

ASCII Based GUI System for Arabic Scripted Languages: A Case of Urdu

Bacha Rehman, Zahid Halim, and Mustaq Ahmad

Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Science and Technology, Pakistan

Abstract: From its advent, computer science is facing language-based problems; as English is considered being the most widely used communication language. Arabic Scripted Languages (ASLs) including Urdu have been considered less in this aspect. To rectify this problem, it is decided to build up the idea of interface designing for Arabic Scripted Language (ASL). All the algorithms and methods are used to develop a Graphical User Interface (GUI) for Urdu language, which belongs to the family of ASLs. This work presents a novel idea of designing an ASL interface for desktop application e.g., databases, applied for Urdu language. American Standard Code for Information Interchange (ASCII) codes been used for mapping the keystrokes to the Urdu character's images using Phonetic keyboard styles. Four fonts are created. The Urdu characters contained in these fonts are mapped through ASCII codes. The main goal achieved is to develop algorithm for Urdu desktop controls, which are the primary entity for its interface designing. Proposed algorithm facilitates the ordinary user to work in any ASL applications. Two implementations of the proposed method have been applied in this paper. This work presents a set of methods for new researchers to investigate these techniques for further improvement and for other ASLs.

Keywords: Urdu, GUI, dictionary, ASCII, interface design, ASL.

Received March 12, 2012; accepted February 22, 2013; published online April 4, 2013

1. Introduction

There exists a natural restriction of communication all over the world due to existence of different natural languages. This natural constraint is quite a big problem to meet the new challenges, such as the concept of globalization. From its advent, computer science is also facing language-based problems, because of a general concept that everybody has to deal only in English language. All Arabic Scripted Languages (ASLs) including Urdu has been considered less in this regard. To rectify this problem on large scale, it was decided to take an initiative for this important aspect of communication. The other consideration, which leads to Arabic Scripted Language (ASL) interface designing is the lack of the basic methodologies, algorithms, source code. Due to non-open source policies of different software vendors, the non-existence of general algorithm and source code regarding ASL interface design [22] to the new researchers is also a big issue. This paper provides a complete framework of how to work on user interfaces for ASL. All the methods of this paper have been applied to develop a Graphical User Interface (GUI) for Urdu language; a language which belongs to the family of ASL.

Around one billion people across many countries speak and understand Urdu and it is Pakistan's national language too. "Urdu" is a Turkish word means horde [3] ("شکر"). Urdu fundamentally relates to Indo-

European language, which is a sub branch of Indo-Aryan, developed under Persian, Turkish and Arabic [3]. Urdu is the national language of Pakistan, and it is also a secondary language in many other Asian countries. Urdu script is written from right to left similar to Arabic, Pashto, and Persian language letters.

ASL interface designing is a set of font that is to be mapped with some basic desktop control for an ASL's GUI, based on Urdu language. In this research activity, four basic true type Urdu fonts have been developed, named as UID1, UID2, UID3 and UID4. Eight desktop controls i.e. Urdu textbox, Urdu combo box, Urdu list box, Urdu label, Urdu limited text box (only for numbers), Urdu button, Urdu check box and Urdu radio button have been developed for this research. For the implementation stage these desktop controls are used to develop any sort of GUI for Urdu applications such as databases with a proper technique. The mapping code used between desktop controls and Urdu font is based on American Standard Code for Information Interchange (ASCII) codes [4] and the Urdu keyboard standard followed is Phonetic keyboard style [8].

This paper presents a novel idea, as it is a pioneer endeavour that will assist the new researchers all over the world to develop applications via Urdu or any interface based on ASL. The vital purpose is to set up and uphold the setting, which can turn out to be foundation and launching pad to build new applications facilitated by Urdu. Its basic logic makes

sure that applications can be developed in every language of the world only by designing its fonts and then using it with appropriate algorithms. The methodologies are applicable to all types of applications that may include databases, general desktop applications and web based applications etc.

The objectives have been defined while considering all work been divided into three phases i.e., develop Urdu fonts and assign mapping code, designing desktop controls, and implement desktop controls in library and dictionary systems for better explanation, elucidation and arrangement purposes.

The following specific objectives are kept in mind while developing the desktop controls:

- To create an environment for Urdu interface design.
- To use basic tools for their common usage but using Urdu interface.
- To facilitate the user with database tools, which can be handled by using both the language in different fields of the same table.
- To interchange and pass the data through different tables handling different language at the same time.
- To write down Urdu text in Urdu style in Urdu textbox.
- To create the basic desktop control, which can work with the same setting, environment and property settings etc.

The features related to implementation perspective that what kind of end application would be there in the end user mind includes entry of record in Urdu language, less time requirements to serve the member, less time to search location of the of a record, less labour requirements if implemented, data security assurance of the record, management of records in separate categories, to reduce redundancy of data with respect to that of manual work and easy modification of data etc.

The main features and benefits can be categorize and summarize as a result of this research work as follow:

- User friendly and provision of complete interface.
- Environment for Urdu interface design.
- Use of basic tools for their common utilization in using Urdu interface.
- Facility of interchange and pass the data through different tables.
- Write down Urdu text in Urdu style in Urdu textbox using common usage fonts.
- The basic desktop control, which can work with the same setting, environment and property settings.
- Merely changing font can change text formatting back to English.
- Integrated development environment.
- Time saving and reliability.
- More flexibility for future enhancement.
- Complete modular approach.

It is decided to work on this research project as an extension to our work, presented in [18], by covering three milestones. Section 2 describes some of the related work done in this regard. Section 3 covers the idea to create Urdu fonts and giving every font a specific mapping code, which has to be used for later interaction with the desktop controls. Section 4 is related to the development of self sufficient desktop controls, which can be utilized at any platform and can be integrated with any kind of software. Section 5 lists the implementation and testing of all the eight developed desktop controls, used in Urdu library and dictionary systems for better understandings. Section 5 presents future work. Finally, section 6 presents the conclusions.

2. Related Work

Our literature search reveals that very limited work on this concept has been carried out. However, we have briefly reconciled this paper with some very close related work and present Urdu software. In this regard we can mention the work done on interface designing in other language especially in French language [17], while developing first non-English oriented operating system. Further, in [2] an interface design is illustrated for database developed in Arabic. In this work the details of a comprehensive database of Printed Arabic text for Arabic text recognition is presented. The main mapping code and font type used in this research activity is based on Unicode standards. Similarly the same scheme has been adapted for development of an Urdu programming language in [5].

There are many types of software that are being used for Urdu language specially concerning text editing of Urdu language such as Urdu Inpage, KATIB, pares Nagger-URDU, Power Inpage, SADAF95, Surkhab, Pars Nigar 2.0, Urdu lead mark, Urdu plug-in, Shahkar, Jang Urdu Fonts, Urdu Page Composer, Gamma Uni True Type [15], etc. These all are quite efficient in their respective task to facilitate the user with Urdu text editing and formatting. There are some software which have different font types, formatting methods and other such basic things but excellent methods to meet the needs of the end users are missing.

Here are few mentioned constraints, which are present in the earlier Urdu software. Efforts are applied to overcome these deficiencies and constraints in this work:

- All software was just for editing purpose for the ASL text but no software was available for other fields of interest in computer field [19] i.e., that presents interface to interact with databases.
- There has not been any significant desktop control and any other controls to fulfil the job of Urdu interfacing [17].

- There has not been any endeavour to develop any logic, which can become base to work in any ASL.
- There exists hardly any work that enables to perform searching mechanism on Urdu language [19].

3. Urdu Font Development

To achieve this goal, we began with development of Urdu fonts [16]. Development of an Urdu font is a tough job to undertake. The primary purpose of selecting the ASCII code as a replacement for Unicode is that ASCII codes is sufficient for the proposed work as all the 39 alphabets of Urdu (including zair and Arabic numbers etc.) can be easily mapped in available 256 ASCII codes. It is also better for implementation prospective as it takes 8 bits space (almost half of Unicode). The other thing which leads us to use ASCII codes is that we can achieve ASL interface designing and successfully can interact with databases and also can use it in other applications without changing the default settings in the operating systems, which are done for the ASCII codes by default.

The basic inspiration related to the separation exists in different type of characters and different positions for each Urdu character are taken from [3]. Urdu language alphabets can be separated into two types (connectors and non-connectors). Moreover, the mentioned types of letters always depend on maximum of four positions for the entire Urdu letter. These positions are initial, medial, final and independent respectively. For instance, Table 1 exemplify all the four positions for Urdu letter “Bay” (ب).

Table 1. Position of urdu letter bay (ب).

Independent	Final	Medial	Initial
ب	ب	ب	ب

The phonetic keyboard standards layout is followed for mapping the Urdu character’s alphabet with ASCII codes for ease and conservative routine. It is the simplest keyboard layout for all the users, as it is very much similar to the ordinary QWERTY keyboard. Figure 1 illustrates the phonetic keyboard layout [9].

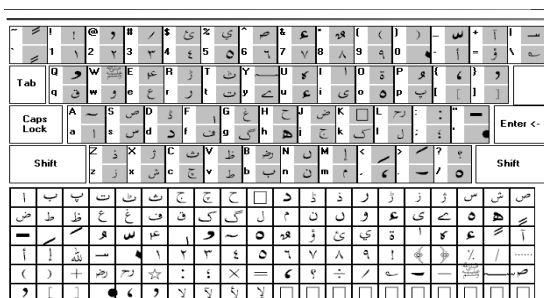


Figure 1. Phonetic keyboard layout.

In Figure 1 the entire thirty nine Urdu characters (alphabets) have been mapped to the subsequent QWERTY keyboard with the phonetic standards

(layout). The available ASCII codes are 256 [4], in which 52 ASCII values have been assigned to the independent position of every alphabet as present in the phonetic keyboard style. The mentioned 52 ASCII values are from 65 to 90 and 97 to 122. For Urdu numbers, ten ASCII value have been assigned from 48 to 57. Twenty eight Urdu characters (ب ت ش ط ی etc..) exist in all the four positions mentioned in Table 1. Further, 10 Urdu characters occupy only independent and final positions (ا د ر و ز ن ژ ڑ ڈ) and finally 1 Urdu character occupy the independent, initial and medial position (ء). The rest of the ASCII codes have been assigned to mapping code, which maps the other positions of the same Urdu alphabets’ independent position. Six ASCII codes have also been assigned to zabar (-), zair () and paish (’) etc. This information shows that a total 173 ASCII codes are been used for all Urdu characters, numbers and zair etc.

As mentioned earlier that the default ASCII codes in phonetic keyboard layout is used for the independent Urdu characters’ positions, for example 97 for ا, 41 for آ. Figures 2 and 3 illustrate the methodology that how to create font image for every position of a single Urdu character and finally assign a mapping code [16] for every Urdu character images by mean of ASCII standards.



Figure 2. A view to create urdu letter bay (ب).

Figure 3 illustrates the assigning of mapping code to all Urdu letters positions Table 1. The mapping code is the key for communication between the desktop control and the Urdu font. In the code it has been decided to retrieve the ASCII value of the Urdu font character first. Then to adjust the position of the Urdu character, unique mapping code for all positions have been assigned. The position of the Urdu letter depends upon the preceding and following Urdu character [3].



Figure 3. A screen shot of software for assigning the ASCII code for mapping.

Identical mapping code have been assigned to each corresponding Urdu character for use in all available supporting operating system platforms (Apple Default, Microsoft and Macintosh) as shown in Figure 4. In addition, we also tested the developed fonts on Linux operating system, which works fine. During the development of desktop controls, the corresponding mapping codes have been accessed which are maintain in two separate arrays to all the array's elements (Urdu font array or lookup table) that is to be discuss in next section.

4. Urdu Desktop Controls Development

Desktop controls (radio button, text box, combo box and labels, ..., etc.) are the set of software components or module [14] that can be used to interact with the end user, irrespective of the programming language in which the components are produced. The best autonomous and user defined desktop controls bring into play in Microsoft platform is called activeX controls [16]. These controls demonstrate an influential user interface in any development platform. Proposed algorithms and methodologies are applied through activeX controls technology but the main logic (Urdu class code) can be used in any environment. For instance, we can instantiate objects of desktop control from the main class code in any platform. ActiveX technology is been used because it provides a user friendly interface for both desktop and web applications [6]. It also provides setting for Urdu interface design and by simply changing in its font, one can transform the formatting of text back to English or system's default language's font.

For a simple explanation, the main idea is shown in Figure 4. This figure illustrates the working of the algorithm of how to assign the corresponding Urdu character to every English character, present in the English word. Further how to reverse and finally combine the Urdu characters to form a full Urdu word by finding the exact position in the font array.

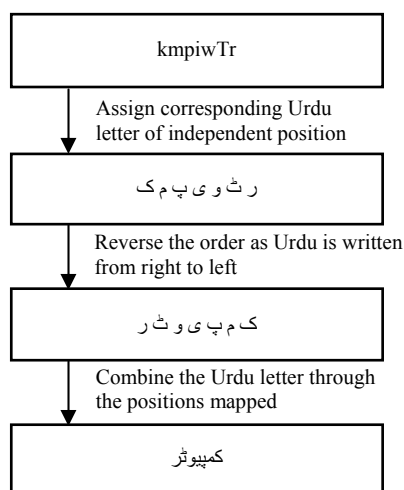


Figure 4. The basic convergence logic.

The complete idea of working of an Urdu interfacing designing is shown in Figure 4. For clarification, let the Urdu word کمپیوٹر (means computer) is written in phonetic keyboard standard by pressing the keys according to QWERTY keyboard are "kmpiwTr". In the initial process, independent positions' mapping link is assigned in Urdu character's ASCII code for every English character. In the following process the order of Urdu independent positions are reversed because Urdu's script is written from right to left. Then in the final process, the algorithm assigns the exact font character [12] by accessing the mapping code of position in the Urdu font by font array.

Figure 4 illustrates the desired logic to be achieved. For instance, we type or access from a database or file, the word "kmpiwTr" and the algorithm display the Urdu word "کمپیوٹر" on configured desktop control. The goal can be achieved with a suitable font which has assigned the same ASCII codes for Urdu characters, which are being used in the font array by employing phonetic style. The second entity required is the logic for both combination and separation of the Urdu characters [13] (find the proper position Table 1).

The main algorithm of converting the default English characters to corresponding Urdu characters is illustrated in Figure 5, and further elaborated parts of Figure 5 related to the Urdu convergence process is shown in Figure 6. Figure 7 illustrates the process of how to combine Urdu characters or we can say how to assign the exact position.

The major logic illustrated in Figure 5 is to get the ASCII value of the keystroke (pressed by user). Then, match the ASCII value in the Urdu font which is linked to the desktop control by the font name property. Finally display the position of Urdu character by matching it with the lookup table into the Urdu font.

There are six processes working in parallel as shown in Figure 6. These bunches of processes are the elaboration of the process related to Urdu convergence as shown in Figure 5. Figure 7 illustrates two types of flow, first flow is for English characters and the second is for Urdu characters. In first flow (English characters convergence), the desktop control's elements are reversed. Then each element has assigned the independent position's Urdu characters based on the primary mapping link from the Urdu fonts. At last the algorithm searches the correct position for the Urdu character based on its actual position in the word and finally returns the Urdu string (word or sentence) to the desktop control. The same procedure is also followed for converting Urdu characters to English characters but in reverse procedure.

In Figure 6, the first flow represents the procedure of extracting the English characters stored in the database. Actually for implementation in database, it

does not change anything in the database. The main trick applied to the way the desktop control display the Urdu text. For example an English word stored in a database field is “ktab,” when the first flow applied on this particular database field, the desktop control will display it as “کتاب”. The reverse procedure applied for converting Urdu characters back to English characters is to store the contents of the desktop control in the database. For online changes during writing into the desktop control these processes can be applied on-change events etc.

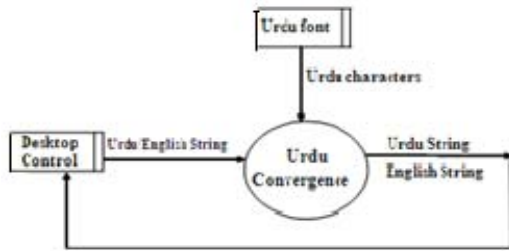


Figure 5. Main convergence logic.

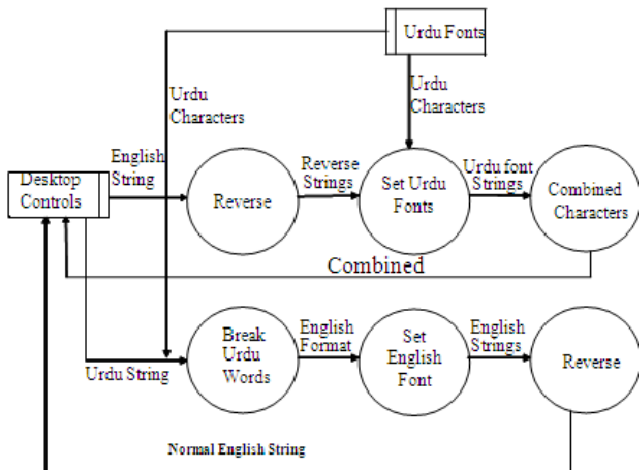


Figure 6. Detailed convergence logic.

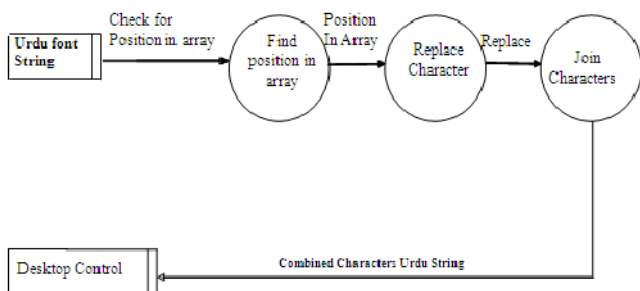


Figure 7. Elaborated combine character process.

The desktop control display is basically a converted font image of the characters. The main ASCII code for independent position remains the same. This technique also can be applied to encryption. For example the stored word for Urdu word “کتاب” (means book) in the database is “ktab”. So, a person, who does not know the whole system, can never know what “ktab” means. Further, the developed fonts are true type and it can also be assigned to the database table’s field. But it will

store only the independent position of every character, for example the same word “کتاب” will be stored in the database like “ب ا ت ک”. Because the combine characters process is not applied inside the structure of database in this paper.

Figure 7 shows detail which presents the combine characters process as shown in Figure 6. In Figure 7, a font array is being utilized to build up the matching algorithm. This process is working within the scope of Urdu desktop control.

On the foundation of information in these figures, certain onset strategies are laid down for employing the font array and code. It is first decided that the font array (lookup table) should control and perform the assigning mapping criteria as was allocated to the every Urdu characters, present in the Urdu fonts.

This methodology is being adopted for the simplicity of developing the code. Further it allows the compatibility with the operating system platforms under the use and facilitates application development. Following are the major font arrays and functions prototypes used in Urdu module/class. This Urdu class can be allocated to any type of desktop control which is supposed to be using the ASL fonts and font array.

Below is the main algorithm for developing Urdu class, which can be applied to any desktop control using ASL fonts and font array. The following two arrays are maintained to cover all the Urdu characters Fontarray [100] [4] “It is the main two dimensional font arrays (lookup table). The first dimension of size 100 represents the main ASCII code which is assigned to the independent position of every Urdu character, including Urdu numbers. The second dimension of size four is set for all the positions of a single Urdu character and stores the mapping code of all the remaining positions of that Urdu letter. Position for mapping code set to -1 for the empty position for those Urdu characters with two or three positions (i.e., ز, د, ر, ... , etc.) available.”

Aziz’s algorithm [6] “For zair, zabar, pesh and some other special characters” for developing the font array is given below:

Start

1. Create_fontarray // lookup table for Urdu characters

2. N = 52

3. “For alaf and starting from A up to Z”

Fontarray[1][1] = 97

Fontarray[1][2] = -1 “alif “only have two positions”

Fontarray[1][3] = -1

Fontarray[1][4] = 163

“B next for bay”

Fontarray[2][1] = 98

Fontarray[2][2] = 133

Fontarray[2][3] = 164

Fontarray[2][4] = 204

And up to Z

4. End of for Loop

5. End

Algorithm for the main function, which does all the convergence is given below:

1. *String_convergence (string)*
2. *Create_fontarray*
3. *Create integers before_status, length, character, t*
4. *Create strings rev_str, temp[10000], first, last, Urdu_string*
5. *Rev_str = reversestring(string)**
 - a. *Length = length of rev_str*
 - b. *If length < 1 then*
 1. *Quit function*
6. *Else*
 - a. *Before_status = 1*
 - b. *Loop for characte = length upto 2 in reverse direction*
Last = previous character
First = current character to be joined with the last
*Join_two_characters (Last, first, before_status)***
Temp [character - 1] = last
Temp [character] = first
t = t + 1
Character = character - 1
 - c. *End loop*
7. *End else if*
8. *Rev_str = “ “ “make it empty”*
9. *Loop for character = 0 upto t+1*
Rev_str = concatenate (rev_str and temp[character])
Character = character + 1
10. *End loop*
11. *Urdu_string = rev_str*
12. *Pass Urdu_string to desktop control*
13. *End*

* *Reversesring function reverse the order of characters in the string*

** *Join_two_characters function takes three input arguments and join two Urdu characters using the mapping code. It also checks the availability of the position of all Urdu characters and finally converts the independent position to its corresponding position.*

By configuring a desktop control by the above algorithms, we can develop the primary entity of any ASL's GUI at any platform. The main purpose of implementation is the intention for developing a quality ASL application that can match the specifications of interface designing.

5. Implementation

After the development of fonts, font arrays and algorithm, the afterwards step is to develop the structures of programs. Program development means to design computerized procedures and functions to accumulate and process data [14]. Eight Urdu desktop controls have been developed and configured [16] as shown in Figure 8. The implementation of the Urdu module has been done for retrieving and storing the data in a database. The end result is exceptional and quite satisfactory, because the logic for convergence converts the Urdu string back to its corresponding English string for storing in the database using phonetic keyboard standards for ASCII. Further it retrieves the English string stored in database to the configured Urdu

desktop control and converts it to Urdu string. Figure 9 shows all the eight Urdu desktop controls which were developed for the whole implementation. These controls include all the necessary types which may be needed for a full Urdu or any ASL's interface design for both desktop and web applications.

Desktop Control	Look
Command Button	
Label	
Text Box	
Limited Text Box	
Check Box	
List Box	
Combo Box	
Radio Button	

Figure 8. Configured desktop controls.

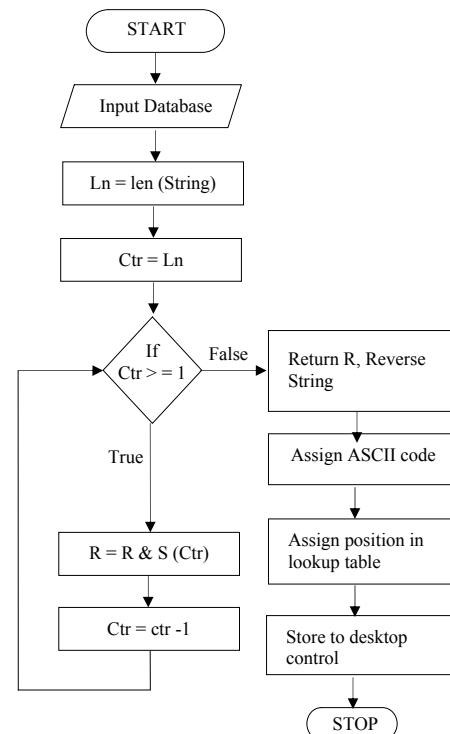


Figure 9. Retrieval of data from database to Urdu desktop control.

These desktop controls include button, label, text box, limited text box (accepts numbers and -sign for dates and numbers), check box, list box, combo box and radio button. The main Urdu module code is integrated with every desktop control, shown in Figure 8. In simple words the Urdu desktop control has

instantiated as an object from the main Urdu code class.

The main idea behind the convergence logic, retrieving the data from the linked database and finally displaying it in Urdu at configured Urdu desktop control is illustrated in Figure 9.

For testing and implementation, the configured Urdu desktop controls are integrated to a library database in the first implementation. An instance of final Urdu interface design is shown in Figure 10. In this interface the book name entry (“پیام مشرق”) corresponds to the base English string “pyam mxrq”.

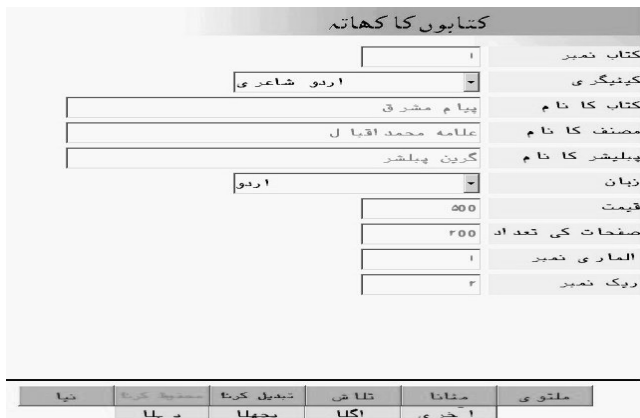


Figure 10. Urdu desktop controls linked to database.

The second implementation of the proposed desktop controls has been done in an Urdu dictionary system. For this purpose, 149466 Urdu words have been accessed from [21] in a dictionary database containing two columns of Urdu and English words. Both checks have been adopted for converting words from one language to another. The deftness of searching the word in Urdu dictionary is very fast as the English backend. It can help to reduce the heavy size of normal Urdu dictionary and also provides modify access to data for adding new words and their respective meanings. The results are shown in Figure 11.

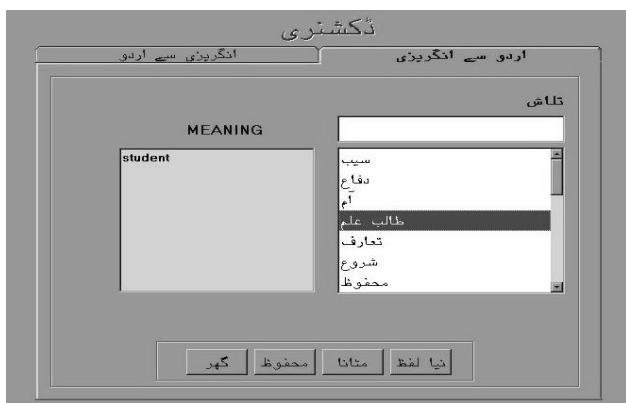


Figure 11. Shows desktop controls linked to a dictionary database.

If we look to the original database entries mentioned in Figure 11, for instance the word “طالب علم” the English text from which it has retrieved is “valb elm,” and its meaning are shown in front to it (means “Student”).

6. Future Work

Nothing is hundred percent perfect and the most permanent thing in this world is the change. Therefore the things which are subjected to change are included in future extension. There are still some things, which can be done in future. It has been ascertained to carry out each step of coding and designing dynamically so that it should have tractability to mould itself with future challenges. The desktop controls have been equipped with all the required properties, but still extra properties can be added easily as and when required. Eight desktop controls have been developed in this work but there are more possibilities for other type of desktop controls too. An endeavour can be ascertained for preoccupation idea to develop modules for other local and foreign languages such as Arabic, Persian, Pashto and Punjabi etc. The feasibility study has been performed which concludes optimistic scenario for this type of future research studies. This work can further be incorporated in a web mining system as in [20] and other tools related to ASL [1, 7, 10, 11, 23].

7. Conclusions

This paper rectifies the problem of communication and less consideration of ASL for interface designing. Thus the language problem for user interface is the primary stone that leads to develop this small but well designed and equipped building. The main objectives have been stated while considering all group of tasks distributed into three parts i.e., develop Urdu fonts, design Urdu desktop controls and their implementation. The major goal achieved is the development of algorithm for Urdu desktop controls which are the primary entity for Urdu interface designing. This algorithm provides facilities to the ordinary user to work in Urdu language’s applications. This work also provides a pioneer methodology to the new researchers to think and further improve the logic used.

This work provides all the main algorithms and procedure which can be adapted for well furnished ASL based software. This is a unique idea that can start a new area of research interests. Some major software vendors also provided software based on Urdu and other ASLs interface, but all the techniques used are classified in majority of cases. Therefore, it is a small but well equipped module by which new researchers and software developers can get the idea of how to develop interface not only for Urdu but also for any ASL. The proposed modules provide one of the best entities for applications like Urdu translation system, huge application using databases and other interface based software.

References

- [1] Abbas Q. and Khan A., "Lexical Functional Grammar for Urdu Modal Verbs," in *Proceedings of International Conference on Emerging Technologies*, Islamabad, Pakistan, pp. 7-12, 2009.
- [2] Al-Hashim A. and Mahmoud S., "Benchmark Database and GUI Environment for Printed Arabic Text Recognition Research," *WSEAS Transactions on Information Science and Applications*, vol. 7, no. 4, pp. 587-597, 2010.
- [3] Anwar W., Wang X., and Wang L., "A Survey of Automatic Urdu Language Processing," in *Proceedings of the 5th International Conference on Machine Learning and Cybernetics*, Dalian, China, pp. 4489-4494, 2006.
- [4] ASCII Codes Table-Format of Standard Characters, available at: <http://ascii.cl/>, last visited 2011.
- [5] Ayub K., Khan A., and Khan H., "U: A Computer Programming Language in Urdu," in *Proceedings of International Conference on Contemporary Issues in IT in OIC Member Countries*, Islamabad, Pakistan, pp. 1-8, 2005.
- [6] Aziz Z., Aziz F., and Rashid K., "Converting Standard Html Input Controls to Urdu," *Information Technology Journal*, vol. 3, no. 1, pp. 69-73, 2004.
- [7] Becker D. and Riaz K., "A Study in Urdu Corpus Construction," in *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*, USA, vol. 12, pp. 1-5, 2002.
- [8] CRULP, Urdu Phonetic Keyboard, NUCES, Lahore, available at: [http://www.crulp.org/Downloads/Phonetic Keyboard.pdf](http://www.crulp.org/Downloads/Phonetic%20Keyboard.pdf), last visited 2012.
- [9] Font Creator Program Tutorials, available at: <http://www.high-logic.com/fontcreator/tutorials.html>, last visited 2011.
- [10] Ganjavi S., Georgiou P., and Narayanan S., "ASCII Based Transcription Systems for Languages with the Arabic Script: The Case of Persian," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 595-600, 2003.
- [11] Hussain S., Sarfraz H., and Dilawari A., "Assessing Urdu Language Support on the Multilingual Web," in *Proceedings of Conference on Human Language Technology for Development*, Alexandria, Egypt, pp. 11-20, 2011.
- [12] Hussain S. and Afzal M., "Urdu Computing Standard: Urdu Zabta Takhti (UZT) 1.01," in *Proceedings of IEEE International Multi-Topic Conference*, Lahore, Pakistan, pp. 223-228, 2002.
- [13] Javed S. and Hussain S., "Improving Nastalique Specific Pre-Recognition Process for Urdu OCR," in *Proceedings of IEEE 13th International Multitopic Conference, INMIC*, Islamabad, Pakistan, pp. 1-6, 2009.
- [14] Jeffrey L., Lonnie D., and Thomas I., *Systems Analysis & Design Methods*, Times Mirror/Mosby College Publishing, St. Louis, MO, French, 1986.
- [15] List of Urdu Software, available at: <http://meltingpot.fortunecity.com/mlk/470/urdupage.htm>, last visited 2012.
- [16] MSDN Library, available at: <http://msdn.microsoft.com/en-us/library/default.aspx>, last visited 2012.
- [17] Myers B., "A Brief History of Human Computer Interaction Technology," *ACM Interactions*, vol. 5, no. 2, pp. 44-54, 1998.
- [18] Rehman B. and Qureshi T., "Urdu as Interface Design Language- A Novel Approach," in *Proceedings of the 1st Saudi International Electronics, Communications and Photonics Conference*, Riyadh, KSA, p 1-6, 2011.
- [19] Riaz K., "Concept Search in Urdu," in *Proceedings of the 2nd PhD Workshop on Information and Knowledge Management*, New York, USA, pp. 33-40, 2008.
- [20] Salman S., Halim Z., Baig R., and Bashir S., "Web Content Mining: A Solution to Consumer's Product Hunt," in *Proceedings of International Conference on Computer, Electrical, and Systems Science, and Engineering*, Cairo, Egypt, pp. 307-311, 2008.
- [21] Urdu Words List, available at: http://www.crulp.org/software/ling_resources/wordlist.htm, last visited 2012.
- [22] West J., "How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies," *Research Policy*, vol. 32, no. 7, pp. 1259-1285, 2003.
- [23] Zobia R. and Waqas A., "A Hybrid Approach for Urdu Sentence Boundary Disambiguation," *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 250-255, 2012.



Bacha Rehman has received his MS degree in Computer System Engineering from Ghulam Ishaq Khan Institute (GIKI) of Engineering Sciences and Technology, Topi, KPK, Pakistan. His area of research is to investigate novel techniques to develop interface designs for Arabic scripted languages. He also developed a speech recognition model based on Discrete Wavelet Transformation (DWT) and Artificial Neural Networks (ANN). He is currently working in Forex industry.



intelligence.

Zahid Halim is currently working as an assistant professor at GIK Institute. He has received his PhD in computer science from FAST-NUCES in 2010. His areas of interest include data mining, machine learning and computational



Scholar in 1967 - 1971, Computer Science Department, Victoria University at Manchester, United Kingdom.

Mustaq Ahmad has vast experience of research and teaching. He is currently teaching in the Faculty of Computer Sciences and Engineering at GIKI. He received his PhD in computer software and computer applications as Colombo Plan