Developing Computational Web Animation
- using Flash and .NET Technology

Keehong Song
Associate Professor
Department of Mathematics Education
Pusan National University, Korea
khsong@pusan.ac.kr

ABSTRACT

The rich internet application is a metaphor by the maker of Flash to denote the emerging internet technology centered on Flash, the deployment of which is to improve the user experience by way of delivering the richer, more interactive web functionality across different platform architectures and the devices. This paper discusses the ri[1]ch internet application, especially running on top of .NET framework, from the mathematical computation perspective with the emphasis on the user interface.

KEYWORDS

Flash, Flash Remoting, .NET, Web services, Mathematical computation.

INTRODUCTION

Flash, initially a plug-in for rendering animated vector graphics, has rapidly become the de facto web animation standard today. The combined technology of Flash and Flash Remoting with .NET can effortlessly enhance the functionality of the web applications up to the level found in desktop applications in terms of user interactivity, execution functionality, scalability, and manageability. The visible trend in the industry is that the traditional role of Flash technology as a front-end presentation is getting extended to the tier of event-handling logic and to that of the server view in the architecture of the Model-View-Controller (MVC) pattern. The attributes of the rich internet applications compared with the typical request-response and document-centric web pages are as follows:

1. small memory footprint with band-with efficiency.
2. extensible, event-driven object model for interactivity.
3. inter-communication with remote components and services hosted on standard servers such as J2EE, .NET, Web services.
4. platform and device neutrality
5. stability and minimum maintenance.

So, the orchestrated quartet of the mathematical engine (i.e., computer algebra system or math

This work was supported by Korea Research Foundation Grant (KRF-2003-041-D00514)

library), Flash movie, Flash Remoting, and .NET tools with real-time server communication produces the animated visual representation of the mathematical concepts in an efficient and creative manner.

To better understand the rich internet application, especially for those who are coming from the Java technology, the virtues and liability of Flash technology approach vis-à-vis the Java counterpart needs to be discussed briefly.

Flash vs. Swing

Although Java is a mature technology as the basis for many great applications, Java for Web applications falls short in many respects. First, it taxes a burden on the developer with a steep learning curve and with a relatively large code in the middle tier. And as the more serious drawbacks, the low penetration rate of the browser plug-in and the slow loading time with the heavy-weight compiled form can be cited, all of which hurt the user experience considerably.

Category	Flash	Swing
Ease of use	√	
Separation of presentation and logic	√	
Plug-in penetration	√	
Size of compiled form	√	
RPC protocol optimization	√	
Multimedia support	√	
Aesthetic values	√	
Security	√	√
Compile time validation		√
UI component availability		√
Error/exception handling		√
Reusability	√	√
Device support	√	√
Learning curve	√	
Deployment time	√	
Size of supporting code in the middle tier	√	
Cost		√

Table 1: Comparison of Flash and Swing

The shortcomings of Swing listed in Table 1 far outweigh its merits including the sandbox security, the compile time validation, UI components availability and the like. Although Java is more popular

programming language in the academic community, Flash marks better in most categories of evaluation matrix as the web application development tool as listed in Table 1.

DEPLOYMENT AND LEARNING CURVE

The deployment is downright straightforward: a matter of copy-and-paste of the necessary files into the right directories as shown in Figure2. Only complexity, if any, is going to be the business of setting the proper permission for the virtual directories. Thus the learning curve for the developer of this technology is bound to be a mild one added by the powerful IDE in both .NET and Flash.

Most on the server side is within the syntax of the server language while the client side uses only ActionScript, the ECMA-conformed language of Flash.

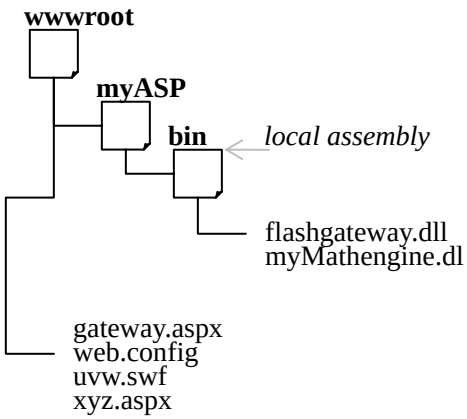

Figure 2: directory structure for the Flash Remoting deployment

Flash Remoting Infrastructure

As illustrated in Figure 1, the basic idea of Flash Remoting is to call the remote services in such a way that the client Flash application uses the data and methods by the remote interface as if they are local objects. The value object and session façade patterns are useful with Flash Remoting MX by reducing the number of remote method calls, lowering the network traffic, and response time.

The pleasing performance is due to efficient Flash Remoting protocol, which is a big leap from URL-encoded name/value pairs used in *Generator*, the previous generation client-server solution with Java class library extension methodology.

Flash Remoting protocol (AMF) is similar to SOAP, but is arguably 50% more efficiency in bandwidth as the former is packing data in binary format. But still it is firewall safe and securable via HTTPS. Once Flash Remoting is installed, with the complexity is hidden, totally zero maintenance is required of the developers with the chores such as data-type conversion, a potential source of errors, taken care of automatically.

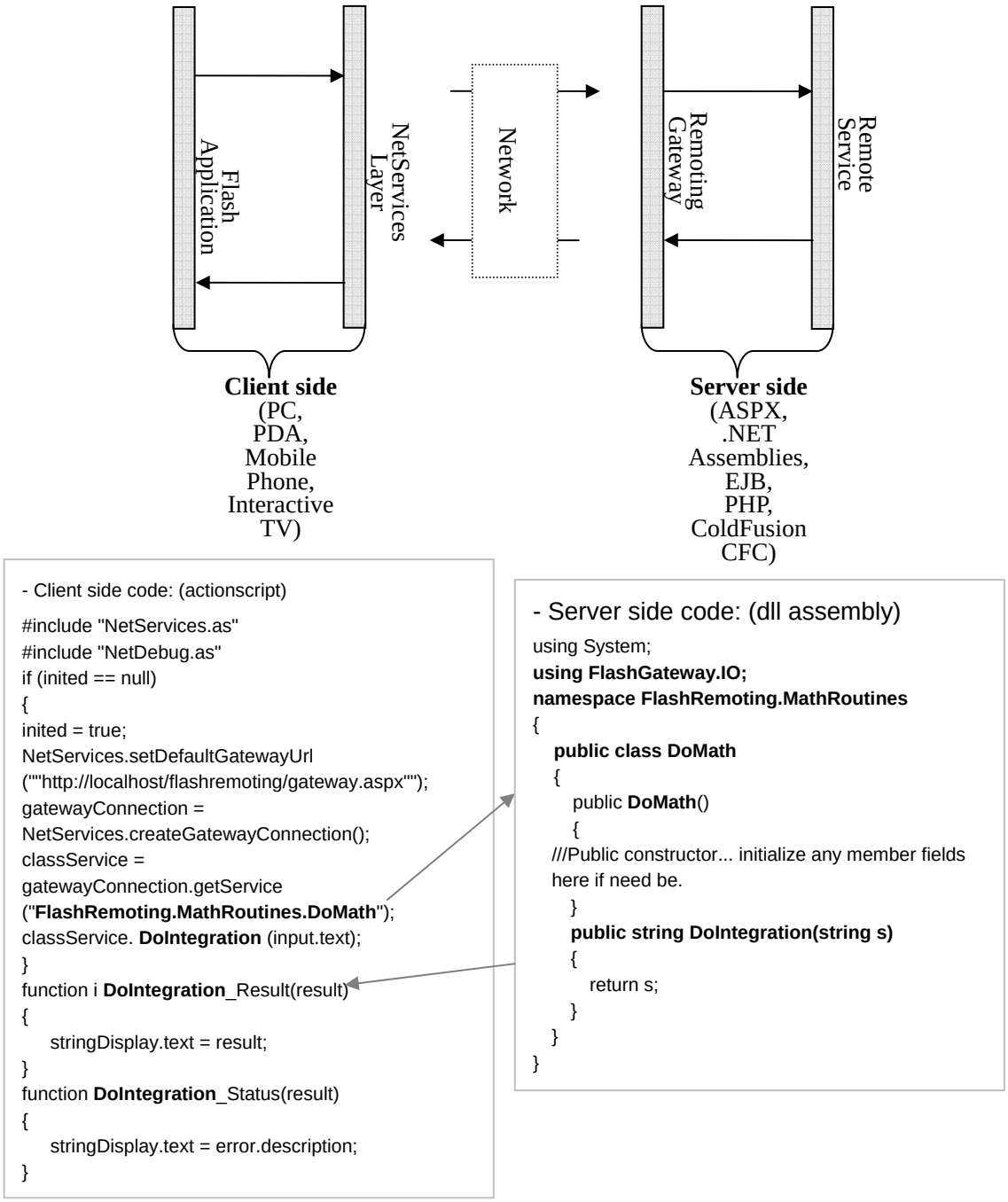

Figure 1: Flash Remoting infrastructure and the corresponding sample Actionscript and C# codes.

Flash Remoting Technology Developers and Projects

In order an emerging technology to be viable, it would be desirable that it keeps a distance from proprietary. In that respect, Flash Remoting technology seems promising as there are several freely available products in the market already employing AMF protocol. Some notable ones are listed as follows:

1. Flash remoting for PHP (http://www.amfphp.org/)
2. AMF::Perl - Flash Remoting in Perl (http://www.simonf.com/flap/)
3. OpenAMF - Java Flash Remoting (http://www.openamf.org/)
4. FlashOrb -Java version (http://www.flashorb.com/)

EXAMPLES

Example 1. *Fisheye Menu*

It is typical for a function in a scientific application to have a long list of options. For instance, in a statistical package a user gets to choose among many types of numerical output and plots. When the number of options, or menu items, exceeds the initial capacity of the menu, the user would not know how many items are hidden until the submerged part is jacked up with the scroll bar. The hidden items become a potential source of uncertainty, which could easily translate into the user anxiety. To remedy the weakness of the traditional menu with a scroll bar, the fisheye menu is an interesting alternative.

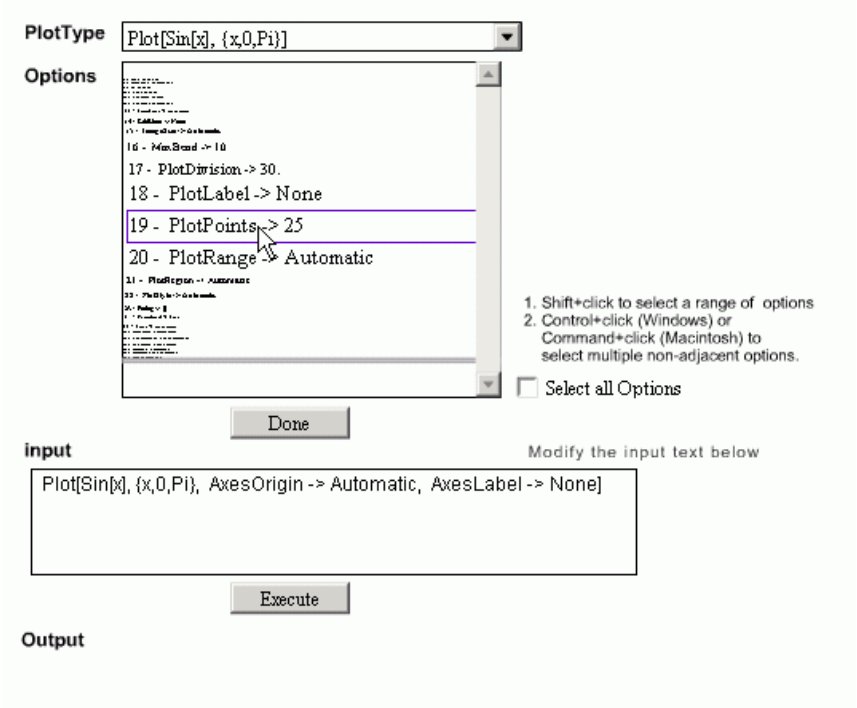

Figure 3; Fisheye menu for the Plot function

The concept of a fisheye menu in a computer interface has been around quite a while. This type of menu style tends to get positive feedbacks from the user. According to Bederson[2], "…10 users compared user preference of fisheye menus with traditional pull-down menus that use scrolling arrows, scrollbars, and hierarchies. Users preferred the fisheye menus for browsing tasks, and hierarchical menus for goal-directed tasks." The menu item of focus and its adjacent neighbors are highlighted with a larger size as the mouse pointer scrolls up and down. The fisheye menu Flash component by Samuel Wan (http://www.samuelwan.com/information/archives/000092.html) for Java version by Bederson[2] is shown in Figure 3.

Example 2. *Gram-Schmidt Process*

Certainly a picture is worth a thousand words, and the mathematical concepts are often best communicated and learned through their visual representation in an animated fashion. Flash is perfect for the type, especially when the visualization needs to be displayed over the web. Here, Flash web animation application can be made in two forms: framed-base or script-based, but the latter has the definite advantage because of the object-oriented advantages entitled.

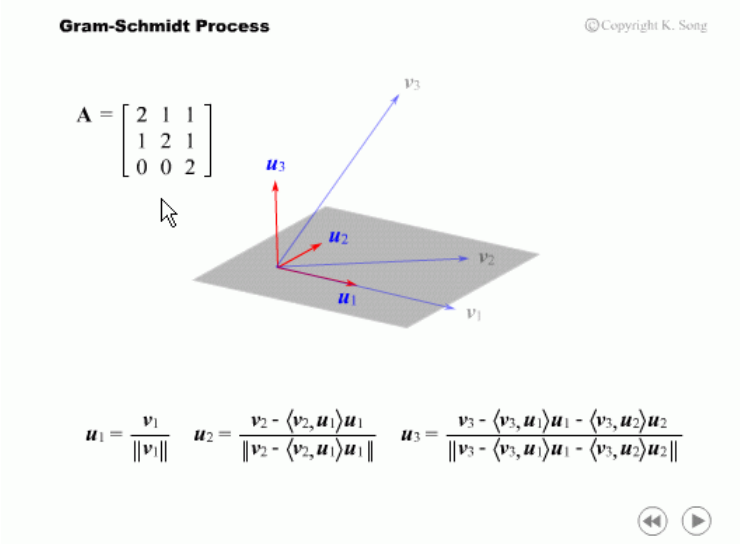

Figure 4: Gram-Schmidt Process

As an example, if the developer wants some visual effect done on a certain object, then he can simply call the right reusable methods similar to the one as shown in Table 2. Likewise, an animation movie clip can be constructed entirely from assembling the reusable scripts.

```
MovieClip.prototype.moveTo = function(x,y,vel){
    var ini_x,ini_y,distX,distY,movX,movY
    ⋮
    ⋮
}
root.createEmptyMovieClip("my_mc", 100)
my_mc.moveTo (200,200)
```

Table 2: a movie prototype example

CONCLUSION

Flash has clearly become the de facto web animation standard while Java web application has not lived up to the initial enthusiasm. NET platform is so powerful and has been generating such a strong momentum these days that .NET technology may eventually replace Flash itself with such standard as XAML, the Microsoft's Extensible Application Markup Language (XAML) specification for creating client user interfaces in the upcoming "Longhorn" version of Windows. No matter how the recent technology takes its shape in the next generation, the basic structure for the scientific web

animation is bound to be similar to the one discussed in this paper.

As a final note, the rich internet client and server applications have evolved primarily around the business community while its tremendous potential for the scientific computation remains virtually untapped. This paper purports to highlight this powerful web technology that can provide an intuitive and unique solution for various types of scientific and engineering computing needs.

REFERENCES

1. Aberdeen Group, Inc., Flash Remoting MX: A Responsive Client-Server Architecture for the Web, December, 2002.
2. Bederson, B. B. Fisheye Menus. In Proceedings of ACM Conference on User Interface Software and Technology (UIST 2000),” pp. 217-226, ACM Press, November 2000.
3. Lott, J., Complete Flash Remoting MX, Wiley Publishing, Inc. 2003
4. Macromedia white paper, Developing Rich Internet Application with Macromedia MX2004, August, 2003.
5. Macromedia, Using Flash Remoting, 2002.
6. Muck, T, Flash Remoting:The Definitive Guide, O’Reilly and Associates, 2003.