Remote Control and Monitoring via the Web

Martin SYSEL

Tomas Bata University in Zlín, Faculty of Applied Informatics,

Nad Stráněmi 4511, 760 05 Zlín, Czech Republic

ABSTRACT

An emerging strategy for application software is to provide web access to software previously distributed as local applications. It requires the development of an entirely different browser-based interface and adapting an existing application to use different presentation technology. This paper describes options for making MATLAB/Simulink applications available via the Web and the new developed web application called SimWebLink. The whole conception is based on the idea that it is possible to observe running simulation or real measurement results in the web browser. SimWebLink is suitable for a remote control and monitoring of the technological process uses MATLAB/Simulink via Internet.

Keywords: Remote control, Monitoring, Web application, Matlab, Simulink

1. INTRODUCTION

Modern communications and information technologies expand into research and all of potential technology branches, often far-away different from IT, what is also chemistry. New age and science progress in technology join different branches and give us a full advantage of all potential resources.

An emerging strategy for application software is to provide web access to software previously distributed as local applications. Depending on the type of application, it may require the development of an entirely different browser-based interface, or merely adapting an existing application to use different presentation technology. In software engineering, a web application is an application that is accessed via web browser over a network such as the Internet or an intranet. It is also a computer software application that is coded in a browser-supported language (such as HTML, JavaScript, etc.) and reliant on a common web browser to render the application executable. Web applications are popular due to the ubiquity of a client, sometimes called a thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. One of the possible application which offers advantage of high performance tools for calculation and simulation is program MATLAB.

1.1 MATLAB Web Server

MATLAB Web Server was the first application taking significant advantage as web applications. The MATLAB Web Server enables the creation MATLAB applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the results in a Web browser. The MATLAB Web Server depends upon TCP/IP networking for transmission of data between the client system and MATLAB. The required networking software and hardware must be installed on the system prior to using the MATLAB Web Server. This toolbox contains tools to connect MATLAB programs and HTML pages. The MATLAB Web Server requires MATLAB Release 11 or later. Unfortunately as of Release 2006b, this toolbox was discontinued and is no longer available for purchase. It is necessary continue using existing MATLAB Web Servers with MATLAB R2006a.

The application development process for MATLAB Web Server requires a small number of simple steps defined by Mathworks [3]:

1. Create the HTML documents for collection of the input data from users and display of output. It can be used a text editor to input HTML directly, or it can be used one of the commercially available HTML authoring systems.

2. List the application name and associated configuration data in the configuration file matweb.conf.

3. Write a MATLAB m-file that:

- Receives the data entered in the HTML input form.

- Analyzes the data and generates any requested graphics.

- Places the output data into a MATLAB structure.

- Calls htmlrep to place the output data into an HTML output document template. The maximum amount of HTML received data from MATLAB is 256 kB.

MATLAB Web Server applications are a combination of M files, Hypertext Markup Language (HTML), and graphics. Knowledge of MATLAB programming and basic HTML are the only requirements [3].

1.2 Web Deployment of MATLAB Applications

Going forward, there are several options for making MATLAB applications available via the Web. Each of the cases below contains brief describe the approach to create the web application.

It is possible to choose to create Web applications that access MATLAB directly (hosting MATLAB on the application server), or by building executables or components using the MATLAB deployment products (MATLAB Compiler, MATLAB Builder JA and MATLAB Builder NE). Deployed applications do not require MATLAB, and may be deployed royalty-free to as many servers as user wish [1].

The table 1 lists recommended deployment options for each platform, but other options are possible.

Table 1. Web deployment options

Platform	Application Server Does Not have MATLAB	Application Server Does have MATLAB
Windows	MATLAB Builder NE MATLAB Builder JA	MATLAB COM Automation
Linux Unix	MATLAB Builder JA	CGI with MATLAB as an Executable
MAC	CGI with an Executable Built by MATLAB Compiler	CGI with MATLAB as an Executable

MATLAB Builder NE: This part describes using MATLAB Builder NE to create a server-side COM or .NET component. Builder NE requires MATLAB, the MATLAB Compiler. The code using .NET or COM components from Builder NE with ASP.NET requires Internet Information Server (IIS).

Here are some pointers for ASP.NET deployment, particularly for transitioning from MATLAB Web Server [1, 2]:

• In M-code, it needs to use the print command to save the figure to a JPEG format as opposed to wsprintjpeg.

• It should be ensured that IIS points to the correct version of ASP.NET (as selected during compilation). After publishing the ASP.NET application to the web using the Publish to Web option in Visual Studio. If this step is not performed, the application displays an error pointing to the web.config file (depends on application).

• MATLAB Web Server HTML tags needed modification for ASP.NET/design mode compatibility: tags caused errors for design mode, and the page would not display in this mode. Errors were very helpful in providing guidance on needed changes. It is required to convert HTML controls to ASP control format (drop-down menus, text fields, etc.).

• If prohibiting client write access for the virtual directory containing the ASP.NET web application, pre-extraction of the CTF-archive file is necessary. Component Technology File (CTF). This file is independent of the final target type (standalone application or library) and specific to each operating system platform. This file, which is named with a .ctf suffix, contains the MATLAB functions and data that define the application or library and is embedded in the binaries of shared libraries and standalones by default.

MATLAB Builder JA: MATLAB Builder JA allows integrate MATLAB applications into the organization's Java programs by creating MATLAB based Java classes that can be deployed royalty free on desktop machines or Web servers [1, 2]. The MATLAB Builder JA product creates the Java classes by encrypting MATLAB functions and generating a Java wrapper around them.

It can be referenced MATLAB based Java classes the same way as any other Java class, for easy integration into servlets or JavaServer Pages (JSP). The Java classes created by the builder run against the MATLAB Compiler Runtime (MCR), which is the full set of shared libraries that support MATLAB. The MCR is provided with MATLAB Compiler for distribution with the Java classes. Both the Java classes created in MATLAB and the MCR can be deployed royalty free.

For Web-based applications, the builder provides AJAX based zoom, pan, and rotate controls for figures created in MATLAB. The builder also provides an API for automatically converting between Java and MATLAB data types. Using the builder's Remote Method Invocation (RMI) interface, the class can also be run as a persistent service, or spread processing across multiple processes [1]. Java classes created with the builder can be distributed at no additional charge.

Calling MATLAB functions via CGI: MATLAB can be called directly as an executable via CGI, or executables created with the MATLAB Compiler can be created and deployed. Using MATLAB Compiler it can be created a shared library which can be called from an executable written in C. This executable can be called by web server using the Common Gateway Interface (CGI). Using this solution requires the following steps [2].

• Compile the M-file into a shared library. (mcc B csharedlib:mylib myfile.m)

• Compile the C-file into an EXE-file which links to the created library. (mbuild mycfile.c mylib.lib)

• Copy the DLL, EXE, and CTF files to the web server.

• Install the MCR on the web server.

Direct access from the Web, using COM automatik: It is possible to call MATLAB through its COM automation interface, similarly to how would be called a COM object from MATLAB Builder NE. Then it is necessary wrap the MATLAB COM automation interface as a Web service.

Note that the MathWorks Software License Agreement specifically forbids exposing the MATLAB command line, i.e. allowing the end users of the Web site to define the XML string to execute. Instead, this approach must be used in the context of an application, executing specific user created M functions.

1.3 Problems and disadvantages

The problem of the above mentioned solutions lies in executing Simulink schema, or only final results are provided. The problem with running results is very serious, because the simulation of technological processes or real measurements can take a lot of time, especially in chemical processes. To avoid the problem with no access to running results, the development of a new application called MatlabLink started in the end of 2005 [5].

1.4 MatlabLink

MatlabLink allows to command and monitor a technological process controlled by MATLAB program (independent on which process it is - simulation or real laboratory experiment) by any computer in the Internet network. The monitoring is accessible to arbitrary number of computers, but just only one computer can control this process. This system uses client/server architecture. The proposed solution considers a contact server which allows communication with computers in private network. The whole application consists of several subsystems and makes use of several different technologies.

• JMatLink – Library, which allows MATLAB connection with Java [4]. By the instance of a JMatLink class it is possible to start up the program MATLAB, execute commands and obtain state of the Matlab workspace. This library uses MATLAB API Reference.

• MatlabLink client– Application that runs on the same computer where program MATLAB is installed. This part is written in Java and uses JMatLink library. MatlabLink periodically executes the main control, reads variables from workspace and communicates with the server.

• MatlabLink Server – Presents a server part of the application MatlabLink. MatlabLink server communicates with MatlabLink clients and stores communication on the server.

• MatlabLink Web - This part works together with MatlabLink server. HTML and JavaScript code automatically generated by this part is sent to the user web browsers.

MATLAB application commonly consists of several M files. It is necessary to initialize the variables, define starting conditions and load drivers in the beginning. Next, the process control proceeds in defined calculation loop. Finally, the application should be finished properly (for example unloads drivers), otherwise we can expect an undefined state or a damage of the controlled system. The developed application for remote control - MatlabLink - requires specific structure of M-files which is based on the above mentioned behavior. MATLAB application should consist of three parts; each of the parts can consist of arbitrary number of M-files. The name of the M-files is not strictly specified.

• The repetition rate should be defined in the first part and the name of this variable should be defined in the configuration file (XML format).

• The second part presents process control. MatlabLink provides periodic execution of this file. This file can call another m-files or functions. This file should be periodically executed by MatlabLink, because Matlab does not offer content of workspace during execution.

• The last part is executed after completing control loop and provides proper termination and disconnection of hardware.

It is possible to use this application generally for simulation of process control or for real-time measurements. The main problems of this application are the cooperation just only with MATLAB (do not support Simulink) and request for modification of existing M-files to special format. MatlabLink is not suitable for very quick measurements because it influences the calculation

of the control loop. Recommended sampling period is about 0.5s or bigger.

As research and development prefer using Simulink for process control and monitoring, the development of a new application called SimWebLink started in 2008. SimWebLink is designed to avoid the all of the above mentioned problems.

2. WEB APPLICATION SimWebLink

SimWebLink application enables the remote control and monitoring process in the program MATLAB/Simulink by the help of common web browsers. The control of this web application is very simple. The whole conception is based on the idea, when one computer control a technological process by a program written in MATLAB/Simulink then from other computers with internet connection is possible to monitor and partially also control the running measurement. The number of the computers that are able to monitor the technological process is not theoretically restricted (practically it is limited only by the server performance). The control commands are accessible only for one computer. It is possible to restrict both actions (monitoring and control) by access password. SimWebLink application consists of four parts, which are in detail described below. The communication scheme of each single application parts is shown in figure 1. SimWebLink application parts are written as a multithreaded application. This is distinguished from multiprocessing systems (such as multi-core systems) in that the threads have to share the resources and allow speedup of the overall system throughput. The developed application is based on a client/server approach; the sockets are used for the network communication. The proposed solution makes possible to monitor and control a technological process, the control computer of which is in local (private) network. The public IP address is required only for the server, because serves major parts of the functionality of the SimWebLink and can be called as a "middleman". In the general case, client user can monitor and control a technological process in private network from another private network.

2.1 SimWebLink Block

The new communication block was created to the program Simulink, which is called to send values of the passing signals to the application part SimWebLink Client. The base element of the block is S-function block, which use C MEX file [6]. SimWebLink Block has just only one input parameter - period of the signal output. All of the signals sending by this present block are subsequently displayed to the one graph in the web browser. Generally, it is possible to use more than one block in the Simulink scheme and so display more than one final graph in the web browser. The communication between SimWebLink Block and SimWebLink Client is

very simple (it is a unsecured text form in this time). SimWebLink Block supposes existence of the SimWebLink Client on the same computer. The communication between these parts proceeds only on the local computer (localhost communication).
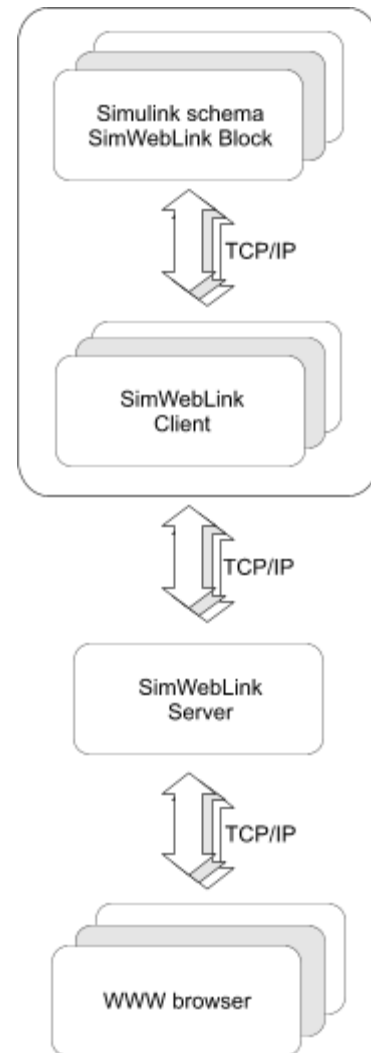

Fig. 1. The communication scheme

MatlabWebLink Function: As a SimWebLink Block can be used only in the Simulink schema, new MATLAB function with similar functionality has been developed. It is enough change to place this function into main calculation loop of the program, where important calculations proceed. The input parameters submitted to the function are sending to the application SimWebLink Client.

2.2 SimWebLink Client

The part of the application SimWebLink Client has to be running on the same computer, where program MATLAB/Simulink is installed. The objectives of this application part can be described in a few numbers of points:

> • Connect to the SimWebLink Server after start-up and report state 'ready to work'. Subsequently XML file describing attached technological process or simulation is sent off. The simplified structure of this document is as follows:

<?xml version="1.0" encoding="UTF-8"?>

<SimWebLink>

> *<label>Schema label</label>*
>
> *<stoptime>60</stoptime>*
>
> *<outputtime>1</outputtime>*
>
> *<passmonitor></passmonitor>*
>
> *<passcontrol></passcontrol>*
>
> *<block id="1">*
>
>> *<graph>Graph label</graph>*
>>
>> *<signal id="0">*
>>
>>> *<name>Desired*

value</name>

>>> *<line>solid</line>*
>>>
>>> *<color>black</color>*
>>
>> *</signal>*
>>
>> *<signal id="1">*
>>
>>> *<name>Output value</name>*
>>>
>>> *<line>solid</line>*
>>>
>>> *<color>red</color>*
>>
>> *</signal>*
>
> *</block>*

</SimWebLink>

The explanation of the tags:

<label> contains the title label of whole Simulink schema.

<stoptime> contains the recommendation for total measuring time value in the seconds. It is possible to set-up measuring time in the web browser interface as well as many others parameters defined in the XML file.

<outputtime> contains the recommendation for the period of sending out the signal values. It is possible to set-up smaller value in the web browser interface, but it is necessary to entertain practical significance of the high frequency sampled data in the client web browser.

<passmonitor> and <passcontrol> contain predefined access passwords. It is possible to change default password in the web browser interface.

<block id="1"> contains the identification of each single SimWebLink Block in the schema. All incoming signals of each block are finally displayed in the one figure that is named by the tag <graph>.

<signal id="0"> contains the description of each single input signal, it is possible to define name, color, and type of the line.

The text must be enclosed between a root start-tag <SimWebLink> and a corresponding end-tag. The only indispensable syntactical requirement is that the document has exactly one root element. The root element is preceded by an XML declaration element stating what XML version is in use; it also contains character encoding and it might contain external dependencies information.

> • Expect control commands (XML file) from the client web browser. In the case of start command is obtained, The SimWebLink Client configures schema parameters according to delivered XML file and launch MATLAB/Simulink schema in the MATLAB engine. It uses MATLAB API reference [7].

> • The last role of the SimWebLink Client is expecting an arrival data from the SimWebLink Block and sending them immediately to the SimWebLink Server to be saved and prepared for presentation.

Presently, this part is started-up as a common executable application, however, let's suppose, final version will be published as a service (daemon) running on the background. The objective is starting-up and monitoring of the technological measurement in anytime, without dependencies on user sign on.

2.3 SimWebLink Server

The server-side part of the application registers connected SimWebLink Clients and receives data from them. The data are saved in the MySQL database and prepared to the suitable form for publishing in the web browser. The server expects commands from users' web browser too.

The SimWebLink Server is written as a multithreaded application, a new thread is created for servicing each connected client. It is possible to limit the number of connected clients according to the server performance. The PHP scripting language is used on server-side for

creating dynamic web pages, the JavaScript language run locally in users' browser (client-side) dispatches requests for information to the server. The new information is continuously updated by AJAX technology (Asynchronous JavaScript and XML) in the users' web browser. With AJAX, web applications can retrieve data from the server synchronously in the background without interfering with the display and behavior of the existing page. SimWebLink Server expects commands from the user's web browser. The web server with a PHP support and database engine MySQL is required for SimWebLink Server to run.

3. CONCLUSIONS

The application MatlabLink is replaced by the new application SimWebLink, which extends computation possibilities of the MATLAB/Simulink. The application SimWebLink is a multithreaded and a multiplatform application. The whole conception is based on the idea that it is possible to observe running simulation or real measurement results in the web browser. The data presentation lag is minimal and SimWebLink Block does not affect Simulink schema runtime too much. The main advantage is a remote control and monitoring from any web browser in the Internet. The presented data are stored in the database for the later uses. The development of the application still continues and new operating functions will be implemented.

4. REFERENCES

[1] Anonymous. **Web deployment of MATLAB applications guide**. The Mathworks support. [internet]. Available at: http://www.mathworks.com/support/tech notes/1600/1608.html [Accessed 30 September 2008].

[2] Anonymous. **Technical Solutions**. The Mathworks support. [internet]. Available at: http://www.mathworks.com/support/solutions/data/1 -2YVMOV.shtml [Accessed 30 September 2008].

[3] Anonymous. **MATLAB Web Server**. The Mathworks Inc., Natick, USA, 2001.

[4] Mueller, S., **JMatLink connect MATLAB and Java**. JMatLink site. [internet]. Available at: http://jmatlink.sourceforge.net/ [Accessed 28 December 2005], 2005.

[5] Sysel, M., Pomykacz, I., Remote process control using MATLAB. **9th IFAC Workshop Adaptation and Learnig in Control and Signal Processing**. Saint Petersburg, Russia, 2007.

[6] Anonymous. **Writing S-Functions**. The Mathworks Inc., Natick, USA, 2008.

[7] Anonymous. **MATLAB C and Fortran API reference**. The Mathworks Inc., Natick, USA, 2008.

5. ACKNOWLEDGEMENTS