically analyzed by the railway permanent way departments with single-function soft-wares. It's not easy to processing or sharing data from a lot of dedicated soft-wares. As such here we build the Management System of Railway Quality Information to achieve efficient data sharing and comprehensive analysis. Besides, the system can generate scientific assessment and support decision.

In this system, the most critical aspect is the integration of data from different sources to charts. These charts are necessary to be able to deliver a sufficient amount of information, but also to make use of each type of data becomes clear. Among the many techniques, we find a suitable development tool to create complex comparison-charts. Although the Java language comes

with Applet support charting classes, but this technology has many limitations, it requires browser support for the Applet. Client requires additional installation configuration procedures, this will give the system adversely affect application scenarios^[3]. For example, you can not use the browsers in mobile smart devices to view Applet pages. Applet also restricted access and use of the user's local files, that makes it currently applies only to the LAN. Although flash-based graphics technologies has better adaptability, when dealing with very large amounts of data, flash takes a long time to load data, and be difficult to guarantee data security^[4].

In the third-party tools library, Highcharts could satisfy our needs. It is based entirely on JavaScript. This tool has a relatively low resource overhead, and a faster response speed. Users do not need to install any plug completely, truly cross-platform. Developers are not limited by browser compatibility and development languages^[5]. It not only comes with a wealth of interaction and display, but also can add more interactive contents by their interface with JavaScript.

This article focuses on achieving contrast showing multiple track inspection data in a chart, and the interactive demonstration of curve information in the corresponding line.

II. DEVELOPMENT STEPS

A. Configuration and Installation

Highcharts requires two files to run, highcharts.js and either jQuery, Moo Tools or Prototype or our own Highcharts Standalone Framework which are used for some common JavaScript tasks. Files needed for this service can be introduced directly from an Internet address, which is connected to development team's servers. However, considering the system server deployment environment and the Internet are not always communicated, we chose the better reliability local file configuration. The Highcharts files can be downloaded from “highcharts.com” and put on your web page.

- Load files from your own domain, use this code to

include Highcharts with jQuery:

```
<script src="js/jquery.min.js"></script>
<script src="js/highcharts.js"></script>
```

- Add a div in your web page. Give it an id and set a specific width and height which will be the width and height of your chart.
- A chart is initialized by adding the JavaScript tag, `<script> </script>`, anywhere in a web page, containing your setting code for jQuery.

Highcharts runs entirely on the client, and works with any web server that can deliver HTML and JavaScript content. Whether your server is PHP, Perl, ASP, ASP.NET, Node.js or whatever, Highcharts is completely ignorant of it. The HTML/JavaScript files may also be loaded from the file system, which is the case in app platforms where Highcharts is loaded in a web component inside the app. Its PhantomJS script package makes it possible to run Highcharts on the server without a client internet browser involved.

B. Development steps

1. Find and read the data from the database, set up a data source. Making it with JDBC and Data Access Object, or with an Object/Relational Mapping (ORM) framework, such as Hibernate^[6,7].
2. Processing and assembling data, save it to generate the required data-set object corresponding the chart, it needs the help of the service implementation class.
3. Create a container in suitable location in the web page, give it an ID which is the same as the configuration

object.

4. Highcharts use a JavaScript object structure to define the options or settings of a chart. Make and apply a set of options.
5. Specify the data through the “series.data” option directly on the configuration object.
6. Add codes of customize contents.

III. REALIZATION OF CUSTOM INTERACTIVITY

B. Proposed special needs

According to our research on the railway permanent way department. We found the following actual demand is difficult to use the tool's own characteristics to achieve:

1. Move the mouse to the chart and stay within a certain area, if this area belong to a curve segment (judging by checking with odometer in curve tables), a message table is automatically suspended the mouse position. This table is used to show the details of the curve segment information. Once the mouse is moved to the area outside of the curve segment, this information table will disappear.
2. In a standard chart in highcharts, one can not view the neighboring part of the chart after enlarging some area in the image, unless exit zoom and select the neighboring part and click “zoom” again. In special demand of track detection, the mileage should be displayed continuously. That's why we need to find a way to show neighboring lines in the chart without exiting the “Zoom” mode.

C. Method of implement

1. To implement this need, we make flexible use of option's property in Highcharts. When a plot band is used on both the x- and y-axis, the plot band on the y-axis will be shown in front, as seen in the Fig .1^[8].



Figure 1. plot bands in Highcharts

"events" are an object which can be created inside a plot band or plot line object. This object defines mouse events according to the band or line. Values that are supported are click, mouseover, mouseout and mousemove.

When the mouseover event become effective, the mouse location is recorded. Then, initiate a request to the back end servlet to get curve data. After that the web page make a data table and set this table visible in the position

```
recorded.  
plotBands:  
{color: '#ffffff',  
from:Position1,  
to:Position2,  
events: {  
mouseover:  
function(e){mouseOverHandler(e,Position1,Position2)},  
mouseout:  
function(e){document.getElementById('showCurveInfo').s  
tyle.display = 'none';  
}  
}
```

Every time the mouse enters the plot band, div of information window is actually asynchronous partial refreshed, that makes us thinking about the need for the use of Ajax technology^[9]. After testing, we found that the plotBands events originally support asynchronous partial refreshing.

2. To solve the second problem, we must first analyze the characteristics of the data set, since the main test data mileage data is sampled at equal intervals, we can use it at dividing the data block. For example, when zooming the chart, we select the first half of the two-dimensional array with mileage arranged, re-assemble it into a new data set,

execute the drawing function again. So we get a new chart drawing with first half of the data. Similarly, "narrow" can be interpreted as to get data of twice size than before from the total data set; "Shift Left" can be interpreted as to get data of neighboring front isometric data; "Shift Right" can be interpreted as to get data of neighboring back isometric data. After getting new data set, execute the drawing function again. This function will not cause the whole page refreshed, because The whole process does not need to request to background programs once again.

When retrieving the data set, boundary conditions require extra attention,data points at both ends need to be made of special treatment. Curve data is not sampled at equal intervals, when zooming or other interactive action happens,we retrieving those data at equal intervals first, then dealing with curve data by the two ends of the mileage obtained.

D. Performance

1. There is one curve segment from mileage 250.3(km) to 251(km), a plot band can be seen from the chart in this area. Once moving your mouse in this special area, a float table of curve data could be displayed at the same location.

2. In Fig .3, the full picture of the left align data in one test shows the existence of multiple overrun^[10].



Figure 3. full picture of the left align data

When user click the button “zoom up”, the chart will change into Fig .4.



Figure 4. zoom up the first part

Multiple use four direction buttons ,show mode.
neighboring lines in the chart without exiting the “Zoom”



Figure 5. Joint use of the up/down and left/right button

IV. SUMMARY AND FUTURE RESEARCH

In a variety of data visualization techniques, combined with the actual situation in the actual project, we select the appropriate technology. Besides, the limitations of develop tools could be overcome with innovative approaches. Results from this study are expected to foster the applicability of the approach in practice.

In order to improve the level of informationization of railway track detection, more research needs to be done. Track detection information combined with GIS will be

next object. The track curve and slope information will be displayed not only in a two-dimensional line, but also in 3D GIS.

REFERENCES

- [1] Peng. Xu and Quan-Xin, “Sun.Key Equipment Identification model for correcting milepost errors of track geometry data from track inspection cars”, Transportation Research Part C, 35, 2013, pp.85-103.

- [2] Luca. Podofillini and Enrico. Zio, "Risk-informed optimization of railway tracks inspection and maintenance procedures", Reliability Engineering and System Safety, 91, 2006, pp. 20-35.
- [3] Li. Boyu and Zhao. Lili, "A study on the dynamic graph method in JSP", Journal of Computer Applications, 23, 2003, pp. 213-214.
- [4] Liu. Zhile, "Web2.0 application security in-depth analysis", Telecommunications Network Technology, 3, 2012, pp. 21-22.
- [5] Highcharts Product, Highcharts, Feb. 2015, Available: <http://www.highcharts.com/products/highcharts>.
- [6] Christian. Bauer and Gavin. King. Java Persistence with Hibernate, Manning Publications, 2006, pp.5-16.
- [7] Jia. Chen and Zheqiong. Yan, "Design and Implementation of Middle Layer for Detached Criteria Query Based on JSF and Hibernate Framework", Journal of Software, Vol. 7(2), 2012, pp.277-283.
- [8] Highcharts API, Highcharts, Feb. 2015, Available: <http://www.highcharts.com/docs/chart-concepts/plot-bands-and-plot-lines/>.
- [9] Ke. Changzheng and Huang Houkuan, "The principle and application of Ajax". Railway Computer Application, Vol. 16(1), 2007, pp. 27-28.
- [10] Liu. Xueyi, Detection Technology for the Railway Permanent Way, Railroad Publication House of China, Beijing, April. 2011.