Information Technology Journal 3 (1): 69-73, 2004
ISSN 1682-6027

Converting Standard HTML Input Controls to Urdu

Muhammad Zaheer Aziz, Muhammad Fayez Aziz and Khalid Rashid
Department of Computer Science, International Islamic University, Islamabad, Pakistan

Abstract: Urdu being a widely spoken language of the world deserves a fair share on the internet. There are many web sites that show information in Urdu using images or character based display but dynamic internet applications accepting input in Urdu and displaying customized web pages are still hard to find. The fundamental requirement for Urdu-enabled web applications are the standard input controls that would work for Urdu in ordinary HTML documents as they do for English. This paper proposes an algorithm to convert the standard HTML input controls to work for Urdu as well. Some essential issues before the conversion process are also handled. These issues include display of Urdu inside input controls, dynamic association of keyboard buttons to Urdu alphabets and use of proper character coding scheme. The algorithm was successfully tested with standard server side scripts which displayed dynamic Urdu text generated by the code or retrieved through connected database.

Key words: Urdu computing, input controls, character encoding, internationalization

INTRODUCTION

Software for Urdu has been a center of attention since early 1980's. Different developers have developed many utilities and applications. The main problem in these software applications is the diversity of standards. Every developer has concentrated on his/her own abilities and available tools. The result being that their product cannot communicate with other products. Data prepared in one Urdu software cannot be imported or used in other available applications. The same problem persisted in case of Urdu web sites. Some developers have contented themselves by making scanned images of the Urdu text and adding the images to the web page for display of Urdu contents. Some others who prefer character-based pages developed their own set of rules and tools for display of information in Urdu. A careful survey has revealed that there are no significant Urdu web sites that allow input in Urdu language using internationally accepted standard mechanism. A fundamental requirement of a web application is input from the user so that a server side script could create a dynamic web page according to the given input. This lack of data exchange standards and unavailability of input controls have kept the development of Urdu web-application behind.

The most important feature of the required Urdu-enabled web input controls is to work in ordinary HTML documents and allow input not only in Urdu but English as well. The widely used input controls in HTML forms are edit-box, text area, list box, check box, radio buttons and submit/reset button. Each of these input controls has to be modified such that they become able to accept user input in Urdu in addition to English. This paper presents the issues in design and development of Urdu-enabled HTML input controls that would work according to internationally accepted standards. Algorithms and other necessary solutions are developed to satisfy all requirements of the problem.

Importance: Urdu is a widely spoken language in the sub-continent and due to immigration on large scale to developed countries; it is also popular in other areas of the world. Literacy rate is extremely low in our country. Further, a small percentage out of the literate population is fluent or comfortable with English. The oceans of information and knowledge available through the internet are of no use while the major part of our population cannot understand it. Few efforts have been made by different developers to display Urdu material related to news, information and entertainment in web pages but almost all of these web sites concentrate on displaying static web pages. Most of these websites are displaying bitmaps of Urdu text instead of character-based display. The main requirement for bringing Urdu in the main stream of Internet is to use Urdu in dynamic web applications with same ease and efficiency as that of English.

Existing systems: Focus of this paper is upon internet applications in Urdu hence only web related products, currently existing, will be discussed here. We can divide

Corresponding Author: Muhammad Zaheer Aziz, Department of Computer Science, International Islamic University, Islamabad, Pakistan E-mail: mzaziz@yahoo.com

existing web applications for Urdu into four categories. The first category includes those web sites that display information in Urdu using images. For this purpose pages of Urdu documents are scanned and converted into images. These images are included in web pages as pictures. Some examples of such web sites are that of Urdustan (www.urdustan.com), Urdu Point (www.urdupoint.com) and Urdu Classic (www.urduclassic.com). In these sites the documents of Urdu are displayed in form of embedded bitmaps. Another website that can be placed in the same category is that of English-to-Urdu dictionary (www.UrduWord.com) in which bitmaps of individual alphabets are mapped to related characters. These pieces are joined together to write an Urdu word against a given word of English.

In the second category we place those web sites that display Urdu text as joined characters using an Urdu font. Some web sites that lie in this category are that of Jang newspaper (www.jang-group.com/jang) and daily Jasarat (www.jasarat.com). These are news reporting web sites. They use some word processing software to type Urdu text and put it into web page as non-English characters. Urdu font is used to give these characters the shape of individual Urdu alphabets. These sites work using proprietary tools and technique of a Pakistani software house named Pak Data Management Systems (www.pakdata.com). Although the text display of Urdu is very good but the encoding scheme used is not a globally accepted standard. The text is written using an ActiveX control and the user has to download their customized font in order to see the Urdu contents on the web page. These web sites do not have any input facilities in Urdu language.

We put the web sites displaying Urdu as character based data using internationally accepted coding scheme and font standards in the third category. Such web sites enjoy the feature that any change in application software platforms does not affect them. These web sites display Urdu text in standard multinational coding schemes such as Unicode (www.unicode.org). The Urdu contents will automatically be displayed on any computer that has support for Unicode. One example of such web site is that of Urdu service of BBC (www.bbc.co.uk). Another example is a web-building software to create Unicode based Urdu sites[1]. The developers of this system have also prepared a Urdu font to work with their software. Input for Urdu data is still missing in this category.

Finally we put the web sites that allow Urdu input in the fourth category. The web sites found so far use ActiveX controls for this purpose. Urdu database with Oracle and ASP[2] is one of the examples. The controls of this system can accept input of Urdu but allow limited editing facility. Similarly a set of Urdu ActiveX controls was developed for input in Urdu in Department of Computer Science, International Islamic University, Islamabad[3]. Such controls do not get popular because of their large size, requirement of permission to execute in the browser from user and being little complex to deal with at programmer level.

Displaying urdu inside input controls: Input controls get their data from two sources. First source is the program that creates them. Initial values are set in the controls by mostly some hard coded statements in the program. Second source of data is the input devices, such as keyboard and mouse, operated by some user. Research has shown that separating display mechanism from business logic keeps the web-application simple and manageable[4]. In the research discussed in this paper, this idea is implemented at lower level of programming and display system is separated from internal logic of the input controls. The internationally recognized standard of Unicode is adopted for the front end of controls. The HTML controls are capable of accepting Unicode characters and displaying them in their proper native shape with help of a Unicode supported font. Hence the conversion algorithm needs to capture the data before it is sent to the display part and replace each character by corresponding Unicode character. This Unicode data is displayed on the control with help of standardized Unicode supported font[5]. This architecture is shown in Fig. 1.

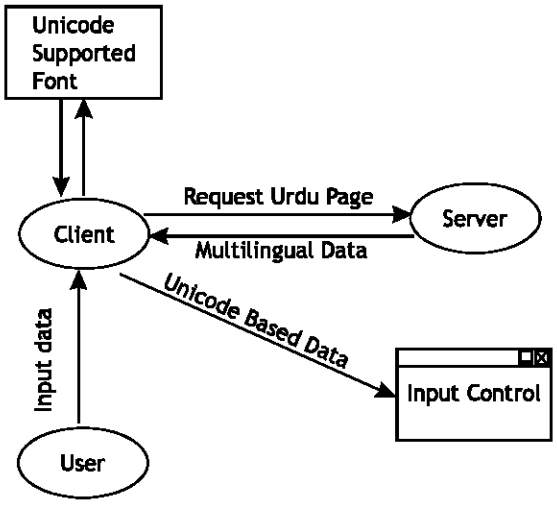

Fig. 1: Architecture of Urdu display in web applications

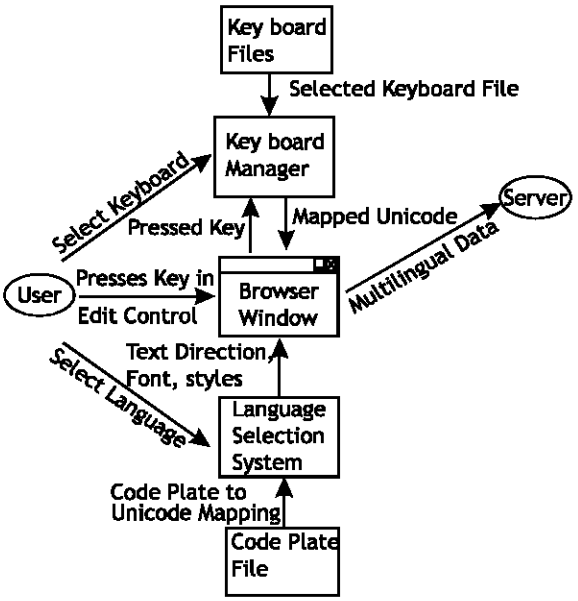

Fig. 2 : Architecture of input control conversion

Role of keyboards: An important issue that needs to be resolved before making an input control to accept Urdu language is the association of keyboard keys to letters of the language. There are many versions of keyboards available for Urdu and each of them claims to be the standard in some aspect. The commonly known keyboards include IBM, MQZ and Phonetic[6]. IBM standard has been in practice since many years while the MQZ standard has recently been developed for improving typing speed. On the other hand a general user feels more comfortable with phonetic keyboard in which the Urdu letter is associated with the closely sounding English letter.

In order to accommodate user's choice, this research proposes to make the keyboard dynamically selectable. Associations between keys and the related letters should be saved in separate files. The file for the selected keyboard should be invoked when a user opts for it and that keyboard should become active right from the next key-press. Role of this keyboard file in the input system is shown in Fig. 2.

Role of code plates: Code plates are universally accepted standards to represent, save and interchange data. Exchange of data between diverse applications becomes possible when all developers save their data according to a common standard. The widely accepted standard for English is ASCII, whereas Unicode is emerging as a multilingual standard. A standard code plate has been developed for Urdu[6] as well but it is subject to modifications for some time before it comes into final shape.

To resolve the issue of varying coding standards for Urdu the best option, that can be proposed, is to allow dynamic encoding scheme. It can be done by keeping the code plates separate from main system instead of hard coding it. The association of characters and their codes will be kept in independent files that could be associated to the main program one at a time. This mechanism will allow accommodating alterations in the encoding schemes and also any new emerging standards. Import and export of data from one standard to the other will also be possible.

Conversion algorithm for input controls: The primary objective of the research discussed in this paper is to convert the standard HTML input controls into Urdu by enabling them to accept Urdu along with English. All commonly used input controls including edit-box, text area, list box, check box, radio buttons and submit/reset button were considered for this purpose.

There are two categories of input controls in HTML. First are those that do not involve feeding of text from user. List box, Check box, radio button and submit/reset button does lie in this category. For Check box and Radio buttons the input values are boolean numbers. List boxes displays a list of strings and user has to select one (or more) from them. The value returned by the control is either the index of selected string or the string itself. Buttons like Submit and Reset only act when they are clicked upon and the only text involved is the caption displayed over them[7]. Conversion of list controls to Urdu requires displaying of Urdu strings in the options and returning the selected string (or its index) when the form is sent to web server. Hence the mechanism discussed for displaying Urdu text in input controls will solve the problem. Similarly the only language dependant part that needs conversion in Check boxes, Radio buttons and Click buttons is their caption, which can be displayed in Urdu using the already discussed algorithm.

The second category of input controls that requires more work in conversion to Urdu are edit box and text area. In these controls user needs facility to enter and edit text while viewing what is being entered. Hence input and display are both involved at the same time. Another associated problem is the keyboard, i.e., which key should be assigned to which letter of Urdu. Similarly another issue is requirement of feeding English and Urdu mixed in the same field. Edit box and Text area primarily accept ASCII codes that are fed into them on each keystroke from the keyboard. In Unicode supported web browsers, these controls are capable of accepting and storing Unicode characters as well. Shape of the inserted Unicode characters is formulated by using the default Unicode font installed in the computer. Hence the task of conversion of

these controls narrows down to devising a mechanism that would insert a Unicode of Urdu range when the users presses a key being in Urdu mode. Pressing of same key should feed the related Unicode of English range while in English Mode. The complexity lies in permitting to allow mixing of two different languages (English and Urdu) in the same edit box. One obvious problem arises when the users clicks in between a text already fed in one language and would like to type in the other laguage.

In order to solve the above-mentioned issue, key and mouse button events were trapped and reprogrammed for the edit control and text area. The HTML 4.0 standard mandates a set of intrinsic events and these are supported to some extent in version 4.0 of both Netscape and Microsoft browsers[8]. Internet Explorer 4.0x features a property that indicates what mouse button was pressed (for a mouse event) and another property that indicates what key was pressed (for a keyboard event). In Navigator 4.0x, one property (which) serves both cases[9]. Here the event of key press will be handled to reprogram these two edit controls. Other internet applications have been developed using key handling like the Doc Dialer which invokes different tasks on press of different keys[10]. Based on the study of available literature, the architecture of the system is designed as shown in Fig. 2. The conversion algorithm has to be configured by selecting a language and a keyboard. Now pressing of a key inside the input control first invokes the keyboard management part of the algorithm. The keyboard manager evaluates the key and replaces it with an appropriate Unicode value using a predefined keyboard file. The Unicode value being inserted depends upon the selected language. In English mode, the value inserted against a key will be picked from the English range of Unicode. On the other hand the same key will be mapped to a character from Urdu range when Urdu mode is on. Insertion of Unicode instead of ASCII automatically forces the browser to display the character in the selected language (such as Urdu). In order to send this data to the server, this Unicode is converted into a 1-byte code according to the standard code plate for the language. When the value of input control consists of text in mixed languages, the system demarcates the languages by a special toggle code. At beginning of text in Urdu, the algorithm inserts the toggle code ~u and for English a ~e is placed. Figure 3 shows a sample of mixed language text and shape of the string formed after applying the Unicode to 1-byte code conversion. This conversion into single byte code is necessary because internal processing of database engines, domain registries and other servers are based on ASCII style data coding.

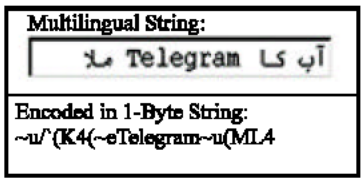

Fig. 3: Sample of multilingual character encoding

As per above description of the conversion algorithm we may summarize its functionality in two components. First is the front end that is related to display of Urdu on the input control. The second component deals the backend in which input of Urdu and English is managed. Figure 2 presents the complete architecture of the conversion algorithm showing all the participants involved and procedure of their interaction. The two components of the algorithm are presented in form of traces. Figure 4 describes algorithmic details of the display mechanism in multilingual Urdu controls. Figure 5 shows the steps and sequence of the part of algorithm that handles internal processing of a converted input control.

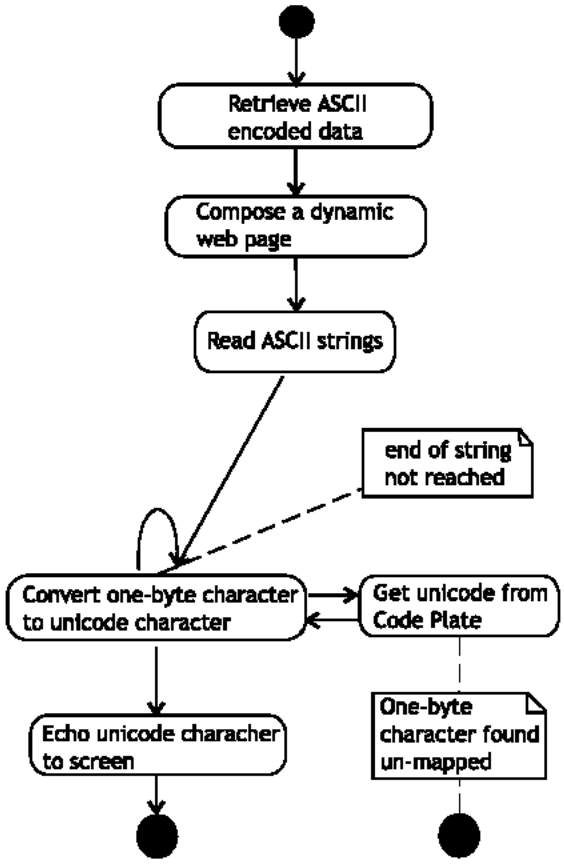

Fig. 4: Traces for displaying Urdu text

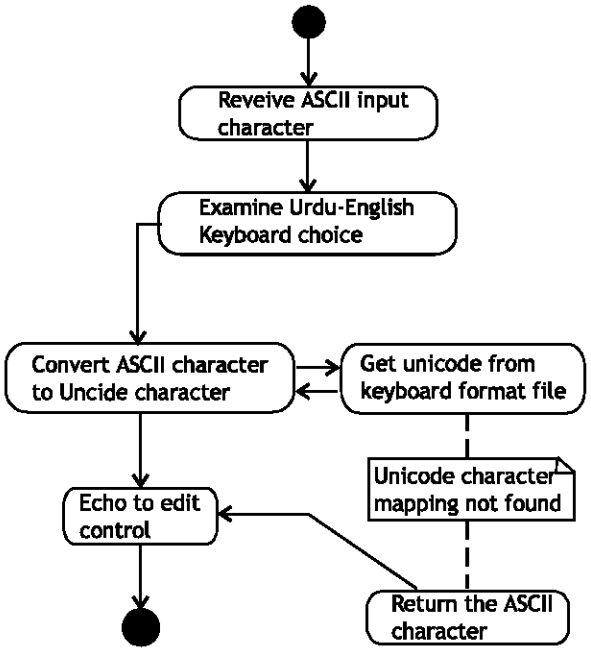

Fig. 5: Traces for input Urdu and English text

The devised algorithm was tested by implementing it in web pages generated by well-known server side scripting language called PHP. The controls were tested by feeding, editing, copying and pasting values in Urdu, English and mixed languages. These values were extracted from the input controls and sent to other server scripts for processing as done normally for English. The Urdu data was also saved into and retrieved from a standard web database engine called mySQL. Hence completeness and validity of the algorithm was thoroughly tested and it was found successful.

After completion of this methodology it has been proved that Urdu can exist on dynamic web application as any other language of the world. The success of Urdu input controls and data conversion algorithms have opened door for web application developers to produce fully automated Urdu applications on the internet. This achievement is along with the advantage that no extra storage space or complex controls will be needed both at client and server ends.

In this research the concentration was focused on the widely used operating system and browser platform of Microsoft. Although the algorithms were implemented in Java Script, which is largely platform independent but still there is room for modification and customization according to different browser platforms. This is because each browser hosts different privileges and features at programming level. A suitable direction for further research in this context could be standard input controls for Unix based web browsers.

REFERENCES

1. Shehzad, A., 2002. Tabish, Urdu Nastaliq Unicode True Type Font, www.arbonet.org/~tabish/u-trans.
2. Irfan, S., M.A. Tahir, S. Tareen, Z. Shehzad and U.H. Khan, 2001. Urdu Database with Oracle and ASP, GenXsol Software Documentation.
3. Qadir, G., M.Z. Aziz and 2001. Urdu ActiveX Controls, MCS Project Report, Department of Computer Science, International Islamic University, Islamabad, Pakistan.
4. Lloyd, H., 2003. Nuts and Bolts of Web Internationalization, 23rd Internationalization and Unicode Conference, Prague, March 2003.
5. Tex, T., 2002. Introduction to: Examples of Unicode usage for business applications, www.I18nguy.com/unicode/unicode-example-intro.html.
6. Mohammad, A., 2002. Overview of Urdu Informatics Standerdization, Conference of AINC, Tunisia.
7. Ivan, B., 2000. HTML, DHTML, Javascript, Perl CGI, BPB Publications.
8. Yehuda, S. and T. Shiran, 2001. Professional Java script, www.webreference.com/programming /javascript/ professional/chap4.
9. Tomer, S. and Y. Shiran, 1998. Button and Key Codes, www.webreference.com/js/column11/codes.html.
10. Yehuda, S. and T. Shiran, 2000. Event Handling, www.webreference.com/js/ column58/.