J. Korea Soc. Math. Educ. Ser. B: Pure Appl. Math. ISSN 1226-0657
Volume 14, Number 2 (May 2007), Pages 85–90

RICH INTERNET MATHEMATICAL SOLUTION SYSTEM

KEEHONG SONG* AND TAI SUNG SONG**

ABSTRACT. Arguably the most important internet technology of today is RIA (rich internet application), whose prime advantage is in improving the user experience on the web. However, the scope and variety of the RIA technology demand an eclectic guideline to build the mathematical services such as mathematical text, graphs, and animations. This paper sorts out and simplifies the complicated roadmap of the RIA technologies by presenting three design patterns for exposing the functionality of the CAS (computer algebra system) with the web browser as the user interface platform.

1. INTRODUCTION

Over the period of less than two years, there are already tens of open source frameworks under the category. By all indications the technological thrust of RIA is not going to be just a short-lived hype as in the case of Java applet, but a legitimate standard technology to stay for the future as manifested in such popular web application as Google Maps. For the mathematical community, the much-hyped technology means a rich set of technologies to leverage to provide the computational services over the web. However, the variety of tools carries both positive and negative connotation. The rich set of choices is a positive sign in that we have many technologies to control to grind out the mathematical solutions that are intuitive and usable. At the same time, that is also negative in that the variety makes the decision making process difficult as well as error-prone.

Combined with Flash with Ajax (Asynchronous Javascript And XML), which is based on Javascript, CSS (Cascading Style Sheets), DOM (Document Object Model) and XMLHTTP Request, the RIA technology purports to improve the user experience on the web that ideally measures up to the level of the desktop application

Received by the editors December 18, 2006 and, in revised form May 2, 2007.
2000 *Mathematics Subject Classification.* 68M10, 68U35.
Key words and phrases. CAS, ActiveX, MVC, Ajax, RIA, Mathematica, Adobe Flash.
This work was supported by Pusan National University Research Grant.

([2], [3], [6], [7]). Needless to say, the internet-based mathematical solution system can benefit from the RIA technologies and their potential is studied in the areas of mathematical texts, graphs, and animations.

This paper discusses three models or patterns, of the system using the web browser-based user interface for CAS. The base model is the simple one that processes the MathML, SVG, and bitmap data stream from CAS via a broker program, e.g. ActiveX control, NET control, or web browser add-ons. Then this model can be enhanced in such a way that CAS can use Java (or .NET) runtime to create a simple on-demand web server that handles such special tasks as generating JPEG on the fly. Further enhancing these offline patterns, the third model is about building the online web application on the platform of Ajax and RIA.

2. Browser-based Offline Mathematical Solution System

The browser-based offline web application in essence refers to the desktop application with the user interface developed with the web browser. The user interface for CAS based on the browser has several advantages over standard desktop user interface application.

More often than not, to build a web page is much easier task than to develop the comparable desktop application. Using standardized tags and scripts, the web browser-based user interface readily supports audio, video, and web animation contents. The web browser application is also easy to maintain as it requires little or no installation. Since the application is based on the web browser, it can easily be converted into the online version. The core strengths of the browser-based desktop application can be summarized as follows:

(1) Low development and maintenance cost.
(2) Platform independence.
(3) Low learning curve for both the developers and the end users.

2.1. Base Model

In the base model, CAS communicates with the web components via the broker program which transports the data stream in the standard form of plain text, MathML, SVG, and/or bitmap. On the side of the web browser, CSS controls the overall look and the Flash movies represent the visual effects, while Javascript orchestrates the incoming and outgoing data stream among the web components.

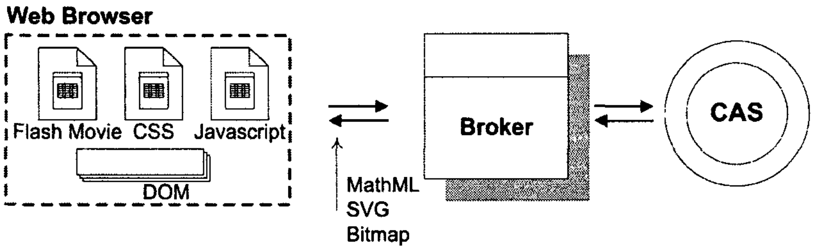

Figure 1.

It harnesses the DOM API to represents the data from the controller in a seamless manner without reloading of the entire page, creating the look and feel of desktop application. Incidentally, that usability feature is at the heart of Ajax application that will be discussed later in this paper. As in Figure 1, this model naturally applies the MVC (Model-View-Controller) design pattern, separating the presentation from the logic.

2.2. On-demand CAS Web Server Model

As an extension of the base model discussed above, this model brings a couple of extra features of CAS into play. They are simply both the web server module purely written in CAS script and the CAS library extension (see Figure 2).

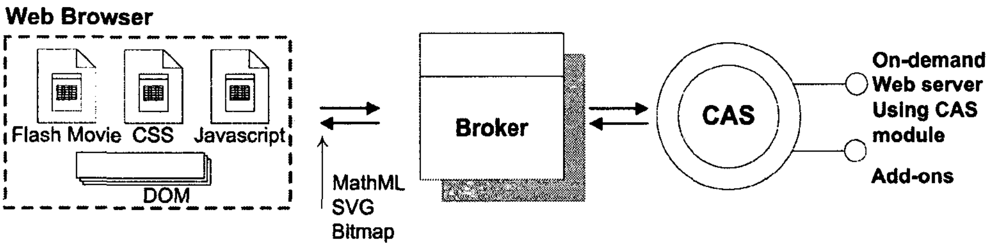

Figure 2.

To illustrate the roles of these extras, let's consider the typical situation where the Flash movie needs to load the JPEG files generated from CAS dynamically. The standard way of handling the task would be to call the local web server with localhost domain. We can obviously run the dedicated standard web server such as IIS or Apache, but for the CAS with the Java Runtime (or .NET) extension, the chore of preparing the independent web server installation could become unnecessary. Instead, we can create a simple on-demand web server entirely written using CAS script to do the job. A sketch of creating web server for the case of Mathematica would be as follows [1]:

```
(* create a new Java ServerSocket object *)
$ServerSocket = JavaNew[["java.net.ServerSocket", 8088];
(* create a buffer using byte array *)
JavaNew["[B", 2000]
(* get an input stream and put in the buffer *)
$ServerSocket@accept[]@getInputStream[]
(* write an output stream *)
```

Often CAS module extension ("add-on" in the jargon of Mathematica [8]) is needed for Flash movie to consume. To illustrate the point, suppose we want to find the triangulation of n-polygon. In a typical triangulation problem, the size of the graphical output becomes easily too big to be practical for display and at the same time we want to have a control over each triangulation for further manipulation. For the space and control issues, a Flash movie provides a workable solution as shown in Figure 3 ([4], [5]).

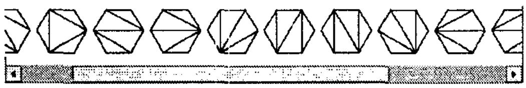

Figure 3.

In that case, it is important to implement using CAS script rather than Flash Actionscript using the MVC (Model-View-Controller) design pattern to separate the logic from the user interface.

3. Online Mathematical Solution System

One fundamental difference between the previous models and this one is the fact that this is an online application. To be more specific, it is based on RIA, a combination of various technological platforms such as Ajax, Flash, or Java (see Figure 4). In particular, Ajax is of interest in that it enables the flowing and intuitive interaction without reloading pages. However, despite the fundamental architectural difference, there is not much visual difference between the offline and online application from the user's perspective.

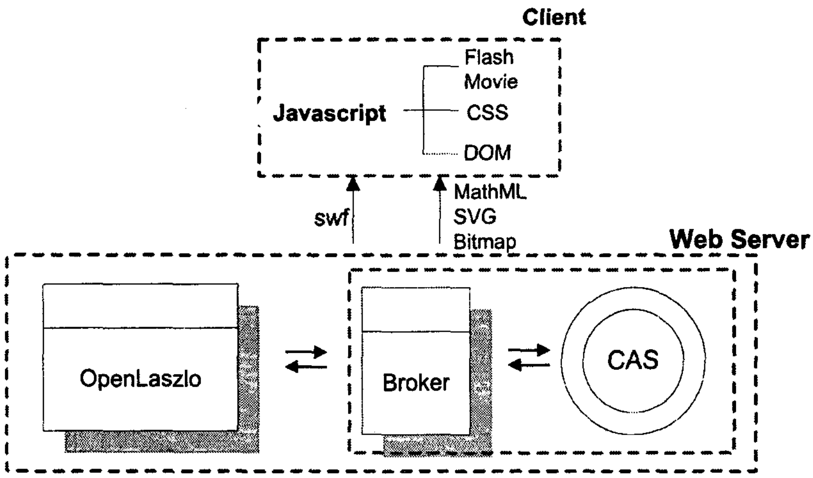

Figure 4.

The prevalent RIA technology in use today has few distinctive characteristics, namely that they are script and XML-based as exemplified by the tools such as OpenLaszlo, Adobe's Flex., Microsoft's XAML and the like. For example, OpenLaszlo can be configured to play an important role for delivering the rich mathematical services. The OpenLaszlo platform consists of the XML-based LZX programming language and a Java servlet, called OpenLaszlo Server. As a Java servelet, the OpenLaszlo server dynamically compiles LZX applications to generate Flash files for mathematical graphs and animations. It can work with other generic web servers such as PHP as well. On the client side, the web browser is sliced into several interoperable divisions so that each section holds different types of web objects such as SWF, bitmap, MathML, SVG in processing mathematical solution services. OpenLaszlo can also generate independent Flash application called SOLO for offline use without the use of commercial IDE. For both offline and online web-based applications, we suggest some pointers in the CAS design and implementation from the user interface point of view.

(1) CAS should be able to call Java (or .NET) from within. For example, CAS needs to create web server on demand for the purpose of dynamic generation in various file formats.
(2) CAS should be able to generate the name spaces (e.g. <m:m and <svg:svg) to place MathML and/or SVG elements directly within an XHTML document.

(3) CAS should be able to manipulate XML DOM symbolically so that both MathML and SVG can be manipulated for various needs.
(4) CAS should be able to generate light-weight MathML, SVG for mobile devices.

4. CONCLUSION

Undoubtedly rich internet application technologies can offer real productivity benefits for the mathematical solution system, especially over the web. This paper analyzes the significance of the technological standards of RIA for mathematical solution over the web and suggests the reusable design patterns in delivering the mathematical services in the form of mathematical text, graphs, and animations. This paper also points out a few important considerations for the computer algebra system design as its implementation is inherently tied to the user interface technology.

REFERENCES

1. T. Gayley: Advanced J/Link Programming. *2001 Mathematica Developer Conference*, Champaign, Illinois, USA.
2. Laszlo Systems Technology White Paper: OpenLaszlo: An XML Framework for Rich Internet Applications. July, 2005.
3. Macromedia White Paper: Developing Rich Internet Application with Macromedia MX2004. August, 2003.
4. K. H. Song: Flash-Enabled User Interface for CAS. *Internet Accessible Mathematical Computation, a Workshop at ISSAC* 2003, Drexel University, Philadelphia, PA, USA.
5. ______ : Packaging Mathematica Packages. *Wolfram Technology Conference* 2005, Champaign, Illinois, USA.
6. ______ : Multimedia User Interface for CAS. *Proceedings of the Seventh Asian Symposium on Computer Mathematics* 2005, KIAS, Korea.
7. ______ : Rich Internet Application Technologies for Mathematica. *Wolfram Technology Conference* 2006, Champaign, Illinois, USA.
8. S. Wolfram: *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, 2004.

*DEPARTMENT OF MATHEMATICS EDUCATION, PUSAN NATIONAL UNIVERSITY, PUSAN 609-735, KOREA
Email address: khsong@pusan.ac.kr

**DEPARTMENT OF MATHEMATICS EDUCATION, PUSAN NATIONAL UNIVERSITY, PUSAN 609-735, KOREA
Email address: tssong@pusan.ac.kr