

# SIMULINK WEB EXPLORER

*Milan Brejl<sup>\*</sup>, Andrzej Lara<sup>\*\*</sup>*

<sup>\*</sup>Motorola Czech Systems Laboratories,  
1. máje 1009, 756 61 Rožnov p.Radh., Czech Republic,  
<sup>\*\*</sup>Motorola Software Development Center,  
Jodłowa 13, 30-252 Kraków, Poland

**Abstract:** *The Simulink Web Explorer is an application that enables the work with Matlab Simulink models for wide range of users over the Internet or Intranet. Our Web Application connected to the Matlab simulation server with our Web Toolbox installed offers users an easy and attractive way to explore and set up multi-level Simulink models, simulate them and view/download results - all using just a browser.*

## I. INTRODUCTION

The Matlab Web Explorer combines two already well-known technologies: the web browsing and simulation capabilities of Matlab. Once a Matlab Simulink model is created it can be accessible via the very common and favorite way for information sharing - the Internet. A user on the Internet can setup simulation and block parameters and try to simulate the system with his or her own settings, view simulation results as pictures or download them as data files. Moreover he or she can open block subsystems and explore the model into deeper levels this way and even modify the model by exchanging blocks with one of prepared block variants.

The whole system consists of three parts (see Figure 1):

- Simulation server - Matlab,
- Web server and
- Client

From a system user perspective the whole system appears as a collection of frames arranged in one window. A main frame contains schematic of an application under consideration. It is possible to explore the hierarchy of the schematic if necessary. Other frames provide other functionality: parameter setting of any block, choice of signals to observe and format of the results from the simulation. If any topology modification is desired, blocks can be exchanged by using a provided library. The system enables also to run a user code on a remote machine.

The Web Server is a link between the Simulation Server and the customer on the Internet. The Web Server dynamically composes HTML pages that maintain the customer moves and schematic changes. Once the customer finishes schematic modifications and pushes a "Simulation Start" button, the Web Server composes the customer-modified model and posts it to the Simulation Server.

The Simulation Server starts the simulation and makes required customization of an application. After simulation has been run, the simulation results are stored and the user is notified in another window.

Apart from the main components of the system, additional utilities have been developed to ease the process of converting a simulation design created for stand-alone machine to the web based system described above.

## II. CLIENT APPLICATION

The Client Application is built on JavaScript, Dynamic HTML and CSS (Cascading Style Sheet) advanced standards.

The user selects one of reference projects in a virtual project library. This opens project window. The project window consists of three main frames:

- **Simulation frame** with global simulation parameters setting, results output format selection and Start Simulation button
- **Navigation frame** with links to application notes, control instructions, project library etc.
- top-level **System frame**

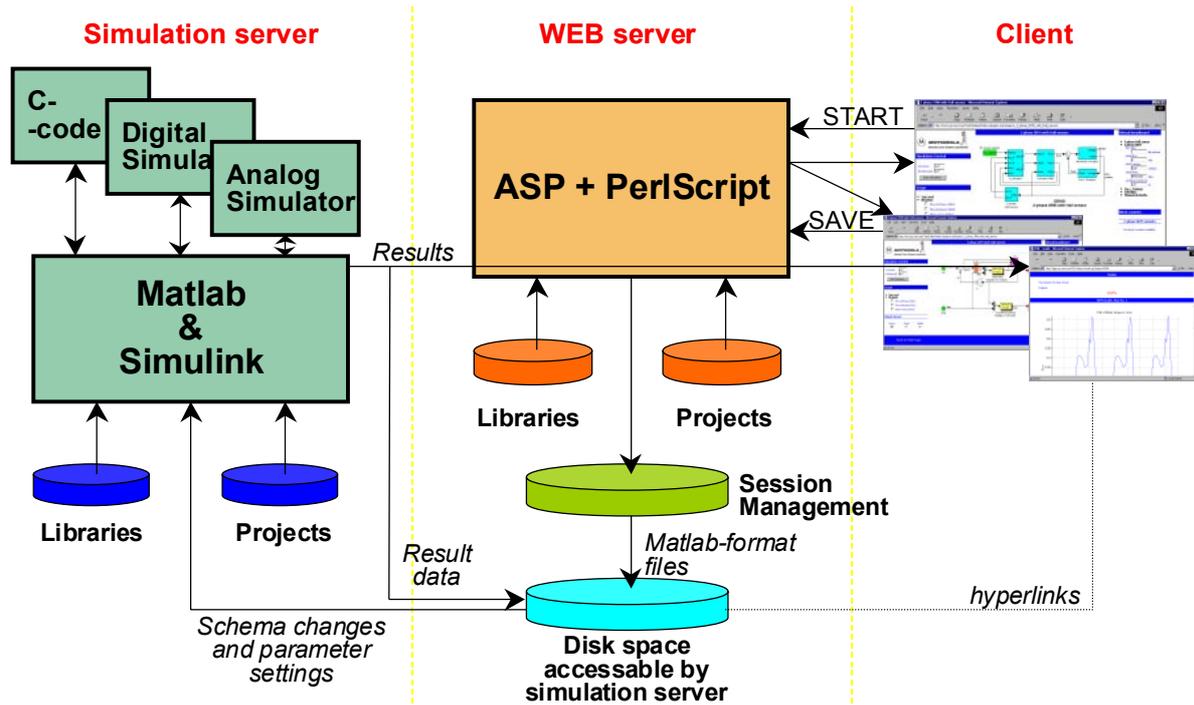


Figure 1: System functionality.

On the System Frame the user can see the click-sensitive *system schematic*, *block parameters tree*, list of exchangeable *block variants* and output *scope signals*. As the user clicks on a block, the block parameters branch in the parameters tree opens and exposes block parameters. The Block Variants pane displays available block variants. The variant blocks must have the same number of inputs and outputs. If the user clicks to exchange a block with one of its variants, a variant block appears in the schematic and its parameters branch in the parameters tree opens, while the original block in the schematic and its parameter branch are hidden. If the user clicks on a scope in the schematic, the corresponding scope signal will be checked in the scope list. If a user double-clicks on a block, two actions can happen. The message “Elementary block. No subsystem to open” will appear if the block is the lowest level block in hierarchy or a new window with a system covered by this block is opened (see Figure 2). In the lower-level system the user can work the same way to setup block parameters, exchange blocks and select scope signals.

### III. WEB SERVER APPLICATION

The Web Server application is a mediator between the Matlab Simulation Server and the user browser. It is supplied to:

- dynamically create requested system pages,
- hold the user session management,
- translate and send all user settings and schema changes to the simulation server,
- start the simulation on the simulation server,
- get the simulation progress and post it to the client during the simulation and
- get the result page and result data.

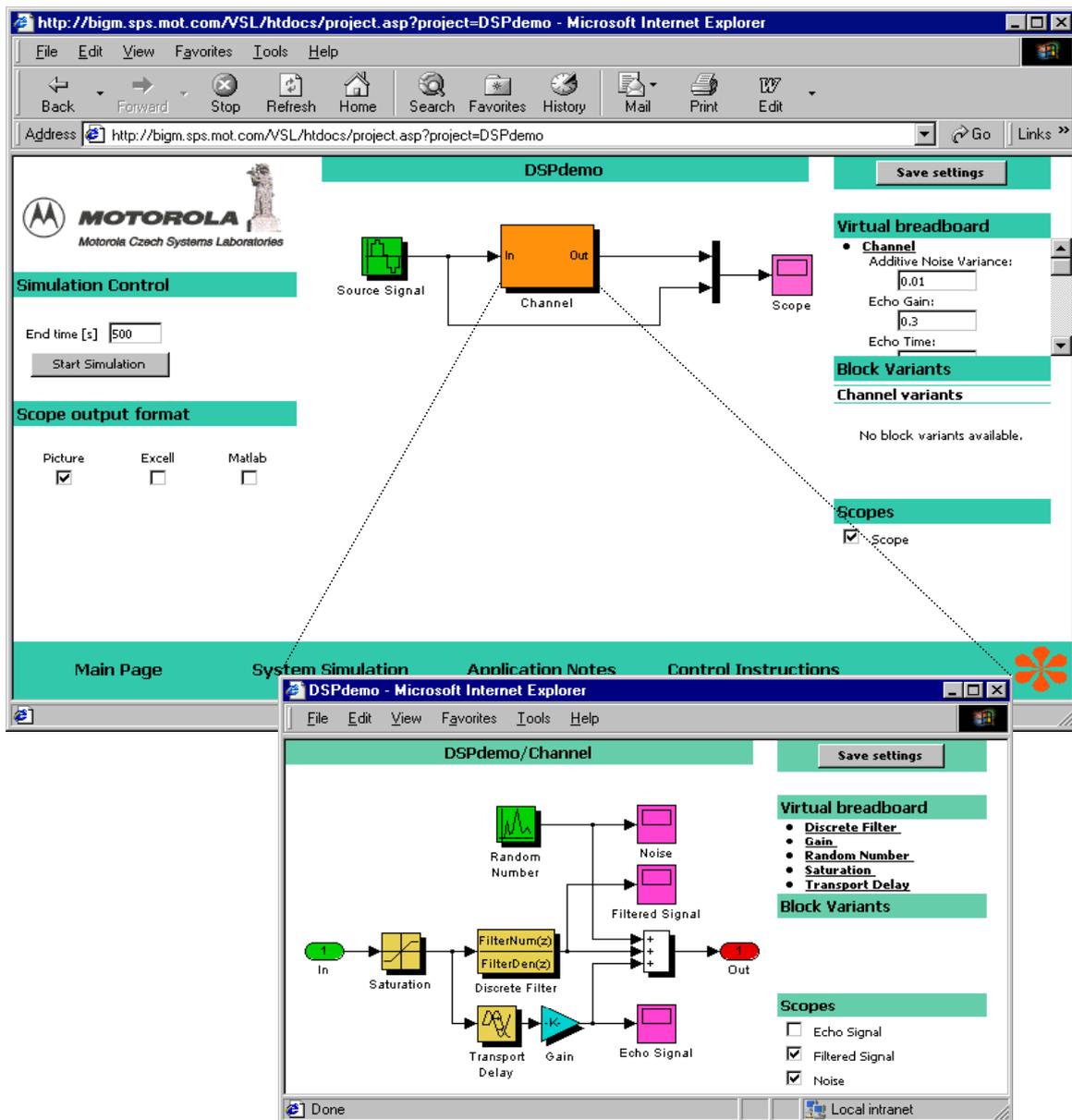
The Web Server Application is based on ASP (Active Server Pages) technology [5] with PerlScript scripting language [6] used. The application uses Perl-language setup files that hold the project models and libraries data for display as a web content. Based on this, any requested system or subsystem page with proper schematic, parameters tree etc. can be created. User schematic changes, parameter settings and scope selection are saved on the web server in a directory, which is specific and unique to the current user session.

A disk space shared by both Web Server application and Simulation Server is required. When the user clicks the Simulation Start button, Matlab format files describing all user changes to the reference project are created and saved there.

## IV. MATLAB WEB TOOLBOX

A specially created Matlab Web Toolbox establishes connection between the Simulation Server and the Web Server. The Matlab Web Toolbox actually runs simulation and produces simulation results. The communication is arranged in two ways: through the CGI interface as in [3] and through files stored in the shared directory. The CGI interface is used to exchange general system information pertaining to network system, directories and files stored by the Web Server. All information supplied by a user through the client Application is contained in the files stored by the Web Server in the shared directory. The Matlab Web Toolbox enables to run each simulation on a different computer running Matlab Web Server.

After a simulation is started by a Web Server, the Matlab Web Toolbox analyses all changes requested by a user. First, based on appropriate files, it adds to the reference system the "To Workspace" block and connects it with all requested signals. Later, the Matlab Web Toolbox exchanges blocks and changes parameters according to the request. The modifications can be done at any level of hierarchy and are analyzed at all times for validity. Invalid requests are discarded and reported. The modified system is saved and then simulated. Progress of simulation is reported to the Web Server on-line. As soon as a simulation is finished, simulation results are retrieved from Matlab and converted to desired formats. Finally, an html code with links to the produced result files is generated and returned.



*Figure 2: Client Application.*

## V. TRANSFER FROM SIMULINK TO WEB

Additionally the Matlab Web Toolbox contains functions that create Perl-language setup files required by the Web Server Application (see Figure 3.). The `gen_perlsys(sys)` function generates files describing the `sys` Simulink system with all subsystems and filenames for system schema pictures. The `gen_perllib(sys)` function processes a `sys` library system. The `gen_perlvar(sys)` function automatically creates variants for a block from all blocks in the `sys` system.

This way any Matlab Simulink system can be easily transferred to the Web.

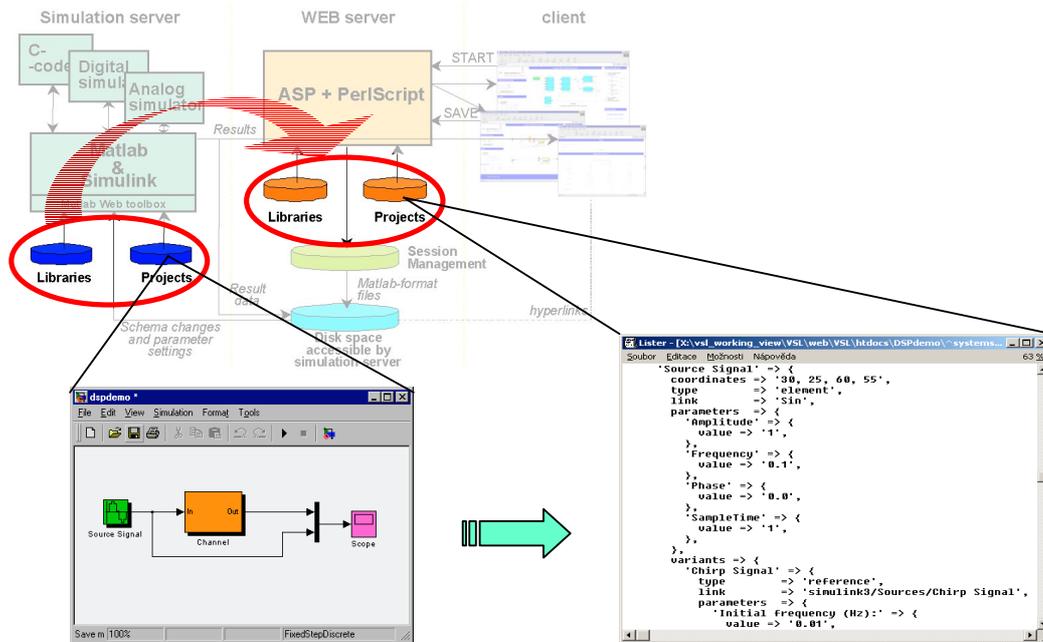


Figure 3: Transfer from Simulink to Web.

## VI. CONCLUSION

The presented application can serve as a tool for presentation of models. The target public can become familiar with the presented subjects very deeply, in a very interesting way, anytime and from anywhere.

## REFERENCES

- [1] Matlab - Using Matlab. The Mathworks, Inc., Version 5, 1999.
- [2] Simulink - Writing S-functions. The Mathworks, Inc., Version 4, 2000.
- [3] Matlab Web Server. The Mathworks, Inc. Version 1, 1999.
- [4] Ivo Vecera, Milan Brejl, Andrzej Lara: Virtual System Lab - a Motor Control Simulation Environment Available Through WWW. In: PCIM'2001 Proc., Nuremberg, Germany.
- [5] Active Server Pages (ASP), <http://msdn.microsoft.com/workshop/server>.
- [6] Active Perl, <http://www.activeperl.com>.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

### Authors:

Ing. Milan Brejl, Ph.D.  
 Motorola Czech Systems Laboratories  
 1. maje 1009, 756 61 Rožnov p.Radh.  
 Czech Republic  
 Milan.Brej1@motorola.com

Ing. Andrzej Lara  
 Motorola Software Development Center  
 Jodłowa 13, 30-252 Kraków  
 Poland  
 Andrzej.Lara@motorola.com