ANADOLU ÜNİVERSİTESİ

Bilim ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik
Cilt: 14 Sayı: 3 2013
Sayfa: 191-198

ARAŞTIRMA MAKALESİ/**RESEARCH ARTICLE**

**Mustafa ULVİ[1], Süleyman EKEN[2], Ahmet SAYAR[3]**

# SERVICE ORIENTED VISUAL INTERPRETATION TOOL FOR TIME SERIES DATA

## *ABSTRACT*

In statistics, signal processing, econometrics and mathematical finance, a time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals. In this paper, the proposed system for time-series data visualization is a distributed loosely coupled system enabling data and resource sharing over the Internet, for data rendering, we use Open Graphics Library (OpenGL) and extend them as Web Services to be used by the third parties for visualization of time-series data.

**Keywords:** Time series, Data graphs, OpenGL, Web services

## ZAMAN SERİLİ VERİLER İÇİN SERVİS TABANLI GÖRSEL YORUMLAMA ARACI

## *ÖZ*

İstatistik, sinyal işleme, ekonometri ve matematiksel finansta zaman serili veriler, eş zaman aralıklarına ayrılmış ardışık zamanlarda tipik olarak ölçülen veri noktaları dizisidir. Bu makalede, zaman serili verilerin görselleştirilmesi için önerilen sistem; internet üzerinden veri ve kaynak paylaşımına izin veren dağıtık gevşek bağlı bir sistemdir. Verilerin işlenmesi için OpenGL Kütüphanesi kullanıldı ve bu kütüphane Web'e servis edilerek üçüncü kişiler tarafından zaman serili verilerin görselleştirilmesi sağlanmıştır.

**Anahtar Kelimeler:** Zaman serileri, Veri grafikleri, OpenGL, Web servisleri

[1], Kocaeli University, Department of Computer Engineering, Umuttepe Campus, Kocaeli, 41380
E-mail: ulvimustafa@gmail.com

[2], Kocaeli University, Department of Computer Engineering, Umuttepe Campus, Kocaeli, 41380
Tel: +90 262 303 35 96, E-mail: suleyman.eken@kocaeli.edu.tr

[3], Kocaeli University, Department of Computer Engineering, Umuttepe Campus, Kocaeli, 41380
Tel: +90 262 303 35 83, E-mail: ahmet.sayar@kocaeli.edu.tr

**Recieved:** 28 November 2012 **Revised:** 03 June 2013 **Accepted:** 10 June 2013

## 1. INTRODUCTION

Understanding temporal relationships enables us to learn from the past, predict the future and plan accordingly. Therefore, the analysis of time-oriented data is a key concern in visual analytics, where the goal is to support decision making and knowledge crystallization with appropriate visual methods.

Time series graphs are vital importance in different applications of statistics. When recording values of the same variable over an extended period of time, sometimes it is difficult to discern any trend or pattern (the weather, business models, populations etc.) (Yılmaz, 2003). However, once the same data points are displayed graphically, some features jump out. Time series graphs make trends and patterns easy to spot. These trends and patterns are important as they can be used to project into the future (Brillinger, 1975).

The proposed system for time-series data visualization is a distributed loosely coupled system enabling data and resource sharing over the Internet, for data rendering, we use Open Graphics Library (OpenGL) and extend them as Web Services to be ised by the third parties for visualization of time-series data. The whole system is a Web Service based application of Service Oriented Architecture (SOA) paradigm.

OpenGL is a standard specification defining an industry-leading, cross-language, cross-platform application programming interface for writing applications that produce 2D and 3D computer graphics, and only major API with support for virtually all operating systems. In this sense, OpenGL helps us to show collected time series data. Web service developed provides great ease for people who need to show data in a graph. Also, Web service can be integrated into various applications by many distributed systems. In order to show the work logic of Web service, an application has been developed by utilizing JSP technologies (Richard, 2003). Our application with Web services provides to obtain needed graph images via a browser, even though library related to with OpenGL graphic is not installed on computer.

The remainder of this paper is organized as follows. In Section 2, relevant works are presented. In Section 3, service oriented visual interpretation tool architecture is proposed. Section 4 gives some results and Section 5 draws a conclusion.

## 2. RELEVANT WORKS

Visualization has proven to be an effective approach to analyze time-series data. A well designed visualization help in answering the following questions for unknown temporal data (MacEachren, 1995):

- Does a data element exist at a specific time? (Existence of a data element)
- When does a data element exist on time? Is there any cyclic behavior? (Temporal location)
- How long is the time span from beginning to end of the data element? (Temporal interval)
- How often does a data element occur? (Temporal texture)
- How fast is a data element changing or how much difference is there from data element to data element over time? (Rate of change)
- In what order do data elements appear? (Sequence)
- Do data elements exist together? (Synchronization)

The visualization of time-dependent data has a long history. Time-series plots appeared for the first time in the illustration of planetary orbits in a text from a monastery school in the 10th or 11th century (Tufte, 1983). In science, time-series charts have been rediscovered not earlier than in the 18th century by Lambert. He applied line graphs to display periodic variation in soil temperature in relation to depth under the surface (Lambert, 1779). Today, a wider repertoire of techniques to visualize time-dependent data is available.

There are two cases of visualization representation of time-dependent data, based on their time dependence: static representation and dynamic representation. In the static representation, the visual representation does not automatically change over time, except for the modification results from user interactions.In the dynamic representation, the visual representation changes dynamically over time and is a function of time. Both forms have their specific benefits, and it has to be decided based on the concrete task, which kind of representation should be used.

Bertin (1983) distinguished eight visual variables which could be used to encode nominal, ordinal or quantitative data visually. Mackinlay (1996) carried on this work and presented an enhanced ranking of visual variables.

More conventional approaches are based on the mapping of time on an available quantitative scale. In case of a 1D scale this leads to a Sequence Graph, in case of a 2D or 3D graph to a Time Series Graph. Even one step further is approaches which succeed in linking such independent representation of data for each time-step to a single map. Examples are the Change Chart, Stacked Bar Chart, and Parallel Coordinates Technique (Inselberg, 1997). Most of the techniques mentioned above have been designed with the focus on presenting simple forms of time-series data.

Nowadays, the analysis of multivariate time-series data becomes more and more important. Visualizing multivariate data over long time periods requires special effort. A well-known technique, ThemeRiver, (Havre, 2000) has been developed for document visualization. Another technique specifically designed to support the comparison and analysis of cyclic data is the Spiral Graph (Weber, 2001). The time axis is represented by a spiral.

Another way to present time dependent data is to use special visual metaphors. One of such metaphor being particularly intuitive is the Calendar View (van Wijk, 1999). Other examples for useful metaphors are a clock or a pencil. The technique SpiraClock uses a clock to present cyclic data (Dragicevic, 2002).

Another common visualization approach is axis-based visualization. In this case axes are drawn and scaled with regard to the range of associated variables. The TimeWheel technique presents the time axis in the center of the display, and arranges the other axes circularly around it.

In our approach, time-series data is plotted and displayed as line and bar graphs. When the user wants to plot graph, he/she enters time series data according to the format and sends them Web service by means of proxy class. Image file is created with respect to sending information and sent back to the user. The user can enter time series through interface or perform the drawing for the values registered in the file.

## 3. WEB SERVICE BASED VISUALIZATION

In the field of service-oriented architecture and distributed computing have recently been major advances. Service-Oriented Architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services which are well-defined business functionalities that are built as software components (discrete pieces of code and/or data structures) that can be reused for different purposes.

To the first published research of service orientation was provided by (Thomas Erl, 2005), one of eight specific service-orientation principles is loose coupling. The principle of loose coupling is that services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other. SOA is a principle concept underlying Web services implementations. It provides loose coupling between software components.

The fundamental component in SOA is services. SOA is based on three basic components (Figure 1): service provider, service requester and service registry. Operations among those components are summarized as publish, find, and bind-invoke. Service provider publishes services to a registry and provides services available on the Internet for the requests of clients. Service requester finds needed service and accesses it by means of performing service discovery operations on the service registry.

Service registry helps service between service provider and service requester.

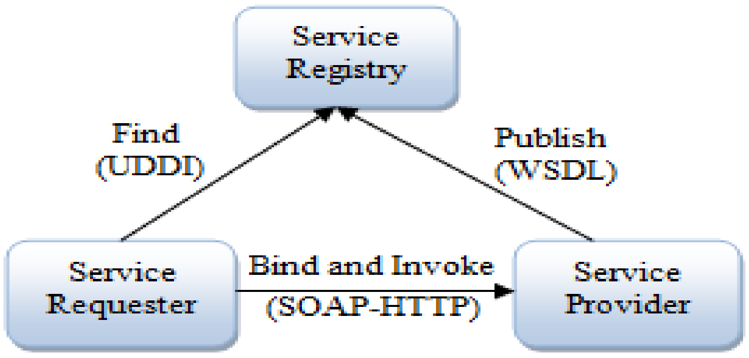


Figure 1. The Fundamental Operations and Components in SOA

Web services can be used to implement architecture according to SOA concepts, where the basic unit of communication is a message, rather than an operation. This is often referred to as "message-oriented" services. A Web service is simply an application that makes a function accessible using standard Web technology and Web services standards to support interoperable machine-to-machine interaction over a network (Doyle and Reed, 2001).

Web Services have three basic platform elements: SOAP, WSDL and UDDI. These standards are a series of protocols that support sophisticated communications between various nodes in a network. These essential elements can be described as following:

- SOAP (Simple Objects Access Protocol) is an XML-based communication protocol to let applications exchange information over HTTP.
- WSDL (Web Services Description Language) is an XML-based language for locating and describing Web services.
- UDDI (Universal Description, Discovery and Integration) is a directory service where companies can register and search for Web services (Web3school).

The main advantage of Web services is that the service can be used remotely without the user's actual knowledge and intervention and by multiple users at the same time, eliminating the need for constant updates to locally installed software. Moreover it minimizes the network traffic, since data do not need to be transferred to the client in every step of the operation (Kotzinos and Chrysoulakis, 2003).

The proposed system consists of two basic components as shown in Figure 2: SOA Client and SOA Server. Here, SOA client can be a desktop program or developed with any software such as web based internet technologies. For data rendering and plotting as 2D-3D graphs we use a Jogle OpenGL library which is developed for JAVA.

All users are able to adapt the application to their own programs using WSDL. The proxy class handles the work of mapping parameters to XML elements and then sending the SOAP message over the network. A proxy class is a class containing all of the methods and objects exposed by the Web service. In our application, proxy class has three variables. Two of these variables are list type variables holding x-y coordinate values (data that the user wants to plot the screen) and the other one is string type variable holding graphic header information, names of x-y axis, and graphic type (1 means straight line and 0 means dashed line). In short, the format of chart font information is as follows: *graphic_header_information#x_axis_name#y_axis_name#graphic_type*. Created string is send SOA server.

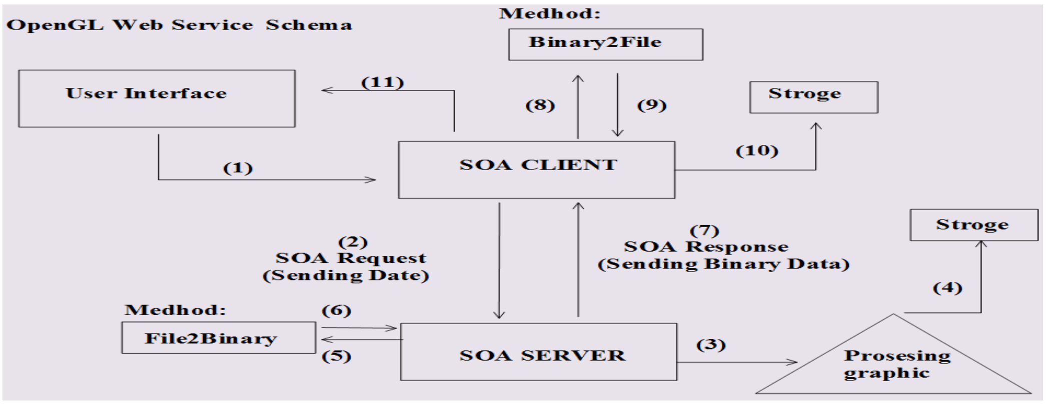
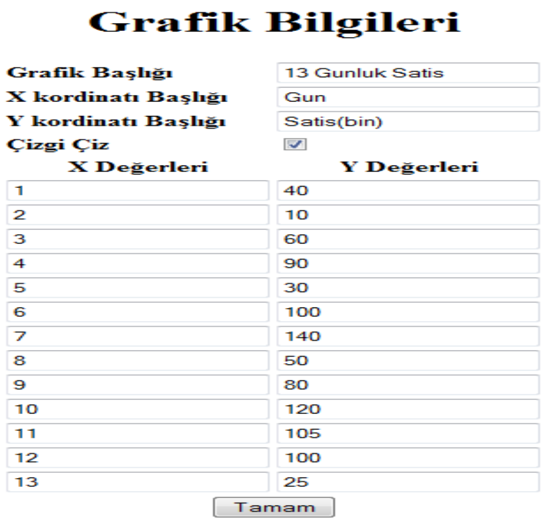


Figure 2. The prototype system architecture of visualization tool through web services

The general working of the system is that user selects which type of the graphic (solid or dashed line) he/she wants to plot via the GUI. At the server side, image file is created based on the user provided parameters and the data itself. The data is fed into the system through two alternative ways. One is using text files with a standard fortmat illustrated in Figure 4, and another is using interactive user interfaces to enter the data set one by one manually. The detailed explanations of working steps are listed below.

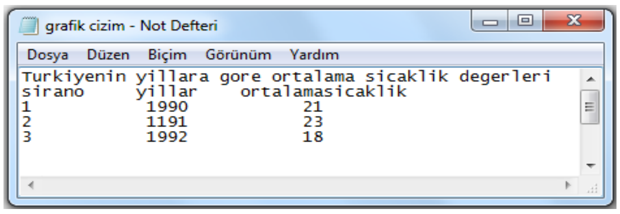
Step 1: The user enters time series dataset through the interface (Figure 3) or uploads a text file containing the data to be plotted (Figure 4). Text file needs to be in a predefined format.


Grafik Bilgileri

| | |
|---|---|
| Grafik Başlığı | 13 Gunluk Satis |
| X kordinatı Başlığı | Gun |
| Y kordinatı Başlığı | Satis(bin) |
| Çizgi Çiz | ✓ |
| X Değerleri | Y Değerleri |
| 1 | 40 |
| 2 | 10 |
| 3 | 60 |
| 4 | 90 |
| 5 | 30 |
| 6 | 100 |
| 7 | 140 |
| 8 | 50 |
| 9 | 80 |
| 10 | 120 |
| 11 | 105 |
| 12 | 100 |
| 13 | 25 |

Figure 3. User interface

Data held in text file format is as shown in Figure 4. In this format, the header information (title) is located on fist line. The first column is line number. Respectively, x-axis name, its values, y-axis name and its values on are located on other columns.



```
Turkiyenin yillara gore ortalama sicaklik degerleri
sirano       yillar     ortalamasicaklik
1             1990            21
2             1191            23
3             1992            18
```

Figure 4. Representation of data stored in text file

Step 2: After getting the parameters and the data, client sends this information to the web service. This is called SOA request.

Step 3: Web service receives user-defined information and creates graphic image.

Step 4: Graphic image is saved into the storage (server's local file system).

Step 5, 6: Graphic image is then converted to binary array by means of "File2Binary" method.

Step 7: Binary array is attached to the SOAP message and the message is sent to the client over the network.

Step 8, 9: At the client side, returned binary arrays are converted to an image file such as jpeg or tiff.

Step 10: Image is saved into the storage.

Step 11: Image is displayed on a GUI screen.

For the group of time series data entered via user interface (Figure 3), line graph image is as shown in Figure 5 and bar chart image is as shown in Figure 6.

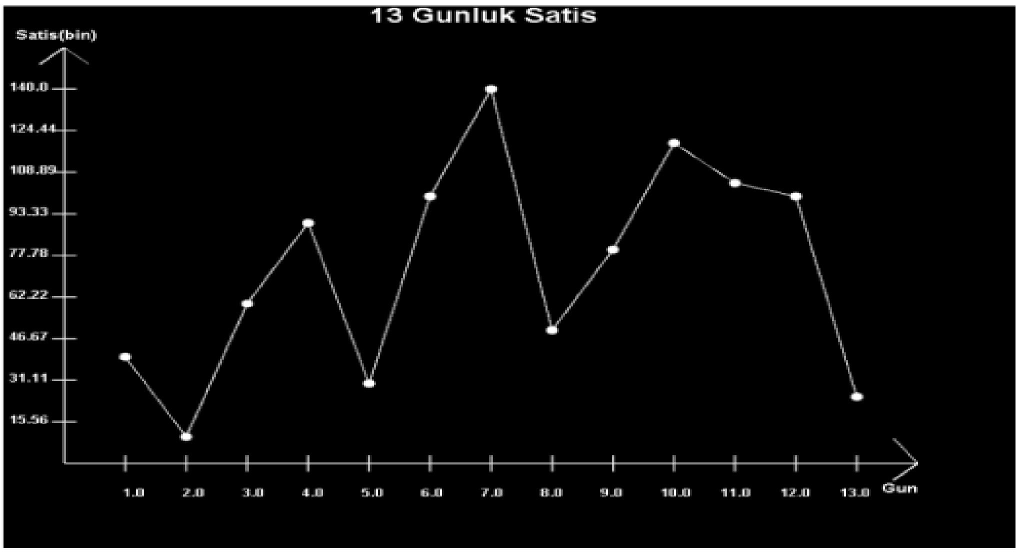


Figure 5. Line graph image for the group of time series data shown in Figure1

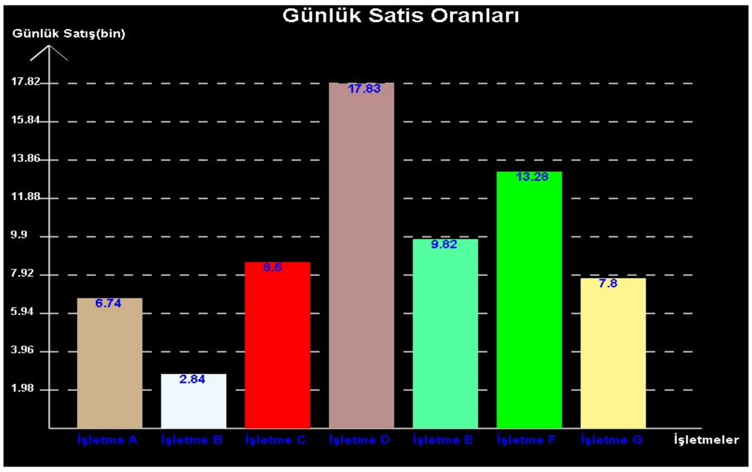


Figure 6. Bar chart image for the group of time series data shown in Figure1

## 4. CONCLUSIONS AND FUTURE WORKS

Tables and graphs enable us to show the amount of data, make comparisons between data, and show changes between the data. In this sense, we have developed the program with the convenience of making a graph of the data provided.

OpenGL library is developed for graphics and this library must be installed on the computer to run the application to be developed. The user can achieve graphics via any internet browser installed on his/her individual computer without the need for the OpenGL library.

Because the application is web service-based, it makes different applications communicate with each other quickly and securely. At the same time application developers can integrate it to their software using the WSDL.

Because the architecture is based on web services, it makes different applications to communicate with each other faster and safer. It is Java-based application, so it has multi-platform support. Both Unix/Linux and Windows users can run it. This application is developed to detect changes in instantaneous changes on the chart can be performed interactively.

Further work is required to improve to detect instant changes on the graphic. In addition, different graphic types such as three dimensional, bar, and circle graphs can be achieved.

## REFERENCES

Bertin, J. Semiology of Graphics. *The University of Wisconsin Press*, USA.

Brillinger, D.R. (1975). Time series: *Data Analysis and Theory*. New York: Holt, Ri-nehart. & Winston.

Doyle A., and Reed C. 2001. Introduction to OGC Web Services, An OGC® White Paper, 2001, http://www.opengeospatial.org (accessed 20 May 2012)

Erl T., (2005) Service-Oriented Architecture: Concepts, Technology & Design, Prentice Hall.

Havre, S., B. Hetzler, and L. Nowell. 2000. ThemeRiver: Visualizing Theme Changes over Time. In Proc. IEEE Symposium on Information Visualization 2000 (InfoVis '00). IEEE *Computer Society, Los Alamitos*, USA, 115-123.

http://www.w3schools.com/Webservices/ws_intro.asp (accessed 17 May 2012)

Inselberg, A. (1997). Multidimensional Detective, In Proc. IEEE *Symposium on Information Visualization* 1997 (InfoVis '97). 100-107.

Kotzinos D., and Chrysoulakis N. 2003. Design of GIS Web Services for Environmental Monitoring: Using Satellite Imaging to Calculate Vegetation Indices, Geographical Information Systems and Remote Sensing: *Environmental Applications*.

Lambert, J.H. (1779). *Pyrometrie*. Berlin, Germany.

MacEachren, M. (1995).*How Maps Work*. The Guilford Press, New York.

Mackinlay, J. (1986). Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*. 5 (2): 110-141.

Tufte, E.R. (1983). The Visual Display of Quantitative Information. Graphics Press, Cheshire, Connecticut, USA.

Weber, M., M. Alexa, and W. Müller. (2001). Visualizing Time-Series on Spirals. Proc. IEEE *Symposium on Information Vi-sualization* 2001 (InfoVis '01), San Diego, USA, 7-13.

van Wijk, J.J., and E. van Selow. (1999). Cluster and Calendar-based Visualization of Time Series Data. In Proc. IEEE Symposium on Information Visualization (InfoVis '99) (ed) G. Wills, D. Keim. IEEE Compu-terSociety, 4-9.

Yılmaz, A. (2003*). Zaman Serileri Analizi*. 1st Edition, İstanbul, Bıçaklar Kitabevi.