AGMAWEB – AUTOMATICALLY GENERATED MATLAB WEB SERVER PRESENTATIONS

R. Láska, M. Kutil

Department of Control Engineering,
Faculty of Electrical Engineering,
Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2, Czech republic
fax: + 420 224 355 703
laskar1@fel.cvut.cz, kutilm@fel.cvut.cz

Abstract: This paper deals with the system Agmaweb. It was built to make designing of Matlab Web Server presentations as simple and fast as possible and implement some additional features (sessions, multilanguage support, integration with other WEB publishing systems). Using Agmaweb the author of the presentation creates only the `m-file` and adds to this `m-file` simple special comments. According to this comments all web pages are generated automatically, dynamically by Agmaweb.

Keywords: Matlab, Matlab Web Server, education, web.

1. INTRODUCTION

The internet textbook SARI (John, J. (2004)) has been previously developed as a support for teaching of two basic courses focused on the ground of Control Theory (Systems and Models, Systems and Control) at Department of Control Engineering, Faculty of Electrical Engineering, CTU Prague. There was an effort to integrate some interactive examples into this internet textbook. These interactive examples (let us say, more generally, presentation instead interactive example) should allow the reader to be a really active student of discussed problems not only a passive consumer of text. These presentations can be powered by Matlab Web Server (Reschke (2002)). Unfortunately we have found some problems, disadvantages and limitations of the use of Matlab Web Server for preparation these presentations on the basis of Matlab `m-files` that had been prepared for education purpose earlier.

Main problems are:

- preparation of one presentation is very time consuming, principally because an input and output HTML page must be prepared as a special template
- the code of earlier prepared `m-files` must be widely modified, it causes problems with updates
- the author of an presentation has to be familiar with internet technologies (HTML, CSS, HTTP protocol)
- there is a security problem, all inputs from a user must be carefully checked [1]
- the author of example must be familiar with server configuring procedure (reconfiguration is necessary for each new example)
- `m-file` that produces an output to the Matlab command window can't be run on Matlab Web Server

These problems have led us to the development of system Agmaweb that solves all of the mentioned problems, moreover, it implements some additional features.

[1] Input data from a web page is passed as strings that must be converted to numbers, matrix, etc. but eval function can't be used, because the user is allowed to enter e.g. !del *. to the input field.

2. PRINCIPLE

Each presentation operates according to these three steps:

(1) A user fills input data in the form on input web page and submit it.
(2) The server computes results.
(3) The server sends a web page with results (including images) to user's web browser.

An author of presentation should prepare `m-file` according to the computation is done, furthermore he should prepare input and output web pages (in fact templates of input and output pages). Main purpose of input template is to specify the HTML page where form with fields for input data is. Main purpose of output template is to specify HTML page where are special marks (`$variable_name$`) that are replaced by computed results. But as it was said above, preparation of these templates is time consuming and annoying. So it is a job for a computer – for Agmaweb. We only have to give Agmaweb basic information: what input data (what type of data and the name of variables) our `m-file` (script) needs and what output data (again type of data and names of variables and also names of figures) it produces. The principle, that we designed, describing this information is called Control Comments.

3. CONTROL COMMENTS

Control Comments are the heart of the system Agmaweb. The only thing that the author of presentation has to do is to write Control Comments to an `m-file`, thereby the whole presentation is prepared. Agmaweb generates an input and output web page automatically and dynamically according to these Control Comments. Main advantage of solution based on comments is that only one version of `m-file` is needed and this version can be run as normal `m-file` in Matlab. Moreover it can be simultaneously used by Agmaweb for creating a web presentation.

There is listing of a small exemplary `m-file`. Input and output web pages, generated automatically by Agmaweb according to Control Comments in this listing, are in Fig. 1 and Fig. 2:

```
% Script for addition of two numbers.
% First addend is in variable add1,
% second addend is in variable add2.

% AgmaWEB ver1 en
% input
% stdhelp;
% int add1 "First addend:" "h1";
% float add2 "Second addend:" "h2";
%
% output
% stdout;
```

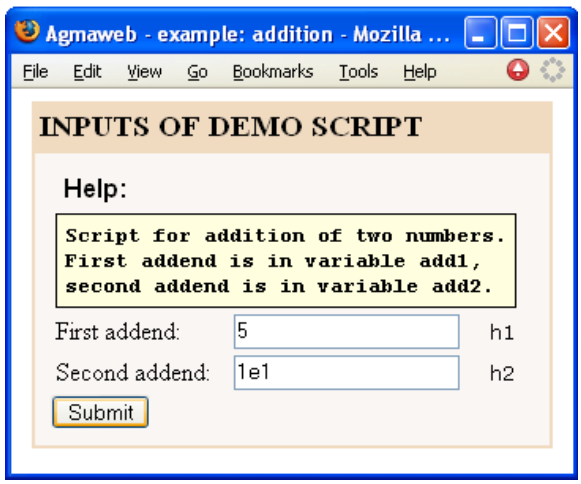

Fig. 1. Input page

```
% int result "Result: " " h3";

result = add1 + add2;
disp(sprintf('%i + %f = %f', ...
     add1, add2, result));
```

The syntax of Control Comments is inspired by language C. Control comments are written as a continuous block of comments, so each line of this block must begin with character `%`. The first line of the block of comments is a header which can optionally specify a language used for reporting errors . Comments are divided into two parts, the first part (Input Comments) begins with the key word `input` and the second part (Output Comments) begin with the key word `output`.

3.1 *Input Comments*

Input comments determine the content of an input web page of presentation. Particulary they determine names and types of variables needed for the `m-file` and these inputs appear as input fields on the input web page. Each input variable is specified by one comment's line. Possible input types are mentioned in Table 1.

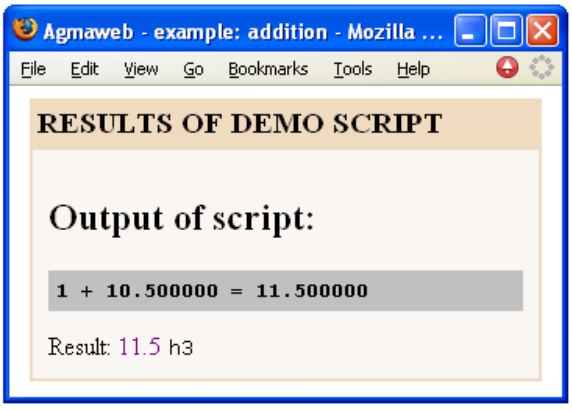

Fig. 2. Output page

type	meaning
`int`	integer variable
`float`	floating point variable, in this version of Agmaweb "." is decimal point
`matrix`	matrix variable, matlab syntax is used
`help`	special type intend for displaying some texts (titles, captions, additional explanations...) on input web page
`stdhelp`	special type intend for displaying standard help of your m-file on input web page

Table 1. Possible input types

type	meaning
`int`	integer variable
`float`	floating point variable
`matrix`	matrix variable
`figure`	Handle on Figure object, details are discussed below.
`input`	Include input page of any presentation, details are discussed below.
`stdout`	Include standard output of m-file in output web page.

Table 2. Possible output types

Fig. 3. Checking of correctness of input data

In this paper the syntax of types is not discussed, see (Láska (2005)) for details. Use of types such as `int float` and `matrix` is common. Each variable of one of these types is represented on input web page as an input field. Correctness of each input is checked by Agmaweb on a server according to its type. For example, type of the first input field in Fig. 1 is `int` when the submitted value isn't an integer, an error web page is shown (Fig. 3). This correctness is checked by regular expressions, which guaranties security as well.

The type `stdhelp` is used for displaying standard help of `m-file` on the input web page. `help` is used for displaying some additional texts (titles, captions, additional explanations, ...) on the input web page.

3.2 *Output Comments*

Output comments determine outputs of a presentation. Rather they determine names and types of variables which contain results of computation of an `m-file`,also they determine when standard output will appear on output web page and can specify some texts (titles, captions) that will be displayed on the output web page. Each output variable is specified by one comment's line. Possible output types are mentioned in Table 2.

Figures in output web page One picture is better than a thousand words, therefore to include the figures in to output web page with computed results is very important. Unfortunately doing so without Agmaweb, only using standard capability of Matlab Web Server, requires writing a special code for each figure to export this figure to a format useable for web pages (e.g. PNG, JPEG). Also deleting of thus exported files must be added to code (Reschke (2002)). Agmaweb deals with the export of figures using type `figure`. If an author of presentation wants to include a figure to the output web page, he specifies it in output control comments by a variable of type `figure` and in code handle of right Figure object is saved to this variable. There are a possibilities to specify the size of figure in pixels and the resolution of the figure. This short code show this technique:

```
...
% output
% figure step_handle "Step" "-dpng -r80 350 230";
...
step_handle = figure;
step(tf([1], [1 1]));
...
```

Input form in output web page To improve interactivity of presentation it is possible to include an input of a presentation in the output web page of presentation. The included input can belong to the same presentation that produces output web page or it can belong to different presentation, even from a different project. Output type `input` provides this feature. This technique is demonstrated in the following example. User enters a transfer function of the system and a controller in order to compute step response of closed loop (Fig. 4). On the output web there is an input included for entering new controller (Fig. 5), the system is the same and user can compare various controllers. The following source code of `m-file` can be for example:

```
% AgmaWEB ver1
% input
% matrix n_s "Num: " "coefficients of numerator";
% matrix d_s "Den: " "coefficients of denumerator";
% help "Regulator";
% matrix n_r "Num: " "coefficients of numerator";
% matrix d_r "Den:: " "coefficients of denumerator";
%
% output
% figure F_step "Step response" "-dpng -r80 350 230";
% input sys2 "demo" "";
```

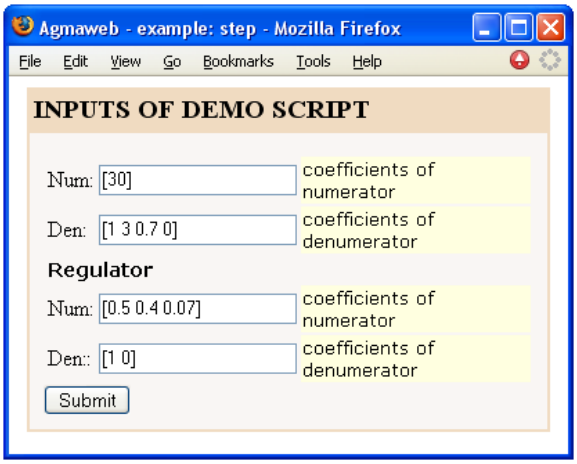

Fig. 4. Input page

```
F_step = figure;
CL=feedback(tf(n_s, d_s)*tf(n_r, d_r), 1);
step(CL, 8);
```

4. PROJECTS

Agmaweb has project oriented design. One project is a set of several presentations that belong to the same theme. All presentations, that belong to the project, have the same design of all web pages (input, output, error page). For this purpose there can be saved CSS file (`projects_name.css`), input template (`input.xml`), output template (`output.xml`) and a template for error web page (`error.html`)in the project folder. Use of these templates is optional. If a file of any template is missing, default template is used instead. It suffices, due to these templates, to save `m-file` to the folder with the same name and the complete presentations is done [2]. One template (for an input, output or error page) specifies uniform design for all presentations in the project, but actual content (e.g. input fields with captions) is generated dynamically for each presentations separately. The template is an XHTML file contains a special tags [3]. This special tag is replaced by dynamically generated content when a page is generated. These replacements are done inside Agmaweb by XSLT processor (Holzner (2002)). For example here is an input template with special tag: `<mws_input>`:

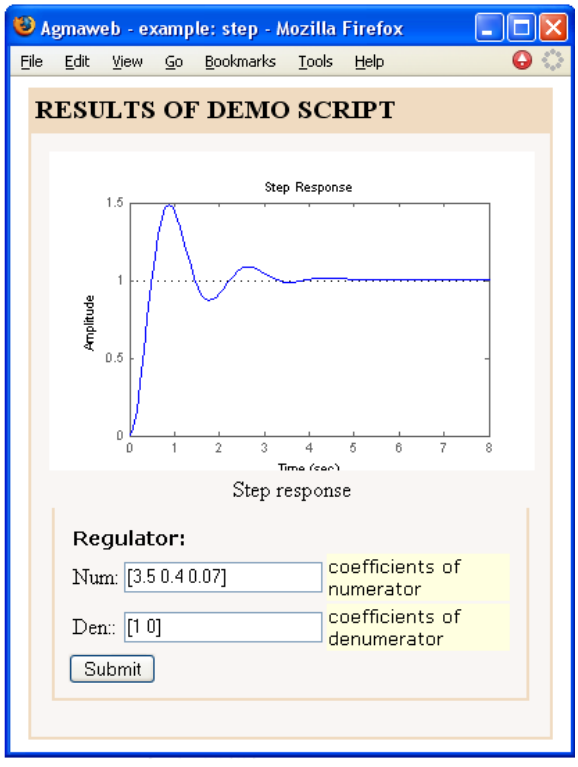

Fig. 5. Output page with included input

```
<?xml version="1.0" encoding="windows-1250"?>
<html>
  <head>
    <title>Agmaweb - example: addition</title>
    <link rel="STYLESHEET" type="text/css"
              href="/agmaweb/demo/demo.css" />
  </head>
  <body>
    <h3 class="caption">Inputs of demo script</h3>
    <mws_input>
      <edits/>
      <settings>
        <stdhelp_cap value="Help:"/>
        <help_newline value="&lt;br&gt;"/>
      </settings>
    </mws_input>
  </body>
</html>
```

5. OTHER FEATURES

It emerges, during time of development, that some other features could be very useful. We implemented some of them (those we consider more important).

Agmawebs sessions HTTP protocol is stateless. It means that the protocol has no built-in way of maintaining the state between two transactions. When a user requests a page, followed by another page, HTTP does not provide a way for us to tell that both these requests came from the same user. In the case of our presentations this means that entered input data would be lost after each computation of `m-file`, and user would have to enter this data again. Agmaweb deals with this problem by implementing sessions [4]. Agmaweb implements sessions using random session ID that is passed along through URLs. The whole Matlab's workspace is always saved immediately after execution of an `m-file`, so all variables that

[2] Design of so prepared presentation is poor, but readable and good useable.
[3] Exactly, template is not valid XHTML according to W3C standard just due to our special tag.
[4] Sessions are well-known technique. E.g. PHP supports it very well.

the `m-file` create are saved on the server. They are saved in a file that is unique for one session, a user never can use data of any other user. Note that a session begins when a user asks the server for the first input web page and the session is closed when the user goes to other web page (goes to another server) or closes browser. Saved workspace is loaded before every execution of an `m-file`. Thus the saved workspace is common for all scripts in one project. This feature can be used to pass variables from one presentation to another presentation in one project or it can be used to pass variables from one instance of script execution to another instance, e.g. an author can simply create a counter of launches of his script during one session. This is also used in an example mentioned above. The transfer function of a system is entered by a user only once, it is saved on the server and later the user enter only a transfer function of controler (Fig. 4) and (Fig. 5).

Remote calling System Agmaweb can be linked up with a web publishing system. This publishing system can run on a different server and can include pages generated by Agmaweb (input, output and error page) as a part of publishing system pages. This solution requires a piece of programing inside web publishing system [5]. To support this technique there is a capability in Agmaweb which specify a URL, that is a destination for user data from input web page. This can be done by the tag `<form_action>` in an input template, for example:

```
<form_action
  value="http://example.cz/index.php?ID=matlab" />
```

Including header and footer Every good web page should have well arranged navigation and should provide basic information about author and last update of the page. To include this information to each web page manually is difficult. Therefore Agmaweb provides support for doing it automatically. There is a special tag

```
<mws_include file="file.xml" />
```

that can be used in the input template (input.xml) or the output template (output.xml). This tag is replaced by the content of `file.xml` during processing of the template. This tag can be used more generally not only for header and footer, it can be used at any place so many times as needed.

[5] We have implemented it in PHP for internet textbook SARI.

6. FUTURE PLANS

We would like to implement some new features into system Agmaweb in the future. Now Agmaweb produces HTML web pages, but we intend to change it so that Agmaweb will produce only valid XHTML pages. We are not entirely satisfied with response time of the server. We would like to find a solution to decrease the response time. Now Agmaweb works only with an `m-file` that is written as a script, our effort is to implement the same functionality also for functions.

We also research possibles of creation of a virtual laboratory accessible from internet using 3-D visualisation of physical models. We assume using VRML for this purpose.

7. CONCLUSION

The system Agmaweb decreases time needed for preparation of Matlab Web Server presentations. At present time Agmaweb is used in internet textbook SARI (`http://dce.felk.cvut.cz/sari/SARI-Matlab.html`) and Aerospace (`http://mws.felk.cvut.cz/agmaweb/aerospace/`).

ACKNOWLEDGMENTS

This work was supported by the Ministry of Industry and Trade of the Czech Republic under Project FD-K3/082.

References

Holzner, S. (2002). *XSLT, Příručka internetového vývojáře*. Computer Press. Praha.

John, J., Kutil, M. (2004). *Internet Textbook SARI* [online]. ⟨`http://dce.felk.cvut.cz/sari/`⟩.

Láska, R. (2005). *Agmaweb - Documentation* [online]. last revision 4th of april 2005 [cit. 2005-04-11]. ⟨`http://mws.felk.cvut.cz/agmadoc/pdf/agmaweb.pdf`⟩.

Reschke, D. (2002). Matlab web server ve výuce. Master's thesis. Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague. Prague.