

Yamamoto, Yoshikazu; Nakano, Junji

**Working Paper**

## A time series analysis system using visual operations

SFB 373 Discussion Paper, No. 2000,32

**Provided in Cooperation with:**

Collaborative Research Center 373: Quantification and Simulation of Economic Processes,  
Humboldt University Berlin

*Suggested Citation:* Yamamoto, Yoshikazu; Nakano, Junji (2000) : A time series analysis system using visual operations, SFB 373 Discussion Paper, No. 2000,32, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin, <https://nbn-resolving.de/urn:nbn:de:kobv:11-10047389>

This Version is available at:

<https://hdl.handle.net/10419/62210>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

# A time series analysis system using visual operations

Yoshikazu Yamamoto<sup>1</sup> and Junji Nakano<sup>2</sup>

<sup>1</sup> Tokushima Bunri University, 1314-1 Shido, Okawa, Kagawa, 769-2193, JAPAN

<sup>2</sup> The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku Tokyo 106-8569, JAPAN

## Summary

We design and implement a prototype time series analysis system named TISAS for fully using modern graphical user interface technologies. An object oriented approach is adopted to represent data, statistics and models as instance objects and to visualize them by icons on the screen arranged as a tree for recording the analysis process clearly. When we point an icon which represents an object we want to handle, a pop-up menu, whose items are statistical procedures available for the object, appears. The operation that we select an appropriate procedure from the menu, produces a new object as a result and an icon on the screen to record the history of the analysis. As we can perform almost all statistical analysis on a tree of icons visually and directly following situation sensitive menus by simple mouse operations, GUI of TISAS is easy to use for many users, and is useful for preventing trivial misuse.

**Keywords:** Graphical user interface, Object oriented approach, Situation sensitive menu, TIMSAC package

## 1 Introduction

Recently, many software products are operated by graphical user interfaces (GUIs), in which users give their commands by clicking icons or menus on the screen by pointing devices, and can see beautiful graphics as a result (Shneiderman 1992). GUIs are easier to use, especially for beginners, than traditional character user interfaces (CUIs), in which users give their commands by typing keyboard or reading program files, and where the software products display their results by series of ASCII characters.

As GUI needs far more computer resources and programming skills than CUI (Olsen 1992), CUIs were mainly used before. Nowadays, however, even personal computers are equipped sufficiently for realizing GUI by the rapid progress of the computer environments. CPU speed and hard disk spaces of personal computers today are almost the same as “high-end workstations” of a few years ago. Software technologies for supporting GUI also have been developed. Popular operating systems, for example, UNIX with X Window System or Microsoft Windows, have GUI supporting functions as their fundamental facilities. Many well-designed GUI developing tools such as Toolkit on Tcl (Ousterhout 1994) are available cheaply or even for free.

In these situations, almost all statistical systems have their own GUIs. However, if we investigate these GUIs carefully, we notice that some of them do not use the full advantages of GUIs. Many popular statistical systems have been used for long time and have their own command languages, and have been operated by them through CUIs for years after they appeared. In many cases, their GUIs are used as the replacement of old CUIs, i.e., GUIs are used for generating original CUI commands. This sometimes makes their menu structures complicated or not well organized and users have difficulties for finding appropriate statistical procedures in a menu tree. Some menus may contain inappropriate statistical procedures for the data.

The recent spread of computers in many commercial and industrial activities promote gathering large amounts of data automatically. At the same time, cheap and powerful personal computers enable executing complicated and time consuming statistical computations at home. Therefore, the number of people who want to carry out statistical data analysis has increased greatly. As many of them have less knowledge of statistics and/or computer science, they need statistical software which is easier to use than before. They prefer GUIs to CUIs, and they need GUIs which are sophisticated enough for their requirements. It is clear that statistical systems should be designed from the beginning considering the use of GUI for utilizing full abilities of GUI. Al-

though it was difficult for statisticians to realize such a GUI before because of the programming complexity, current advanced programming environments make it possible for statisticians to do the GUI programming by themselves.

We design and implement TISAS (TIme Series Analysis Supporting system) to show an example of GUI for statistical works. We propose that data, statistics and models are expressed as icons on the screen, arranged for recording the analysis process clearly. By this design, we can grasp easily what we have done in our statistical analysis. Operations can be proceeded by selecting items of pop-up menus which appear when we specify icons. The menus are situation sensitive, i.e., states of items vary depending on the past operations, and are useful to indicate appropriate procedures for specified data and to prevent inappropriate use of statistical techniques.

In section 2, we investigate user interface of statistical systems. We describe basic design principles of TISAS GUI in section 3. Section 4 describes how TISAS is operated in a real situation. Implementation details are explained in section 5.

## 2 User interfaces for statistical systems

Until a few years ago, a lot of programs were operated through CUI. CUI uses characters for communicating information between users and the application program. Users give their orders to the application by commands or programs. The application returns calculation results as plain texts on the screen or in the file. It means that all the history of the analysis remains as a series of commands or programs in text format. Professional users or frequent users who are familiar with the command language, can give their directions easily by it. Any operation can be expressed as a program. For non-frequent users, however, CUI requires serious study for mastering the command language.

GUI uses graphic objects for expressing information. Users give their directions mainly by pointing icons and items in menus by mouse click. Outputs from the application are graphics or tables. This approach is easy to use for non-frequent users, because they do not need to memorize the commands. Recording the history of the analysis, however, usually is not an easy task. Professional users sometimes feel frustration for mouse operations in many pop-up and/or pull-down menus of several levels. Bundling related procedures together is also difficult in GUI.

As CUI and GUI have their merits and demerits, statistical applications had better have both. In many statistical software products, their CUI and GUI are realized by the “Composite user interface” approach (Liu, Chan, Montgomery & Muller 1995), in which GUI parts are programmed independently

with its system parts as much as possible. Statistical systems, such as SAS, SCA (Liu et al. 1995) and SPSS, had CUIs when they were first made. After environments for supporting GUI became popular, they built their new GUI on their old CUI. This approach has some merits. Professional users can continue to use the familiar command language and the programs written before are still available. The inclusions of new bugs into the main calculation programs on the way of GUI programming can be avoided. For these reasons, “Composite user interface” approach is mainly adopted by modern statistical systems, such as S (Chambers & Hastie 1992) and XploRe (Härdle, Klinke & Müller 1999).

In this approach, as GUI is a kind of a replacement of the CUI commands, it is difficult to have completely new functions. Besides, GUI is strongly limited by the syntax of original language, which reflected computer environments of the time when the application was made.

In order to have novel GUI, which is required by many statistically naive users, we have to design a system considering the use of GUI from the beginning and using effectively current computer technologies. An example of such attempts is the project STABLE (Morgan & Craig 1998), in which new visual programming ability is implemented. We design our system TISAS for visualizing analysis process as clearly as possible.

### 3 Basic design principles

The main purpose of data analysis is to make clear the mechanism of the data generating process by fitting appropriate models to the data. We usually have to transform data to find simpler behavior, calculate some descriptive statistics to search for clues of suitable models. When we choose a promising model and fit it to the data, we have to diagnose the result for checking the appropriateness of the model. This process is repeated until we have a satisfactory model. To support these processes, it is important to show what the user did, i.e., the history of the analysis, clearly and intuitively. GUIs of many statistical systems, however, lack this ability. For example, lots of graphics windows which are not well organized are produced as analysis proceeds and the user has difficulty searching for a particular graph drawn before.

We propose expressing all the history of the analysis visually on the screen. When we read data from a file, the data is represented as a rectangular bisque icon in a window. Statistics calculated from this data are represented by a rectangular green icon located inside the original data. Numerical values or graphs of data are shown in other windows, which can be easily specified from the icons. Fitted models for the data are also represented as hexagonal

reddish icons in the original objects. When we modify data, for example, by log transformation or taking a sub-series, we have a new data icon outside the original data icon. The two icons are connected by a line, which expresses that the new data is derived from the original one. Then we focus on the new data icon and carry out some statistical procedures for it. As analysis proceeds, generated data icons form a tree structure whose root is the original data read from a file. This tree shows the history of the analysis directly. An example of the TISAS session is shown in Figure 1.

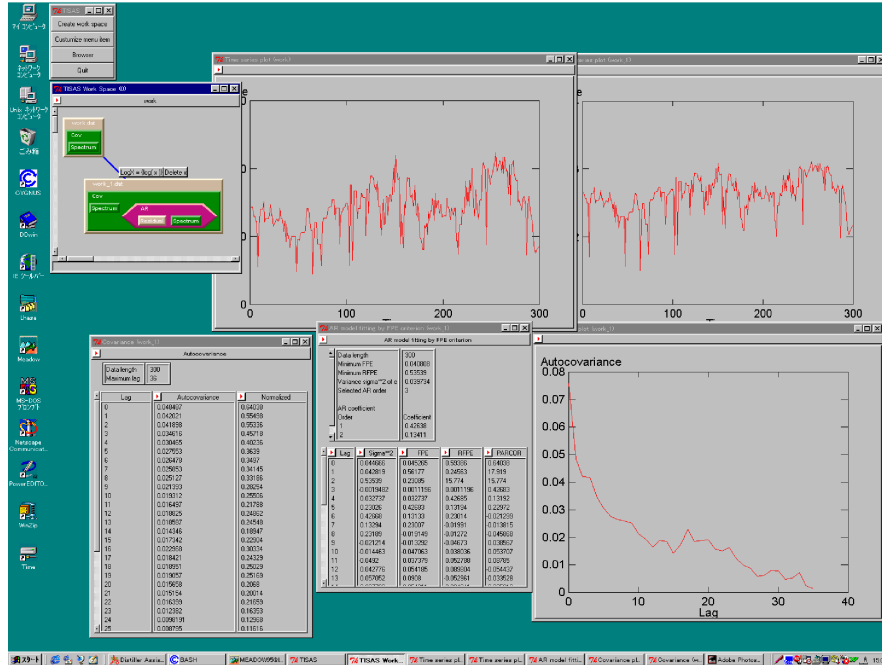


Figure 1: An example of the TISAS session

As we stated, TISAS operations are data based, i.e., we first specify data, then run a statistical procedure for it. When we point at a data icon by the mouse, available data handling procedures are listed as a pop-up menu. We choose a menu item and execute the procedure, then calculated statistics or fitted models appear as icons, which also have pop-up menus of available statistical procedures. These pop-up menus are situation sensitive, in other words, the menu items reflect analysis situations, for example, some calculation are not allowed by shadowing the menu item once it was performed. Each data icon can display menu items of appropriate procedures for the object. This is useful so as not to show unnecessary menus and to prevent using clearly

inappropriate procedures.

Many statistical systems, however, are designed following a procedure based approach, which means that we first specify a statistical procedure, then feed data to it. It is true that all the statistical techniques are expressed as algorithms, and can be implemented as functions or subroutines. If we know the analysis methods for particular data and make a program to work for it, we can repeat the same analysis for similar data easily by applying it.

On the other hand, at the beginning of the data analysis, we have to apply many statistical procedures to the data. Their results should be arranged properly for users not to lose their way in many calculated statistics or graphs. We can arrange these objects easily by a data based approach, because all calculated statistics or graphs are strongly connected to the data.

This user interface is well realized by the object oriented programming. Objects are basic elements which store both data and functions (or methods) in them. Objects send messages to other objects, and the object which receives a message performs action for changing its inner state and sending new messages to other objects. In object oriented technologies, class objects are a kind of type declaration, and instance objects are realizations of class objects. This framework is known to be useful for building large application programs, especially for building window based GUI programs. It is also known to be useful to arrange classified things or ideas by an inheritance mechanism among class objects.

In TISAS, all icons express instance objects and their pop-up menus are lists of methods which are defined in their class objects. Selecting a menu item means to send the message to the instance object which is represented as an icon. Statistical class objects in TISAS consist of data, statistics and models. Contents of each icon can be shown by separate windows. One icon may have more than one such windows, for example, sample autocovariance function object has two windows, namely, a table window and a graph window (See Figure 1). As the icon and such windows represent the same instance object, we treat them in the same way, i.e., the same menu appears when we specify them.

We try to make keyboard operations as few as possible. For example, when a user needs to indicate a variable name, TISAS will show the list of the existing variable to select. This prevents mistyping of their names. Only a new variable name needs to be typed.

## 4 Operating TISAS

When we start TISAS, two windows appear. One is for controlling the TISAS process, and another is a work window on which the history of the analysis is shown. As work windows themselves are instance objects, a user can send messages to them. Available messages for them are `Load data` and `Load log`. `Load data` is for reading new data from a file and shows a data icon on the work window. `Load log` is to read a log file of the former analysis and to go back to the last stage of it. TISAS has its own object oriented language, in which all the messages sent by icon operations during the session are kept internally. When we close our sessions, TISAS asks whether it saves the history of the analysis written in the language to a log file or not.

If we specify an icon on the work window or the upper left triangle red button of a table or a graph window by mouse, a pop-up menu appears. Some menu items may be shadowed and not available, because, for example, it was performed before. Choosing a menu item is to send that message to the object. In case additional information or arguments are required, a new window for inputs appears. In many cases, input areas are filled by appropriate default values.

As was described, when we send a message to an object and a new object is generated, a new icon for the new object appears in the work window. Icons for derived information, such as calculated statistics or fitted models, are placed inside the icon for the original object to which the results of messages depend. When an icon becomes large because a lot of derived icons are located inside, we can collapse it to conceal details, and can expand it to the original shape.

Generating new data by data transformations produces a new icon which is connected to the original data icon by line. The executed transformation equation is displayed on the line. We can drag icons and change the locations of them for easiness to see.

Equations for data transformations are also defined by the icon handling. Selecting `Transform` item on the pop-up menu generates a window for defining an equation. As there are icons for basic operations, we can build a new equation by combining them. Generated equation can be stored and can be added as a method of the class, and automatically, is added on the menu.

## 5 Details of implementation

TISAS is written in two computer languages, Tcl/Tk and C++. The main system and GUI of TISAS are programmed in Tcl/Tk language (Ousterhout 1994). Tcl (Tool Command Language) is a scripting language and Tk (Tool



Sample autocovariance function (Univariate and Multivariate)  
 Spectral density estimation by sample autocovariance function  
     (Univariate and Multivariate)  
 AR model fitting by FPE criterion (Univariate and Multivariate)  
 AR model fitting by AIC criterion (Univariate and Multivariate)  
 Spectral density estimation by AR model fitting (Univariate and Multivariate)  
 ARMA model fitting by the maximum likelihood method (Univariate)  
 Noise contribution of AR model (Multivariate)

Table 1: Available procedures in TISAS

Kit) is a standard add-on to Tcl that provides commands for GUI programming. Tcl/Tk runs on UNIX, Windows and Macintosh platforms, and is available freely.

We choose Tcl/Tk mainly due to the ability of creating GUI parts so quickly and easily (Nakano 1998). Although Tcl/Tk was not designed based on an object oriented approach, it is not difficult to realize an object oriented framework on it because of its simple and programmable syntax.

As Tcl/Tk is a scripting language, it is not good at numerical calculations. Therefore, we call C++ version of TIMSAC programs for calculating statistics from the Tcl/Tk main program (Yamamoto, Nakano & Tamura 1995). The original TIMSAC (TIME Series Analysis and Control) package (Akaike & Nakagawa 1988) was developed in Fortran language, and has been used widely. We implement basic and standard linear time series analysis procedures listed in Table 1.

TISAS also has basic data handling procedures such as data transformation, and procedures for prediction and simulation by ARMA models. It can draw graphs of time series (including residuals and predicted values), autocovariances, spectral densities, and noise contributions.

TISAS runs on almost all UNIX workstations and Microsoft Windows. For UNIX system, in case GNU C++ compiler system is available, Tcl/Tk and TISAS can be easily compiled. For Microsoft Windows, as we prepare an executable setup file, the installation is straightforward.

## 6 Conclusion

The importance of GUI for operating software products is clearly understood in recent computer environments. GUI programming becomes much easier than before because of recent powerful hardware and well-designed GUI supporting software. We are in the stage that GUIs of statistical systems should be improved entirely in much friendly and intuitive ways, especially for in-

creasing the numbers of statistically and computationally naive users.

In this article, we propose an example of a new GUI for a time series analysis supporting system. We show that an object oriented approach is useful for realizing visual and intuitive GUI for a statistical system. We also propose to use a display of a tree which expresses the history of the analysis, and focus firstly on data, not on procedures. Although our system is limited to time series analysis, these design principles will be applicable to other general purpose statistical systems.

## References

- Akaike, H. & Nakagawa, T. (1988), *Statistical Analysis and Control of Dynamic Systems*, Kluwer Academic Publishers.
- Chambers, J. M. & Hastie, T. J. (1992), *Statistical Models in S*, Wadsworth.
- Härdle, W., Klinke, S. & Müller, M. (1999), *XploRe - Learning Guide*, Springer.
- Liu L. -M., Chan K. -K., Montgomery A. L. & Muller M. E. (1995), A system-independent graphical user interface for statistical software, *Computational Statistics & Data Analysis*, **19**(1), 23–44.
- Morgan, G. & Craig, P. (1998), A visual future for statistical computing, in 'COMPSTAT 1998 Proceedings in Computational Statistics' (ed. Payne, R. and Green, P.), 389–394. Physica-Verlag.
- Nakano, J. (1998), Graphical User Interface for Statistical Software Using Internet, in 'COMPSTAT 1998 Proceedings in Computational Statistics' (ed. Payne, R. and Green, P.), 407–412. Physica-Verlag.
- Olsen, D. R. (1992), *User Interface Management Systems: Models and Algorithms*, Morgan Kaufmann.
- Ousterhout, J. K. (1994), *Tcl and the Tk Toolkit*, Addison-Wesley.
- Shneiderman, B. (1992), *Designing the User Interface 2nd edition*, Addison-Wesley.
- Yamamoto, Y., Nakano, J. & Tamura, Y. (1995), Porting TIMSAC72 for MS-Windows to UNIX, *Proceedings of the Institute of Statistical Mathematics*, **43** (2), 285–292.