Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 07:00:25

Annales UMCS Informatica AI 1 (2003) 325-330

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Using PHP and HTML languages to create graphical interfaces and to remote control of programs

Bogdan Księżopolski*, Paweł Łuka

Institute of Physics, Maria Curie-Skłodowska University,
Pl. M.Curie-Skłodowskiej 1, 20-031 Lublin, Poland

Abstract

In this document we describe how to use PHP and HTML languages to create graphical interfaces to any program. The interface is used to send data and parameters to the program from a web page. We can choose different kinds of data and the format of received data. We show an example, in which a graphical interface controls a program written in Fortran 77.

1. Introduction

In normal situation if we want to use any program we must first have a version of this program designed to our operating system and an appropriate platform which it is not always possible. We can also use the source code, but we need to have the right compiler. Our idea is to create graphical interface to a program installed at a telecommanded server. After that it's possible to send data and parameters to the program from a web page accessed by any web browser at user host. After the program execution, the results are sent back to the user in a text or picture form to the browser or they are sent by the e-mail. Interface can by right viewed by any kind and version of graphical web browsers, while we use first standard of HTML language, also PHP is server-side script, it means they are executed at server and the result is formatted to the HTML page. One of these interfaces is located at: *http://kft.umcs.lublin.pl (*banner Interactive Database} with the HFB program written in *Fortran 77* which calculates proton and neutron density distributions in atomic nuclei [1].

2. Client side

At first we will describe the web page visible by a client at his web browser. This page contains interface with a form, where the user can enter the input data

* Corresponding author: *e-mail address:* ksiezop@tytan.umcs.lublin.pl

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 07:00:25

and choose an appropriate set of parameters. The program can be then executed by clicking the START button at the submit type.

This page contains some types of form fields: "text", "radio", "checkbox", and "submit" visible in Figure 1, where an example of the HTML code of these fields is given [2]:

```
<FORM ACTION="o2.php" METHOD="POST">
<INPUT TYPE="text" NAME="neut" VALUE="liczba neutronów">
<INPUT TYPE="radio" NAME="sk7" VALUE="Skyrm Sk7">
<INPUT TYPE="checkox" NAME="wynik2" VALUE="Wykres Halo">
<INPUT TYPE="submit" NAME="start" VALUE="start">
</FORM>
```

At this web page user can fix the proton and neutron numbers, select the type of the Skyrme forces, and give his e-mail address, which allows to send him back the text output and the figures in the postscript format.

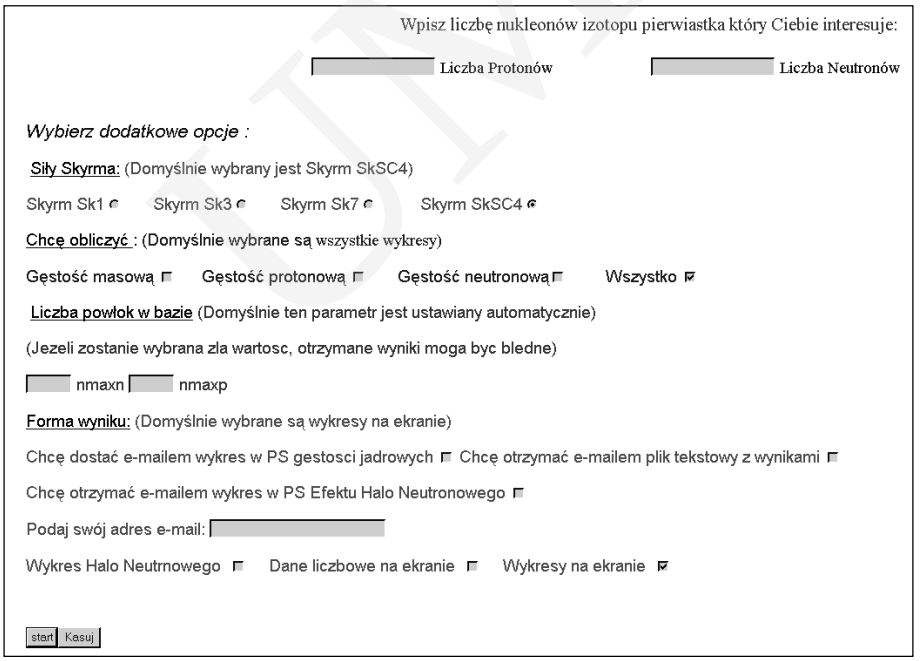


Fig. 1. Web page with graphical interface to the HFB program

3. Server side

After the short description of creating of the form fields we would like to describe the most important part of the interface, i.e. the PHP code. The PHP code is used to configure and execute the program and to format the results. In this chapter we divided description into smaller parts. The input values and

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 07:00:25

parameters are transmitted from the form fields at the HTML web page to the PHP program which is a part of the same file. This is possible by using special environment parameters.

3.1. Program executing from the PHP script

There are two ways to transmit the variables to the program by the PHP script:

a) With additional files created by the script before execution of the program

First we need an access to this file by creating handler in a PHP code [3]. Then we put the data from the HTML form and execute the program. Instead of typing the parameters from keyboard we put them from this file. In the HFB program for example there are some default parameters defined inside the program calculating the density of nuclear matter. There is sample code of this method:

```
120 $fp=fopen("/var/tmp/test.inp", "w+");
121 fputs($fp, "0,\n 61,0.25,\n 0,2,\n 80,1.e-5,1.e-5,-4,0.0001, \\
122 \n,,,\n1,1,0,0,0,\n $t0,$t1,$t2,$t3,$t4,\n$x0,$x1,$x2,$x3,\n \\
123 $power,$acoul,$gapp,$gapn,\n$lnn,$lp,$nmaxp,$nmaxn,\n -1,");  \\
124 fclose($fp);
125 chmod("/var/tmp/test.inp", 0777);
```

The function in line 120 *fopen* opens a file */var/tmp/test.inp* in rewrite mode. It means the all data in file are erased every time after opening. The next function *fputs* writes to the file */var/tmp/test.inp* the values and parameters from environment variable and the data from the web page. Lines 121 to 123 define the variables from the web page and also some constants needed to execute the program. The function *fclose* in line 124 closes the file handled with *$fp*. On some system the configuration is needed to set the rights to the file. This is done by function in line 125 *chmod,* number *0777* means full access to file for all users.

Now we need to execute the program with the prepared input file:

```
138 passthru("/var/tmp/a.out < /var/tmp/test.inp");
```

Function *passthru* executes the program at server as a normal user, one difference is that all messages from program are redirect to browser.

This method of executing of programs is also used to present some results as graphical plots. To do this we choose freeware program called *gnuplot* [4].

This program can by configured by special scripts. Script can be modified dynamically as normal text files like the one used before. We choose two

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 07:00:25

formats of plot. The first is *"png"* for viewing to all graphical web browsers. The second format is "postscript" for sending by e-mail.

Here is an example of code of preparing script files for *gnuplot*:

```
126 $fg=fopen("/var/tmp/dwa3.gnu", "w+");
127 fputs($fg, "set terminal png color\n set output 'r2.png'\n \\
128 set multiplot \n set title 'Calkowita gestosc pierwiastka \\
129 o liczbie neutr.=$lnn, prot.=$lp o s.SkyrmaSk$Sk'\n \\
130 set xlabel \"r[fm]\" \n set ylabel 'nukl/fm^{3}'\n plot \\
131 '/var/www/html/KFT/baza/za.den' u 2:$gp t 'rho-prot' w l  ,\\
132 '/var/www/html/KFT/baza/za.den' u 2:$gn t 'rho-neut' w l ,\\
133 '/var/www/html/KFT/baza/za.den' u 2:$gm t 'rho-mass' w l ");
134 fclose($fg);
135 chmod("/var/tmp/dwa3.gnu", 0777);
```

A description of how to use *gnuplot* is included in *gnuplot* documentation. File */var/www/html/KFT/baza/za.den* was created by program */var/tmp/a.out* executed in line 138. After that one has to run *gnuplot*:

```
139 passthru("/usr/bin/gnuplot < /var/tmp/dwa3.gnu")
```

One can create the postscript file and send by the PHP function *mail* which uses program usually installed on Unix based system *sendmail*.

b) The second method is serving all variables as parameters to the program

We can also compile the program before it will be executed:

```
passthru("gcc program.c -o program.out -include functions.h");
```

Program *gcc* is a free c++ compiler, *program.c* is a source code of *program.out*, *functions.h* is a file with functions headers can be easily modified by PHP script. But compiling in the fly needs good secure tricks from programmer.

This method is the simplest safer than the previous one, but the code is less clear.

3.2. Showing plots at monitors screen

Now one knows how to create script files and how to execute programs with these script files. The next step is description how to display plots at monitor screen by the web browser. Our platforms are accessible for all users from Internet. We choose *"png"* format because most of the web browsers can view this format and this format has loss-free compression that is needed for the best

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 07:00:25

quality of plots. To display a plot we need also two additional web pages *o2.php* and *obrazek.php*. These files will be described later.

When the form is submitted from a graphical interface (Fig. 1), the series of instructions is executed, below there is a fragment of PHP code skipped early:

```
140 if($start):
141 header("Location: o2.php?id=$wartekran&ii=$wykekran
142 &aa=$wykhalo");
143 clearstatcache("/var/www/html/KFT/baza/for011");
144 endif;
```

The *if* condition in line 140 checks a boolean value *$start* and the next lines are executed only after button "start" is clicked. Functions in lines 141-142 open new web page *o2.php* by sending three new variables *id, ii, aa* having the values *$wartekran, $wykekran and $wykhalo*. Function in line 143 *clearstatcache*, cleans static cache system memory in a file */var/www/html/KFT/baza/for011* that is needed not to display out-of-date results.

Nearby opened web page *o2.php* is responsible for displaying plots in *"png"* format. Below we show a fragment of the code:

```
145 if($ii==1)
146  {
147 echo("<IMG SRC=obrazek.php?obrazek=0>");
148  }
149 if($aa==4)
150  {
151 echo("<IMG SRC=obrazek.php?obrazek=1>");
152  }
```

If one of the sent variables will have the value *$ii=1* or *$aa=4*, which can happens if the user select adequate options at the graphical interface (Picture 1), then the functions in lines 146-148 or 150-152 will be executed. These functions display a picture created from the web page *obrazek.php* and the variable *obrazek* will have value *0* or *1*. From this code it is visible that in viewing pictures also the web page *obrazek.php takes part. B*elow the corresponding part of the code is presented:

```
153 $obrazek_t['0'] = "/var/www/html/KFT/baza/r2.png";
154 $obrazek_t['1'] = "/var/www/html/KFT/baza/rhalo.png";
155
156 header("Content-type: image/png");
157 $im = ImageCreatefromPng($obrazek_t["$obrazek"]);
158  ImagePng($im);
```

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 07:00:25

In lines 153 and 154 there is created an array *$obrazek_t* with two elements "0" and "1". Element "0" corresponds to the early created picture */var/www/html/KFT/baza/r2.png* and the element "1" to another picture */var/www/html/KFT/baza/rhalo.png*. Function *header* in line 156 informs the web browser that the created element is in the *"png"* format.

In line 157 the function *ImageCreatefromPng* creates from the array *$obrazek_t* the variable *$im* with the selected picture and the last function in line 158 displays the picture at screen.

4. Summary

We can use the mechanisms shown above to prepare servers, which can contain many specialized programs like those written in Fortran 77, C, C++ and other which have the text output file. Access to the programs will be possible through graphical interfaces. By using this kind of solutions one can distribute many programs whose maintenance is complicated and whose visualization of results takes a lot of time. There is a possibility to join this mechanism with the database system, which will make it possible to find appropriate programs.

References

[1] Fortran program wrote by members of Department of Theoretical Physics in Lublin.
[2] Pfaffenberg B., Karow B., *HTML 4 – Biblia*, Helion, (2001).
[3] Official web page of PHP language - http://www.php.net
[4] Official web page of program gnuplot - http://www.ucc.ie/gnuplot/gnuplot-faq.html

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com