Programming GUI Independently of Window Systems

KAWAGUCHI, Yuuichi*

(Received 30 NOVEMBER, 1998)

Abstract

To make application programs friendly, GUI is effective. There are many GUI systems for building GUI. They usually depend on some computing environments, *ie.*, window system, operating system, and so on. This paper aims to show the way to build GUI independently of them. This paper shows that the combination of HTML and CGI can be a solution.

1 Introduction

Recently, most of application programs use GUI (graphical user interface). A few old fashioned applications still use non-graphical, *ie.*, character-only, user interfaces. In some cases, the success or failure of GUI of application programs have serious effect on their sales, independently of their inner systems. In the case of programming application programs, GUI is as important as the inner system of applications.

GUI is a interface between an user and an application program. It provides "look and feel" to application programs. The inner system of application programs interact with their users through GUI. GUI system is a programming system that constructs a GUI. In general cases, GUI systems supply some components to construct GUI. Windows, frames, menus, buttons, scroll bars and so on are instances of components. Thus, GUI systems usually are constructed with a basic window system. If window system changes, then GUI system and GUI may change or be seriously influenced.

Some operating systems, such as Microsoft Windows series, include GUI system in themselves. In this case, the word "operating system" and the word "GUI" have equal meanings. Some operating system, such as UNIX operating system, are independent of GUI systems and window systems. In the case of general UNIX, they provide `sh` or `csh` as an user interface. This is a character-only interface. Recently X window system is a standard GUI system for UNIX.

In any cases, programmers of application programs must obey each specification of each GUI system, and such specifications are heavily depend on GUI system. Therefore a GUI written for one operating system hardly ever run other GUI systems. This causes a problem. In general cases, vendors supplies application programs for some different operating systems. Programmers are forced to write programs for each operating system with many different GUI systems and window systems to implement one GUI. This is very insufficient.

This paper aims to show the way to construct GUI independently of GUI system, window systems and operating systems.

2 HTML

For more details, please see [1].

HTML (hyper text mark-up language) is a language that is used for publishing web pages to the Internet. It is developed as the derivative language of SGML (standard generalized mark-uplanguage)[3]. HTML originally aims to denote the logical structure of pages, *ie*, headings, paragraphs, items, and so on. Looks of

*Associate Professor: Department of Computer Engineering

pages, such as centering, boldface, font-sizes, were not considered. It was left for rendering softwares, *ie*, web browser, to decide looks of the pages. Authors of web pages, however, has demanded HTML functions to render his/her pages. Responding these demands, developers of web browsers provided their unique extensions to HTML. Since the extensions did not keep the specification of HTML, there ware some cases that people could not see web pages according to their browsers. Recently, the specification of HTML is updated to HTML 4.0. It includes the functions specifying looks, so those problems are being solved.

Web browsers render web pages denoted by HTML. HTML does not depend on computers, operating systems, window systems. Web browsers are depend on computing environments, however, web browsers are provided to many environments. Therefore, if such browsers keep the specification of HTML, authors of web pages need not to consider in which environment his pages are rendered, including looks. Browsers running on different environments may render web pages in the same way. Thus, HTML and web browsers provides a publishing environment that is independent of computing environments.

HTML provides functions to construct GUI. I use the functions in this paper. Since HTML is independent of computing environments, GUIdenoted by HTML is also independent of them. Each command specifying GUI is like "there is a button" instead of "place a button in this way." It is left for web browsers how to render GUI components into looks.

This section explains how to write GUI in HTML, and how to connect them and application programs.

3 Tags

In general, text in web pages is shown in web browsers as it is written. Inter-word spaces and line feeds are, however, changed. If there are more than two spaces between words, then they are put into one space. Line feeds are simply ignored. Web browsers decide where lines are broken, rather than authors.

Table 1: Tags and Their Meanings (extracts)

Start Tag	End Tag	Description
<HTML>	</HTML>	HTML document
<HEAD>	</HEAD>	definitons
<TITLE>	</TITLE>	the title
<BODY>	</BODY>	beginning of the body
<H*n*>	</H*n*>	headings. $n = 1, \cdots, 6$
<A>	</A>	a hyper link
<EM>	</EM>	emphasizing text
<UL>	</UL>	the list of items
<TABLE>	</TABLE>	tables
<IMG>	none	images
<FORM>	</FORM>	input forms
<INPUT>	</INPUT>	GUI components
<SELECT>	</SELECT>	GUI components

There are special letters, < and >, in web pages written in HTML. Text surrounded by < and > is called "tag." Tags specify special functions to web browsers. Table 1 are the list of some tags used by HTML. Some tags have end tags. In such cases, if <ABC> is a tag, then it is called a start tag and </ABC> is a endtag. Text surrounded text by start and end tag is affected. Other tags have only start tag, and works to a next one word. Note that all tags are case-insensitive.

Web pages are written as Fig. 1. Note that → stands for not breaking the line. It is broken for the sake of the short width of the paper.

The <A> tag is called anchor tag. In the Fig. 1, the tag shows a hyper link from the text "Computer Engineering" to `http://www.jo.tomakomai-ct.ac.jp/students.html`. When an user clicks the text with mouse, then he/she goes to a web page specified by `http://`$\cdots$. This way to specify linked pages is called URL (uniform resource locator). URL consist of three parts: (1) the protocol, (2) the host name of the web server, and (3) the path to the page.

All web pages are controlled under web servers. Using URL, web browsers request to a server to send the page. Typical protocols are `http` (hyper text transfer protocol) and `ftp` (file transfer protocol). To get web pages writ-

```
<HTML>
<HEAD>
<TITLE>A Sample Page</TITLE>
</HEAD>

<BODY>
<H1>Chapter One</H1>
This is a <EM>pen</EM>.
<IMG SRC="flower.jpg"→
   ALT="red rose">
<A HREF="http://→
   www.jo.tomakomai-ct.ac.jp/→
   students.html">
   Computer Engineering</A>
</BODY>
</HTML>
```

Figure 1: A Sample Web Page

ten in HTML from servers, `http` protocol is used. The host name part can be FQDN (fully qualified domain name) or a short name. In general, the file name part of the path must end with `.html`.

4 CGI

Section 3 shows some tags to display information. Authors, however, are not satisfied with them. They want to interact with users. They want to publish web pages that dynamically change according to responses from users. In order to make web pages interactive with users, a special method is needed. With the method, web browsers send responses from users, via a web server, to application programs. Application programs send back results of a processing, via web server, to web browsers, and then web browsers render the results into looks to show them users.

To make users and application programs interactive, a method called CGI (common gateway interface) is provided (Fig. 2). An application program using CGI method is called CGI command. CGI method itself does not specify any programming language to program CGI commands. It specifies the way to invoke CGI commands and the mechanism that passes arguments to CGI commands.

When results returned from CGI commands reaches a web browser, then they are rendered into a new page.

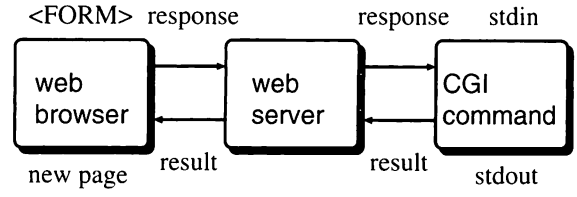

Figure 2: Browser, Server, and CGI command

4.1 Perl

I said that CGI does not specify any programming language, however, usually Perl[2][1] is used to program CGI commands. Perl provide many useful and high-level functuions, especially for processing text. In many cases, programs in Perl are shorter than programs in C. Perl is a interpreter language. In some cases, programs in Perl is slower than programs in C. Programs in nterpreter languages runs on interpreters. If the interpreter of the language is provided, then programs can run on it without re-compiling. This is one of the good points of interpreter languages. There are many interpreter languages in the world such as JAVA[4]. Perl are chosen since it is stable and functional.

Detailed explanations for Perl are beyond this paper. In the rest of this paper, I show programs in fragments and explain them if needed.

4.2 Kanji Characters

I am writing this paper in English. I and perhaps readers are Japanese. Japanese people usually use Japanese letters, *ie.*, Kanji and Kana. In this paper, the word "Kanji" stands for all Japanese letters, which are not encoded by ASCII coding system.

Because of historical reasons, there are three major coding systems for Kanji characters, *ie.*, JIS, EUC, and Shift-JIS (or MicrosoftKanji). They are major on a equal level in Japan. They

[1]The opinion what the origin of Perl is is divided. "Practical Extraction and Report Language" is one opinion.

all represent one Kanji character as a two-byte sequence. Please see [5] or [6] for detailed descriptions.

JIS coding system is usually used for authoring web pages that include Japanese letters. It follows international standard coding system for Kanji characters iso-2022-jp. It is consistent with other national languages. It is also used in sending e-mails via the Internet. JIS coding system uses escape sequences such as `ESC$` to change coding systems in text.

EUC coding system is used for CGI commands. Escape sequences used by JIS coding system sometimes interfere with CGI commands. EUC does not use escape sequences if it is used in simple way. Shift-JIS coding system also does not use escape sequences, however, it easily interferes with ASCII coding system. This is a problem. EUC does not interfere with ASCII. Therefore, if CGI commands that does not recognize Japanese coding system can deal with Kanji characters, taking them as two one-byte characters. In such case, of course, some regular expressions such as '.' does not work correctly. Perl has been spreaded widely in the world. There are some japanized versions of Perl[2], *ie.*, they can recognize Japanese coding system.

In conclusion, JIS is used to publish web page, and EUC is used to program CGI commands. There are some softwares for converting them. I usually use `nkf` (network kanji filter). In the CGI commands, `nkf` is invoked and converts input JIS characters to EUC charcters (Fig. 3).

4.3 Invoking CGI commands

Web pages and CGI commands can be linked by `<A>` tag. For example, a CGI command `name.cgi` is linked in HTML as:

```
<A HREF="http://server/path/→
  name.cgi">Link to CGI</A>
```

In this case, when the text "Link to CGI" is clicked, then `name.cgi` is invoked. In the case of an usual configuration, the name of a CGI command must end `.cgi`.

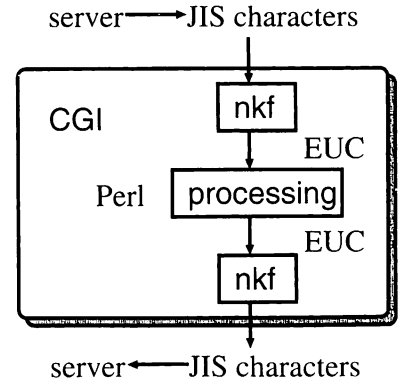

Figure 3: CGI command and nkf

This is very simple. In usual cases, however, other methods are used for invoking CGI commands, since the method using `<A>` tag is not graphical, not user friendly[3], and unattractive. In many cases, `<FORM>` tag is used for invoking CGI commands. For example, the following statements are placed in a web page:

```
<FORM ACTION="http://server/path//→
   name.cgi:METHOD="POST">
 Your Name:  <INPUT/→
   TYPE="text" NAME="yourname">
 <INPUT TYPE="submit" VALUE="ok">
 </FORM>
```

This fragment is rendered by a web browser, and a box where user can input text is placed. The look maybe is like Fig. 4.

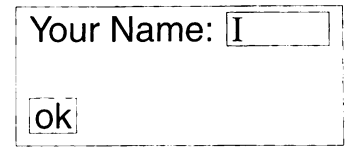

Figure 4: A Input Form for Text

Users can insert one line text into the box placed after "Your Name:." When the button

[2]Sometimes they are called "jperl."

[3]Grphical and friendly have poor relation. Some people, however, say so. I don't know why.

labelled "ok" is clicked, then the text is send to the CGI command `name.cgi` as its argument. In this case, since the POST method is used, the text is passed through the standard input of the CGI commands, and the length of text is send through aenvironment variable `CONTENT_LENGTH`.

Perl can easily deal with both the standard I/O and environment variables. Perl statements to receive those arguments is like:

```
$length = $ENV{'CONTENT_LENGTH'};
$arg = "";
for ($i=0; $i < $length; $i++) {
   $arg .= getc;
}
```

The text box have an attribute `NAME` and its value is `yourname`. Received text is stored in the variable `$arg` in a format:

yourname=*inserted text*

In the *inserted text*, some characters are encoded into a `%` character followed by two hexadecimal numbers. For example, Kanji characters are encoded. Spaces are encoded into + sign. + signs originally inserted are encoded into `%2b` or `%2B`. In CGI commands, these encoded characters should be decoded. In Perl it is easy:

```
$arg ~= s/+/␣/g;
$arg ~= s/%(..)/pack("c",hex($1))/g;
```

If the passing method is `GET`, then passed string is stored in an environment variable `QUERY_STRING`. An environment variable `REQUEST_METHOD` shows the passing method.

After processing arguments, CGI commands send back results to web browsers as a web page in HTML. In sending back web pages, CGI commands pass statements in HTML through the standard output:

```
print STDOUT "Content-type:→
   text/html;\n\n";
print STDOUT "<HTML><HEAD>\n";
print STDOUT "<TITLE>abc</TITLE>\n";
print STDOUT "</HEAD><BODY>\n";
   results of processing
print STDOUT "</BODY></HTML>\n";
```

In the case of specifying web pages by URL, the name of the file ends with `.html`. This tells web servers that it is a web page. In the case of CGI, statements in HTML is sent back through standard output. There is not any file name. Alternatively, the first line of the statements sent back must be "`Content-type···`," and the next line must be an empty line.

`<SELECT>` and `radio` are other GUI components. HTML statements:

```
<FORM ACTION="http://server/path//→
   fruit.cgi" METHOD="POST">
<SELECT NAME="menu">
<OPTION SELECTED>apple
<OPTION>orange
<OPTION>banana
</SELECT>
<INPUT TYPE="submit" VALUE="ok">
</FORM>
```

are rendered as shown in Fig 5.

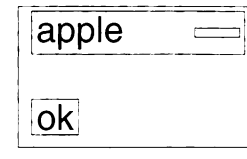

Figure 5: An Example of A Select Tag

Users can choose a favorite fruite out of apple, orange, and banana. In clicking the submit button, the name is sent to the CGI command named `fruit.cgi` in a form `menu=apple`.

If there are two or more `<SELECT>` tags and the value of their attribute `NAME` are `sel_1`, ···, `sel_n`. In this case, the form of passed arguments is:

sel_1=val_1& ··· & sel_n=val_n

where *val_i*'s are selected values.

By using `<INPUT>` tag and specifying radio for its attrinute `TYPE`, radio buttons are rendered. The group of radio buttons that have same `NAME` attribute implements a menu. If one of those buttons is pushed, then others are pulled.

5 GUI Example

This section shows an simple calculator as an example of GUI. The web page for calclator is shown in Fig. 6. The rendered looks is shown in Fig. 7. The CGI command linked to the web page is shown in Fig. 8.

```
<HTML>
<HEAD>
<TITLE>A Calculator</TITLE>
</HEAD>
<BODY>
<H1>A Calculator</H1>
<FORM ACTION="http://server/path/→
   calc.cgi" METHOD="POST">
<INPUT TYPE="text" NAME="one">
<SELECT NAME="opr">
<OPTION SELECTED>+
<OPTION>-
<OPTION>×
<OPTION>÷
</SELECT>
<INPUT TYPE="text" NAME="two">
<INPUT TYPE="submit" VALUE="ok">
</BODY>
</HTML>
```

Figure 6: A Calculator in HTML

In Fig. 6, two `<INPUT>` tags with `TYPE="text"` attributes are used to get numbers from users. Since error check is not done, any invalid number such as `10h` is also sent. A `<SELECT>` tag is used to specify an numerical operator. Users can choose out of four operators, `+`, `-`, `*`, and `/`. Invalid operators, such as `%` and `\`, are not sent to the CGI command.

In Fig. 7, I used the Netscape Navigator for Linux. If another web browser is used, the result of rendering maybe different, but the calculator works as well. HTML is independent of web browsers, window systems, and so on. The calculator made at this time is also independent of them.

In Fig. 8, since any Kanji characters are used, the `nkf` command is not used. It is not needed to convert coding systems. Any error check is done. When an invalid number such as `10h` is sent, the Perl interpreter outputs some error messages to the standard error stream, and no result is sent back through the standard output to the web server. In such cases, a blank page or a page that shows "Internal Server Error" is rendered.

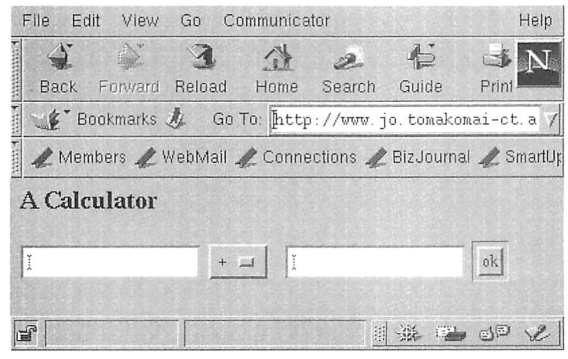

Figure 7: The Rendered Calculator

6 Concluding Remarks

As the aim of this paper, by using HTML and CGI we construct GUI independently of any computing environments. Authors are demanding many useful functions to publish cool and killer web pages. I think it is more important for us to keep the specification, and we should give full play to our creative abilities in it.

Acknowledgement

This paper is based on the school subject "joho kougaku jikkenn" for the J4 class in the latter term of 1998. I thank students in J4 for some innocent and valuable comments in their reports.

References

[1] MARY E. S. Morris, HTML for Fun and Profit, Prentice Hall, 1995.

[2] Johan Vromans, Perl 5 Desktop Reference, O'Reilly & Associates, Inc., 1996.

[3] Nihon Unitech SGML Salon, hajimeteno SGML, Gijyutsu Hyouronnsya, 1995.

```
#!/usr/local/bin/perl

$length = $ENV{'CONTENT_LENGTH'};
$arg = "";
for ($i=0; $i < $length; $i++) {
   $arg .= getc;
} $arg ~= s/+/ /g;
$arg ~= s/%(..)/pack("c",hex($1))/g;
@pair = split(/&/, $arg);
$one = "";
$opr = "";
$two = "";
foreach $i ( 0..$#pair ) {
   @p = split(/=/, $i);
   $one = $p[1] if ($p[0] eq "one");
   $opr = $p[1] if ($p[0] eq "opr");
   $two = $p[1] if ($p[0] eq "two");
}
$result = 0;
$result = $one+$two if($opr eq "+");
$result = $one-$two if($opr eq "-");
$result = $one*$two if($opr eq "*");
$result = $one/$two if($opr eq "/");

print STDOUT "Content-type:→
   text/html;\n\n";
print STDOUT "<HTML><HEAD>\n";
print STDOUT "<TITLE>R</TITLE>\n";
print STDOUT "</HEAD><BODY>\n";
print STDOUT "The Result:$result\n";
print STDOUT "</BODY>";
print STDOUT "</HTML>";

exit 0;
```

Figure 8: A CGI command for Calculator

[4] Patrick Niemeyer, Joshua Peck, Exploring JAVA 2nd Edition, O'Reilly & Associates, Inc., 1997.

[5] Ken Lunde, Understanding Japanese Information Processing, O'Reilly & Associates, Inc., 1993.

[6] Kiyokane Yoshihiro, Suehiro Youichi, International Programming – An I18N Handbook, Kyouritsu Syuppann, 1998.