

Flash-Enabled User Interface For CAS

Keehong Song

Pusan Nation University, Pusan, Korea

www.mathematica.co.kr

Abstract

As the de facto web animation standard today, Flash has a potential to become a powerful vehicle for teaching and doing Mathematics over the Internet other than its commonly known utility as a fancy banners construction tool. This article discusses the versatility of Flash as the middleware for the mathematics engines such as CAS or Java classes in the server-side tier. The examples in this article demonstrate the Flash-enabled UI harnessing the power of Flash, which brings the sophistication of the desktop application onto the standard web application over the Internet.

Introduction

Flash technology has evolved into a powerful application development platform, adhering to today's client-server technology standards. Based on ActionScript, the versatile javascript-like object-oriented programming language, the Flash UI supports TCP/IP socket that sends and loads the external text files as well as the JPEG images on the fly to and from the mathematics engines at the backend. With the use of the supporting technology, such as Flash Remoting, a Flash movie on the web page can invoke the remote methods on the server side and consume web services, thus opening up the array of possibilities for the powerful web application. On top of that, it has many other advantages such as the virtual cross-platform compatibility, the bandwidth-efficiency, the convertibility to and from SVG, the java-bean-like reusable components, the sophisticated XML parser, and so on as visualized in figure 1, the road map of the Flash-based client-server architecture. This article uses a dedicated scaled-down web server to show how a Flash-enabled UI communicates with the server handling the text and the graphics data in utilizing the problem-solving power of CAS.

This article was written with the grant from Korea Research Foundation for the project of Object-oriented Courseware Development System.

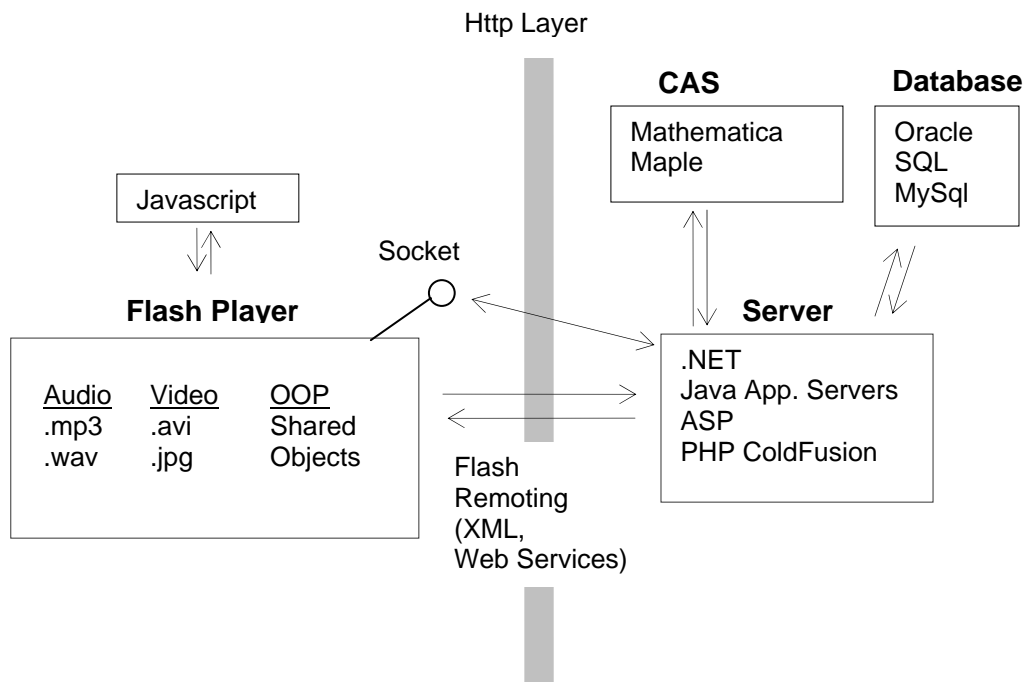


Figure 1

Design and Implementation

The versatility and flexibility of Flash as a UI development tool is a double-edged sword. Conscientious efforts need to be made to keep the complexity of the UI design in moderation, the quality the ones in this article strive to achieve. Even though, the tone of the user-interface is largely determined by the nature of the subject matter as well as the user type, the parsimony in design is an important ingredient as we often see that a potentially interesting project in no time degenerated into a dizzy eye baloney simply because "they did it because they could."

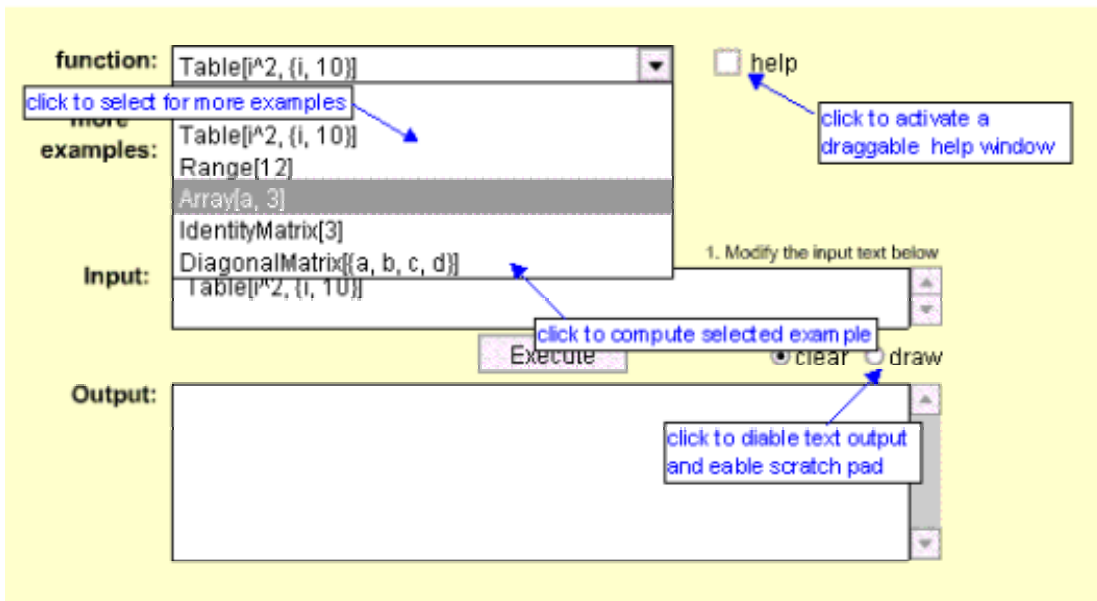
That said, the Flash development system allows the developer with ease a high level of sophistication and complexity in developing the web application. As Figure 1 indicates, the design of Flash-enabled UI requires a careful planning as the elements in the Flash development environment as they are intricately involved. For the examples in this article, a dedicated scaled-down Java web server was used in interacting with the mathematics routines on the server side through a socket. This simple scheme can be immensely improved using other technology such as .NET, Flash Remoting, and so forth, which allows invocation of the server-side methods as if they are the local flash routines.

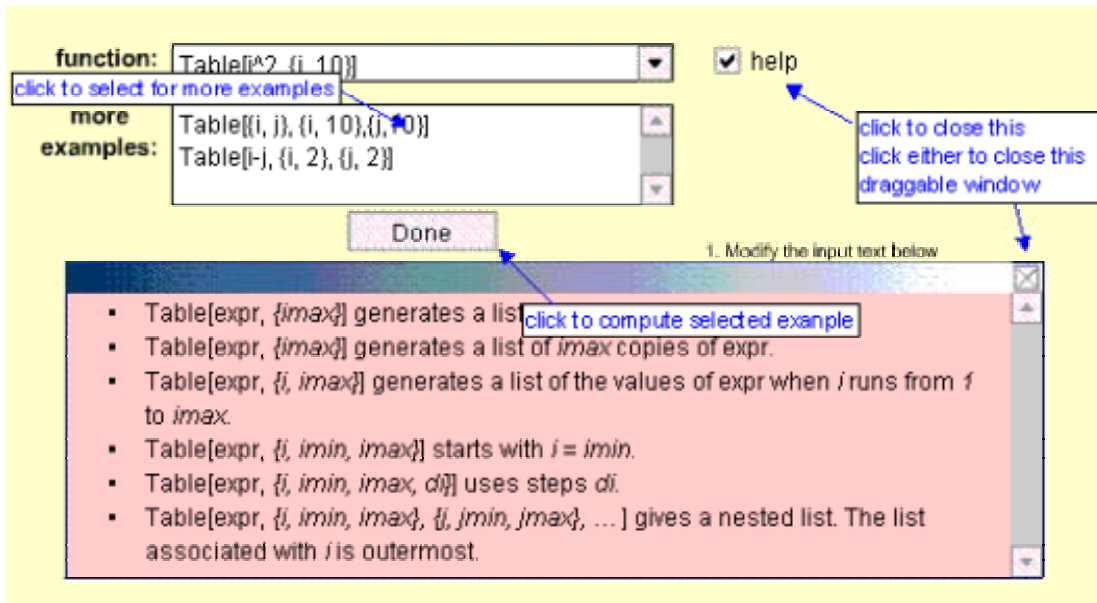
Examples of Flash-enabled User Interface

1. List Functions

The effective means of teaching the CAS program language is crucial in the class room for its success. This example intends to demonstrate how a menu can be used in teaching the Mathematica's language command set. Some of notable functions in the example are as follows:

1. Click help button to display the help files for the selected command. With the button checked, you can switch to other Mathematica functions and then the content of the help file changes accordingly.
2. Click draw button to use the screen as if it is a whiteboard in the classroom. This function is useful in a class room situation in directing the student attention around.
3. In the Input area, the user can modify the default line of command in order to practice on the command.

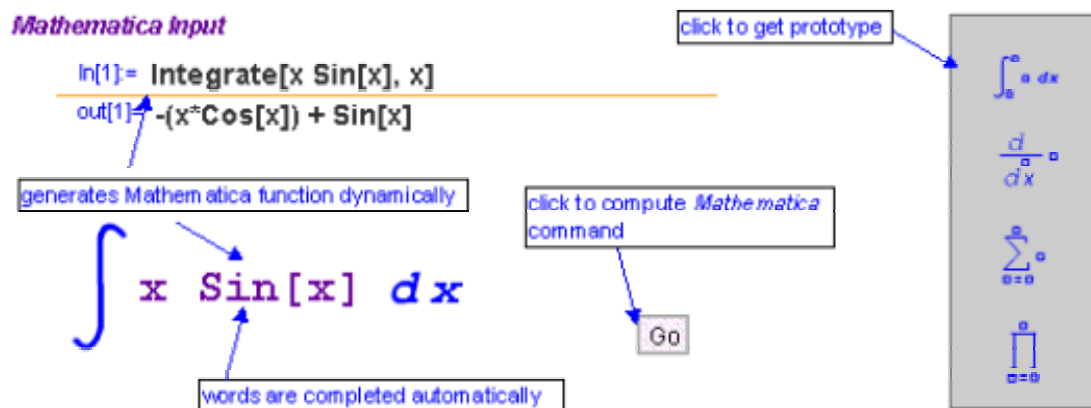




2. Palette

The palette saves the manual typing to simplify the command input. Basically, this is the example of the Mathematica Front-End simulation and some of the notable features are as follows:

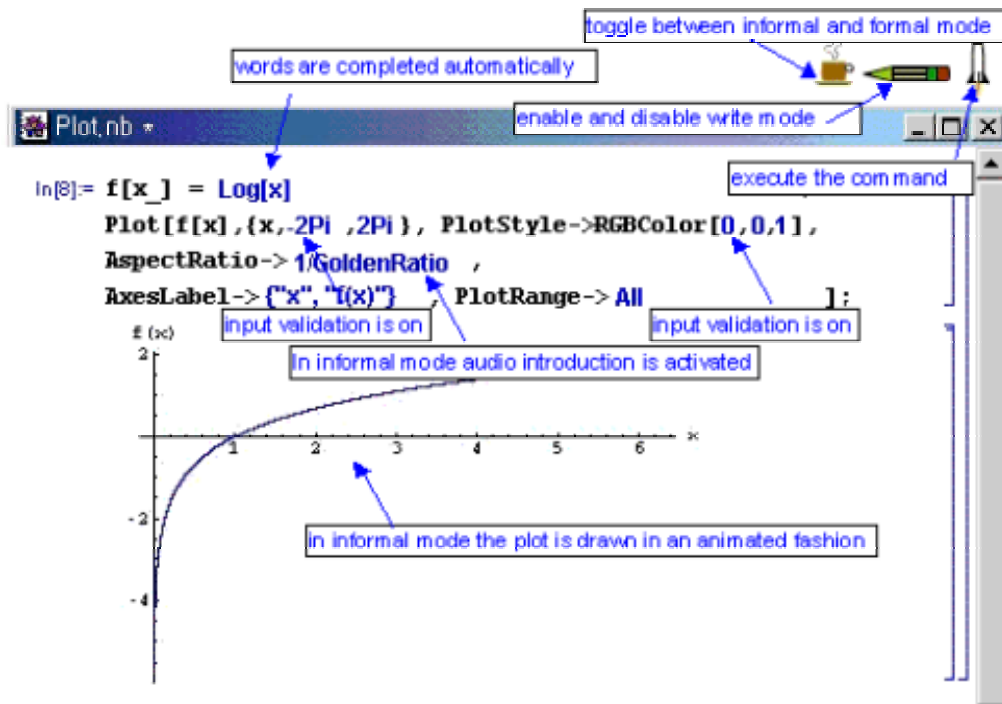
1. It has the intelligent word completion capability. For example, typing of "x" automatically yields "x Sin[x]" and "s" does "Sin[x]" and so on. Move the cursor to the intended position to change the command typed and start with the initial letter for the intended function, e.g., a letter t for "Tan[x]".
2. As the user goes along completing the input form, input area changes dynamically and display the correspondence between the conventional mathematical notation and its matching Mathematica command.
3. For the lengthy output, the scroll bar appears automatically.
4. The integrating bounds are also editable.



3. Cute Front-End

This type of tutorial intends to acquaint the user with Mathematica front-end via simulation along with a set of cute icon buttons whose primary purpose is, of course, to bring visual relief to the user in a casual manner. Some notable features in using the example are as follows:

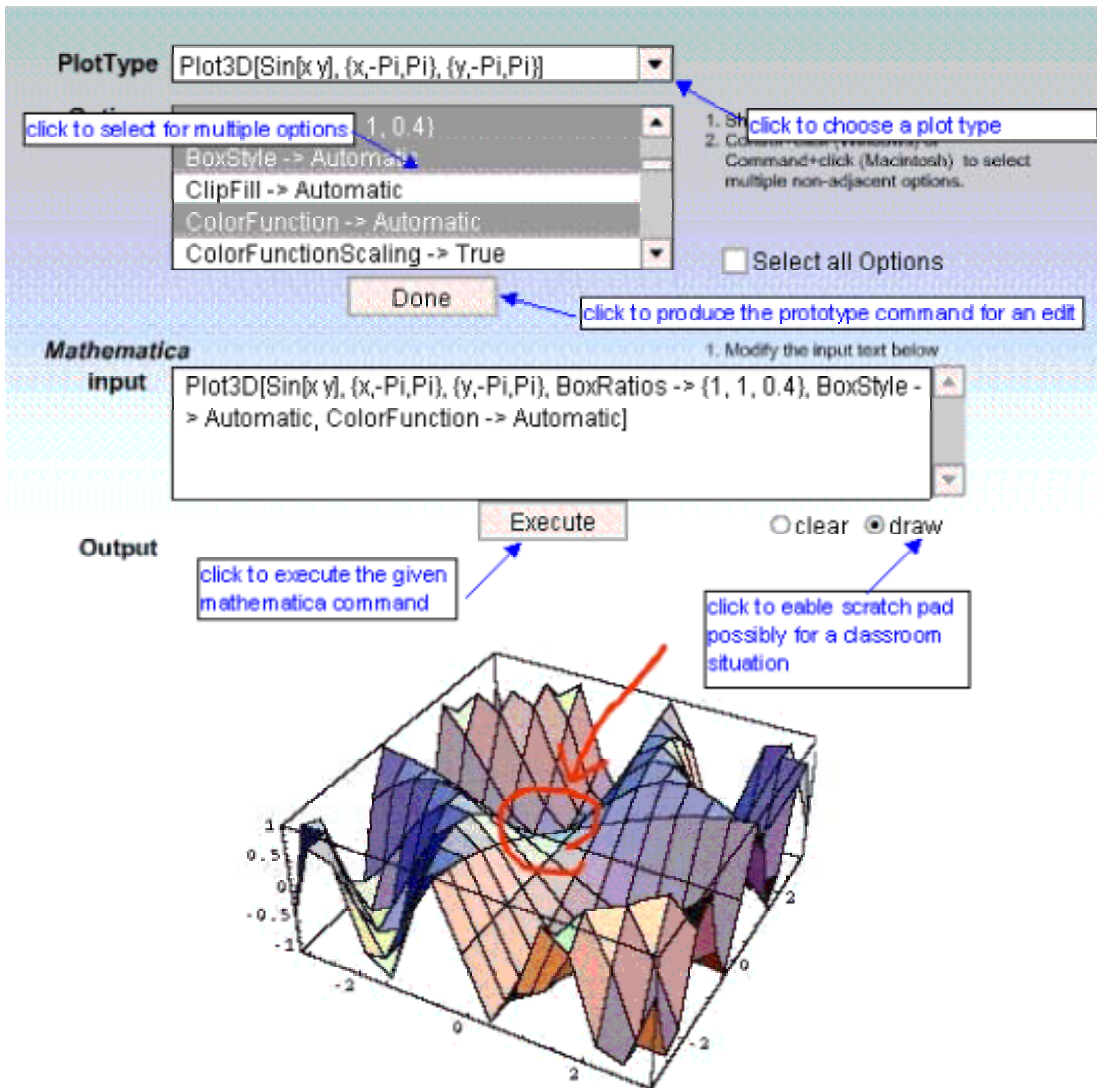
1. The "mug" icon is to toggle between informal and formal mode. In the informal mode, several changes take place. First, when mouse clicked on the colored text box, that is to say, when the focus is on the input text field, the voice introduction for the corresponding field is activated. Similarly, the voice is deactivated when the focus is off. Second, when the Mathematica command is executed, the graphic output appears in an animated fashion in order to catch the user's attention. In this mode, when "rocket" icon is clicked for Mathematica function execution (that is, when Mathematica command is executed), the audio representation of the action plays, which is done only once before it becomes a nuisance after the first encounter.
2. Mathematica functions are getting completed in an intelligent manner in the order of frequency in the ordinary usage.
3. The input validation is also activated.



4. Plot Functions

This example packs all of the Mathematica plot functions in a single combo box so that the user knows what types of plot functions and its options are available. Here is how to use the example application.

1. Multiple options are selected using the standard key combination, which afterwards could be edited in a separate text field.
2. Once the plot is drawn, the screen can turn into a scratchpad, which could be useful for the instructor in the classroom.



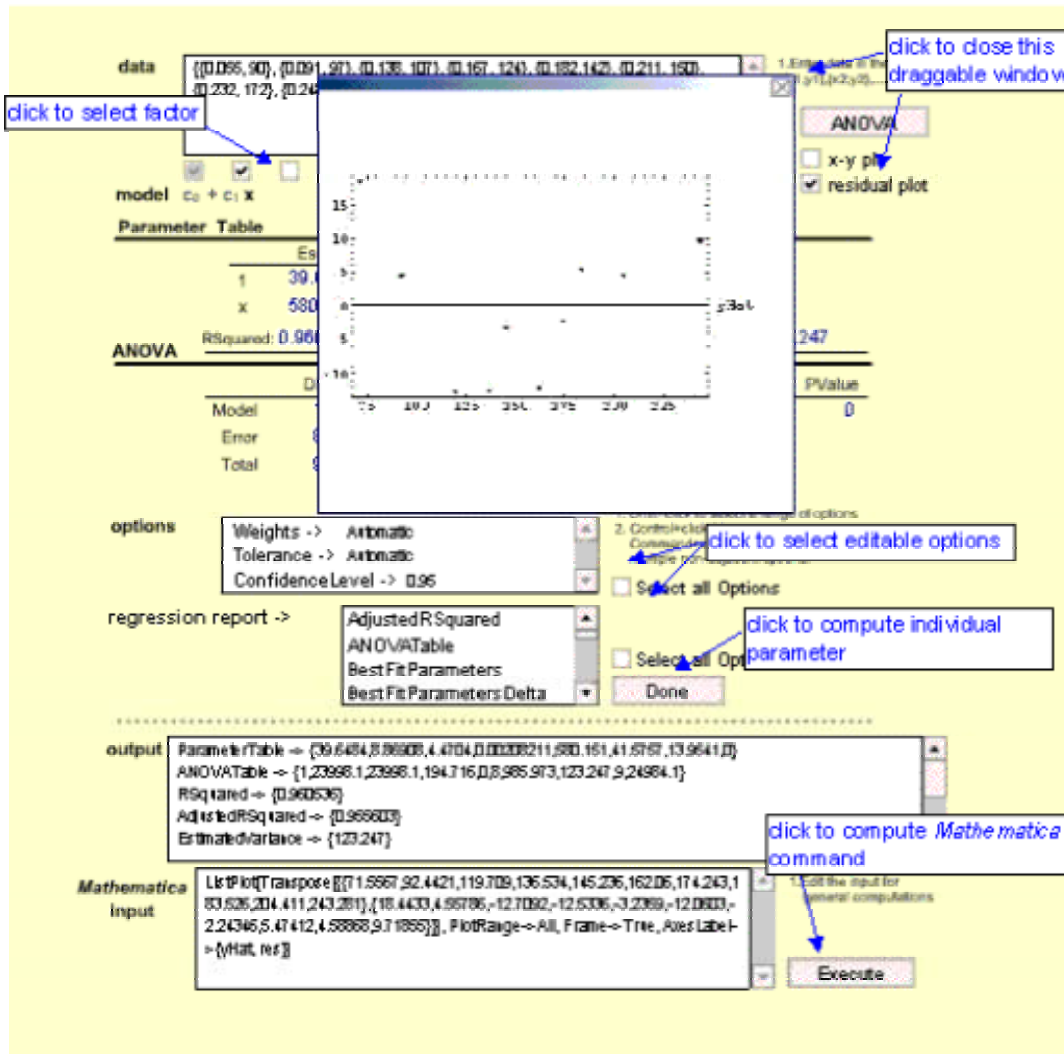
5. Mathematical content - Linear Regression

In most cases, the mathematical concepts tend to be more complicated and subtle to communicate than a programming language command set. For that reason, a web application UI often needs to be more sophisticated than just html pages with a menu. In fact, Flash-enabled UI excels in this category as Flash makes it possible to enhance the complexity of the web application at the level of the desktop application easily. As an example, with the Flash-enabled UI a simple Mathematica Add-on package such as Linear Regression can turn into an intelligent statistics package equipped with functionality only CAS can perform.

The Flash-enabled UI guides the user to plot the dataset, to select the model, to perform the regression analysis, and finally to validate the statistical model with a residual plot. Then more

specialized statistical analysis could be made when needed afterwards.

In helping the user, the error messages are intended for several likely situations. For example, an attempt to do the residual plot without performing the regression analysis prompts an appropriate error message. From this point onwards, many different types of variations and improvements are possible, notably by adding multimedia functionality such as voice and animation.



Conclusion

Flash is loaded with many powerful features for building UI when the mathematics engine is running on the server side. Now, as a downside one may point out the fact that Flash development is a proprietary solution, but practically this is insignificant as there are many open source projects matching the proprietary technology in Java Library routines and PHP. In a way, Flash is fulfilling the expectation many had on Java Applet for the multimedia applications.

References

1. Wolfram, S., The Mathematica Book, Cambridge University Press (1999)
2. Flanagan, D., Java Examples In A Nutshell, O'relly (1997)
3. Enete, N., Java Jump Start, Prentice Hall (1997)
4. Branden, H. and Wan, S., Object-Oriented Programming with ActionScript, Free Riders (2002)