Ref: Proceedings of INMIC2001, Organised by IEEE & Lahore University of Management Sciences, Lahore, December 28-30, 2001, pp: 216-222

Urdu Computing Standards: Development of Urdu Zabta Takhti (UZT) 1.01

L2/02-003

Muhammad Afzal* and Sarmad Hussain**

Introduction

The benefits from Information Technology (IT) revolution cannot be reaped unless the masses use it, which is not possible unless computing is possible in a language that is understood by masses. This realization has come to many nations already. According to Jones S et al. (1992), the German *Munchener Oberlandesgericht* court decision of 1985 restricts the delivery of computers if it does not accompany operating instructions in German. Similar measures have been taken in many of the European and Far-Eastern countries, to enforce their local/national languages. Urdu software development dates back to late 1970s and early 1980s. Different applications have been developed by individuals and vendors since then, desktop publishing leading the scene for Urdu Software[1]. As all these packages were developed without any underlying computing standard, each has its own character set[2] and code page[3]. Therefore data exchange between them is not possible. They even have their own keyboard settings, therefore making it difficult for a typist to switch from one application to another.

With increasing usage of computers in Pakistan, and emerging Urdu applications from word processing to mega-scale projects such as National ID card project and proposed e-governance projects by Government of Pakistan, there is a growing need for consistent way of storing and exchanging data among applications in Urdu. However, consistent data storage and communication is not possible unless computing standards are defined. Otherwise, despite the deployment of individual systems, they would not be able to communicate among each other and therefore would be largely ineffective.

This paper narrates the details of the recent efforts put in the evolution of one such standard, the code page for Urdu: Urdu Zabta Takhti (UZT) 1.01. The Government of Pakistan (GoP) approved this standard in July 2000 (News 2000).

Evolution of Standard Code Page

All involved in Urdu software realized the need for a standard Urdu code page[4] and made individual efforts but with no gainful results. The two main reasons were that the vendors did not respond favorably due to competition and the responsible government departments did not realize its severity.

In Oct 1997, at 4th National Computer Conference, Islamabad, organised by Pakistan Computer Bureau, the problems faced by Urdu software Industry were discussed and need for standards was emphasized (Afzal 1999). Here standards for code page, keyboard layout, and orthography were identified. This time the response was over-whelming. To materialize the formulation of these standards, FAST-Institute of Computer Sciences, Lahore, organized the first day-long seminar on "The Standardization of Urdu keyboard Layout and Internal Character Representation" on September 12, 1998. Based on personal initiative of the authors, this seminar was projected through personal contacts. Over 35 experts from computer science discipline, Urdu linguists, software developers, Urdu software vendor and academics attended the seminar. Seminar focused on the issues of code page and keyboard standardization, which resulted in the formation of four separate committees focusing on Urdu code page, Urdu keyboard Layout,

* Professor, Fauji Foundation Institute of Management and Computer Sciences (FFIMCS), New Lalazar, Rawalpindi, Pakistan – 46000, email: afzal@ffimcs.edu.pk

** Head, Center for Research in Urdu Language Processing, FAST National University of Computer and Emerging Science. Email: sarmad.hussain@nu.edu.pk

[1] 'InPage' using ligature based Nastalique (www.concept-software.com) (Aziz 1987), 'Raakim' using character based Nastalique (www.systemsltd.com), 'PagePro' (www.softnetsystems.com), 'Global Plus' (www.inaamalvi.ca), 'Himala' (himala@pobox.com) and 'Urdu'98' (www.pakdata.com) are few such applications.

[2] Character set contains all characters, numerals, punctuation marks, symbols etc. required to write a language.

[3] Code page contains the mapping of the character set onto binary code, as represented for storage and communication of data, e.g. ASCII for English

[4] Code pages for other languages already exist, e.g. Microsoft standard for US English (1252), Greek (1253), Turkish (1254), Hebrew (1255), Arabic (1256).

Urdu Email and Internet. As the last three standards depended on the standard Urdu code page, so it was prioritized over others. The Urdu code page committee was further divided into two sub-committees, based in Islamabad and Lahore. The following sections discuss the interim proposals from these two committees, which resulted in the evolution of UZT 1.01. UZT 1.01 standard is discussed in detail by Hussain and Afzal (2001).

Urdu Alphabets and Standard Character Set

Before embarking on standardization of code page, i.e. to assign binary codes to characters, it was necessary to finalize Urdu character set. Different authors have quoted different number of characters in Urdu alphabet (e.g. even the elementary books for children do not agree on the same alphabet. Kifayat (1993), Siraj (1999), PTBB (2000), BUQ (1999) and KUQ (1999) have 36, 51, 53, 47 and 37 characters respectively; also see Hussain and Afzal (2001) for further discussion). In addition, various vendors have included a diverse set of punctuation marks and other symbols in their applications. As no general agreement was available, the committees agreed to consider the alphabet used by National Language Authority (NLA), which contains 57 characters shown in Figure 1 below.

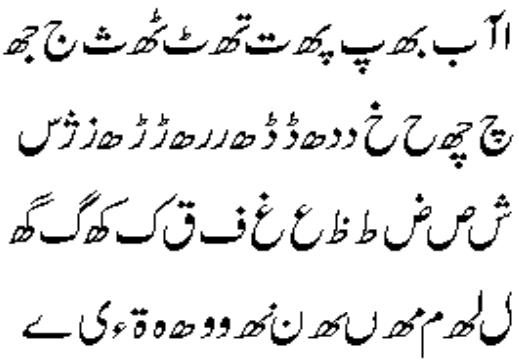

Figure 1: NLA Recommendation for Urdu Alphabet Urdu Collating Sequence

Once the number of characters was recommended, the next task was to determine their sorting or collating sequence to facilitate sorting and searching, which are required for language processing, databases, dictionary writing and other applications. Again, it was decided to follow guidelines of NLA for this purpose. The sequence of these characters is also given in Figure 1 above.

In addition to characters, Urdu also has diacritics (or aerab) that effect the sorting sequence. According to NLA guidelines, only three aerab, i.e. zabar, zer, pesh, effect the sorting order, and a character with the aerab comes after the same character without them, in the order of aerab listed. Therefore, at first level the sorting is performed on characters. Then, within any set of words that contain same initial character sequence, the aerab determine the sorting order.

In addition, these guidelines required the following five words to appear in the order listed (from right to left) in Figure 2. To keep this sorting order, especially for the last two words, distinction between two different types of spaces in necessary for Urdu.

Figure 2: Sorting Sequence of Some Urdu Words, as Specified by NLA

These issues about sorting and implementation using UZT 1.01 are discussed in detail by Hussain & Afzal (2001).

Code Page Version 0.1

Lahore sub-committee proposed this code page. This code page (shown in Figure A.1 in Appendix A) was based on seven bits, like ASCII, taking a total of 128 slots ($=2^7$). It contained the common punctuation, arithmetic and other symbols used in Urdu, in addition to the characters and aerab. For efficiency and to establish the correct sorting order, it was proposed that there should be two file formats - one without *aerab* for general word-processing like newspapers, and other with *aerab* where sorting is significant. In the latter file-format each character would be followed by an *aerab*, or null aerab, so it would have twice the storage requirements. However, to cater to aerab which have to be included in the normal (without aerab) writing, e.g. those shown in Figure 3 below, six extra characters were inserted (annotated by an encircled asterisk in Figure A.1 in Appendix A). These are included as characters for the file format without *aerab* and would be stored as character and respective *aerab* when stored in double-byte or with-a*erab* format.

Figure 3: Example Words with Aerab which are Normally Written

Code Page Version 0.2

Code page version 0.1 needed explicit toggle from one code page to the other, especially to type English text. It was discussed that if Urdu related character set was moved to the upper 128 (128-255) slots, lower half of the code page could

have the ASCII character set. The most significant bit would identify the language (0 referring to Engilsh and 1 to Urdu). To refine sorting, a null *Aerab* was introduced at slot 210 to act as place-holder for no a*erab*. As the character-*Aerab* occur in pairs, the characters without *Aerab* would be followed by this null to maintain the sequence. This code page is shown in Figure A.2 in Appendix A.

Commonly used characters, e.g., '@' symbol used in email addresses, were in ASCII part of the code page. Therefore, these were not repeated and the available space (eg 226-228 and 248-254) was reserved for future extension of language or symbols. Notably, Urdu space was included at slot 160 to enforce ligature[5] divides, whereas the normal space at slot 32 will act as delimiter for words.

Code Page Version 0.3

To avoid two level sorting and double byte storage a novel code page was proposed by Islamabad sub-committee. Each variant of a group, i.e. basic character, and its three variants, with *zabr, zer* and *pesh,* was allocated a separate slot. This required almost all 256 spaces and therefore was an 8-bit solution, as shown in Figure A.3. In addition, it required explicit toggle to switch to another language.

Applications requiring excessive sorting can exploit this encoding at application level by using the two least significant bits, as 00 represents no aerab, 01 with *zabr,* 10 with *zer* and 11 with *pesh*. For example, characters *ghain, ghain-zabr, ghain-zer* and *ghain-pesh* have binary codes 1011 1100, 1011 1101, 1011 1110 and 1011 1111 respectively.

However, there was a lot of redundancy in the code page. Also, for this code page addition of an extra character would require 4 places, whereas addition of an *aerab* (like *zabr, zer* and *pesh*) that could alter sorting would require an additional 41 more slots. Future extension of language would not be possible, because all the 256 spaces were already filled. In addition, this storage scheme was difficult to explain on linguistic grounds, because *bay* is a consonant character, while *bay-zabar* is a consonant-vowel sequence. Thus, the mapping was both on single and dual characters.

Code Page Version 0.4

Storage of *aerab* along with each character doubled the storage requirements. However, in common usage, *aerab* are rarely used in Urdu text. Enforcing aerab would therefore waste a lot of storage space and cause typing inefficiency. Thus, version 0.4 of code page reverted back to single byte storage system. This resulted in more space to for symbols or ligatures that are in common use. Some more symbols were also added in this version as shown in Figure A.4. It contained free slots 137, 157-159, 165-175, 186-191 and 233-254, which could be used for extension.

Code Page Version 0.5

In Pakistan, we need multilingual applications that can support English, Urdu and other regional languages including Balochi, Punjabi, Pushto and Sindhi languages. Thus while formulating standards the committees also considered this aspect. Code page version 0.5 was proposed, containing super-set of characters from Urdu and regional languages of Pakistan. This code page is shown in Figure A.5. In this case, the implementing software application would have to control which characters would be available to the application user.

Code Page Version 0.6

In a joint meeting of Lahore and Islamabad chapters on March 13, 1999, at Pakistan Computer Bureau, the code page version 0.5 was considered. It was agreed that as each language has its own identity, character set and collating sequence, so each should be considered independently. Further, as the scope of this committee was to formulate a standard code page for Urdu, so consideration of other languages was out of its scope. Thus, all effort was re-focused on Urdu code page.

It was further discussed that there are some machines in the current heterogeneous Internet environment, which still use 7-bit ASCII character set. Therefore, when a file with an 8-bit (or 1-byte) encoding is encountered, its most significant bit is truncated. So to avoid any unexpected results from such machines, we must avoid the use of slots 128-159, 127 and 255.

Code page version 0.6 contained all Urdu characters, symbols and commonly used ligatures. It also had reserved areas for language extension and vendors specific use. Version 0.6 specification also used toggle code 254 followed by language digit: '0' for Urdu or '1' for English (other languages to be added later). Its image is shown in Figure A.6.

Code Page Version 0.7

On May 12, 1999, in a meeting at National Language Authority, the code page version 0.6 and version 0.3 were once again compared. For concrete and objective comparison, first Terms Of Reference (TOR) were agreed upon (original transcripts shown in Appendix - B). TOR comprised of two sections, User Requirements and Technical Requirements. The two contesting code pages were compared for each of the requirements. And it was finalized a final effort be made to enhance, tune and finalize the code page based on version 0.6. The new code page had

[5] Ligature is the single connected series of characters forming a word or sub-word in Urdu writing.

exclusive blocks for each class of characters, symbols, *aerab* or user needed special symbols.

Code page version 0.7 was finalized. National Language Authority volunteered to get this code page approved by the GoP. However, it was decided that before finalization, this draft code page will be publicly circulated for feedback from Urdu software developers, vendors, Urdu linguists, computer scientists, academia and general public having some interest in the standardization activity.

Figure A.7 gives the image of the finally agreed code page version 0.7. This was circulated to public through media and was mailed to all concerned with the activity. All feedback was collated by NLA and forwarded to these committees. In final joint sessions, all the feedback was collated and considered. Much thought had been put into the formation of the final code page, therefore, there were only minor changes to it. These minor changes were incorporated and the code page was finalized. It was decided to be called Urdu Zabta Takhti (UZT). As version 0.7 was UZT 1.0, the final form submitted to GoP for approval was UZT 1.01. This code page was approved by GoP in July 2000 (News 2000). UZT 1.01 is discussed in detail by Hussain and Afzal (2001).

Conclusion

A lot of voluntary hard work and debate went into the evolution of UZT 1.01. Experts from all the concerned areas, including computer science and linguistics, were involved in this process, which lasted for two years. A lot of variations were experimented, testing both computational (storage and performance) and linguistic dimensions. What was eventually agreed was the compromise which provided the best computationally efficient solution within the bounds of the linguistic constraints. The deliberations therefore resulted in a robust, concise but flexible standard. However, this effort has not come to an end. UZT 1.01 standard may mature with time, and other standards (e.g. keyboard, Unicode (Zia 2000)) still need to be developed.

References

Afzal M, 1999; "Urdu Software Industry: Prospects, Problems and Need for Standards." In *Science Vision* vol 5(2), Islamabad.

Ashraf Nadeem, 2000, *Jadeed Urdu Lughat (talaba key liyey)*, Muqtadara Qaumi Zubaan, Islamabad.

Hussain S and Afzal M 2001; "Urdu Computing Standards: Urdu Zabta Takhti 1.01", *this volume.*

Jones S et al, 1992, A Digital Guide - *Developing international User Information*, International Edition, Digital Press, USA.

Kifayat 1993; Kifayat Rangeen Qaida, Silver Jubilee Edition, Kifayat Academy and Chambers, Urdu Bazaar, Karachi, May 1993.

KUQ 1999; Kids Urdu Qaida, Javed Publishers, Al-Faisal Market, Urdu Bazaar, Lahore.

Microsoft MS-DOS 5, 1991, User's Guide and Reference, Microsoft Corporation USA.

Mufti T 1988,"Research & Devlpt. in Urduization of Computers" in *Datalog* November 1988.

News 2000; "CE approves Urdu Code Plate for Computers", *The News Islamabad, Daily Newspaper, dated August 01, 2000*, p 12.

NLA 2000; "Chief Executive Approves Standard Urdu Code Page", *Ikhbar-e-Urdu, Islamabad, Monthly magazine of National Language Authority, August 2000*, pp 2.

PTBB 2001; Maira Qaida, 3rd Edition, Punjab Text Book Board Lahore.

BUQ 1999; Baby Urdu Qaida Rangeen, Moby Plaza, Haider Road, Saddar, Rawalpindi.

Siraj D Z 1999; Phool Aur Kaliyaan – Urdu Qaida, Feroze Sons Lahore (written according to integrated syllabus by National organization for syllabus & Text books, Federal Education Ministry, Government of Pakistan).

Unicode 1991; The Unicode Standard – Worldwide Character Encoding, version 1.0, volume 1, Addison Wesley Publishing Co, Reading

Unicode 2000; The Unicode Standards, version 3.0, Addison Wesley Publishing Co, Reading.

Appendix A: Earlier Versions of Urdu Code Page A.1.