Keystroke Saving in a Language with Highly Transparent Orthography

Francesco Curatelli¹ and Chiara Martinengo²

¹Department of Biophysical and Electronic Engineering (DIBE), University of Genova, Italy
²Department of Mathematics (DIMA), University of Genova, Italy

This paper proposes a pseudo-syllabic soft keyboard for the Croatian language. The orthogonal keyboard layout makes it possible to improve typing efficiency in terms of keystroke saving, and is based on a highly ordered arrangement of pseudo-syllabic keys. The positions of the consonant and vowel graphemes that constitute a pseudo-syllable are used to access it orthogonally and independently of each other. This allows the user to input a pseudo-syllable with a lower cognitive load than with non-orthogonal 2-D layouts. Moreover, due to the almost perfect transparency of the language, a pseudo-syllable to be input can be accessed fast and with a reduced cognitive load starting from its phonetic sounds. The results of the present study show that the obtainable keystroke savings are comparable with those scored by word prediction tools with one suggestion, i.e., those requiring only a moderate cognitive load by the user.

Keywords: human-computer interaction, soft keyboard, transparent orthography, orthogonal keyboard, keystroke saving

1. Introduction

One of the major research fields in Human-Computer Interaction (HCI) concerns the study and implementation of new text entry methods, which allow users to input more words per minute (WPM) with hardware or on-screen soft keyboards [13].

Indeed, standard hardware keyboards and touchscreen soft keyboards (where typing is done with a stylus or a finger) allow WPM values exceeding 40 and 30, respectively [7]. However, only expert users can achieve these speeds, and anyway typing speeds greatly decrease when mice or pointing devices (such as tracking ball, graphics tablet or eye-tracking) are used [14]. Moreover, non-expert and motor-impaired users typically obtain significantly lower values; only rarely, Alternative and Augmentative Communication (AAC) users are able to input (letter-by-letter) more than 10-15 WPM, whereas in many cases, because of the motor impairment, typing speeds can be as low as 1 WPM or even less [20].

Although the use of optimised hardware and soft keyboards increase typing speed [11, 12], only expert users fully exploit this possibility, being able to access keys without the need of visual searching the keys on the keyboard layout. Instead, all other users typically need a significant time to visually scan the keyboard to access a key. This time is suitably modelled by the Hick-Hyman law: \( T_v = a' + b' \log_2(n) \), where \( n \) is the number of the keys, \( a' \) is the response time and \( 1/b' \) is the bandwidth [18].

So, for non-expert and impaired users the best solution is to improve input speed by reducing the number of keys the user has to select to input the text. This reduction can be measured as the mean number of key selections needed to input a character \( KSPC = \#\text{Keys}/\#\text{Chars} \) [13]. This means that the user is able to input \#Chars characters by selecting \#Keys < \#Chars keys, so obtaining a high keystroke saving rate (or simply: saving rate). \( KSR = 100 \times \left( \#\text{Chars} - \#\text{Keys} \right)/\#\text{Chars} \). Obviously, the standard letter-by-letter input methods are characterised by \( KSPC = 1 \) and \( KSR = 0 \).

To this aim, a common approach is to use text prediction software tools, which provide the user a window with one or more suggestions for completing the current or, sometimes, the next word [4, 16, 15, 22]. In some cases, the
suggestions are not provided within a window, but displayed as moving texts on the screen area, so that the user selects the desired texts by forcing them to pass through a target zone [23]. With prediction tools savings up to 60% can be reached, but only when 5 or more suggestions are displayed [21, 19]. Instead, savings up to 37.1% are obtained for English with only one suggestion [19], and similar values (35%) are obtained with inflected languages, such as Dutch, French, German and Swedish [15]. This limitation cannot be ignored because, in any predictive tool, the higher the number of the proposed suggestions, the higher is the cognitive load required by the users, who have to visually scan the suggestion list to find the right word (if any). This is the reason why the high KSR values obtainable with prediction tools do not yield comparably high communication rates [20]. In fact, although the adoption of good word prediction systems can be often advantageous, this is not always the case. For example, some studies on the use of query suggestions in mobile phones have shown that using suggestions nearly halved the number of pressed key to enter a query, but the corresponding measured time was not reduced at all because of the cognitive load required by the user to read and evaluate the list of suggestions [10].

Another way to increase KSR is to implement on-screen keyboards where the keys do not hold single letters, but syllables or sub-syllables of the target language. This solution can be highly advantageous because a clever choice of the syllabic entities that are present in the keyboard makes it possible to score high KSR values, which is the main target of any new text entry paradigm. On the other hand, as just shown for word prediction, keystroke saving is fully exploited only when a moderate cognitive load is required by the user for searching and assembling the sequence of keys in the keyboard layout.

This has been the rationale for the introduction of a novel orthogonal keyboard paradigm, which is based on the definition of pseudo-syllables (p-syllables) as basic text entry units [2]. A target p-syllable is accessed by composing, orthogonally and independently of each other, the positions of its consonant and vowel graphemes (column and row, respectively), without the need to visually scan the whole keyboard or to memorise it. According to the Hick-Hyman law, this implies a reduction of the number of the items that have to be inspected or memorised and, consequently, of the required cognitive load and search time.

This obviously holds for a whichever alphabetical language, but especially for languages with transparent orthographies, i.e., with highly regular correspondence between orthography and phonetics. In fact, the presence of an almost one-to-one correspondence between phonemes and graphemes, which is typical of transparent languages, exempts the user from the additional cognitive load that is required to perform the complex sound to text translation needed in non-transparent languages such as English. Moreover, the presence of a direct and intuitive phoneme to grapheme association makes it possible for an efficient use of the orthogonal keyboard, not only by normal users (expert and non-expert), but also, and mainly, by motor-impaired users and by users who are affected by difficulty in reading and writing (i.e., dyslexia and dysgraphia).

The orthogonal keyboard paradigm has been suitably applied to Italian and Spanish [2], which are two languages with transparent orthographies, and, more recently, also to a highly non-transparent language such as English [3]. In this last case, similar keystroke savings have been obtained with a clever choice of the p-syllable set, which makes it possible to directly input many monosyllabic English words [3].

However, since Italian and Spanish are far from being really perfect transparent languages, from the beginning of our research in this field we have been intrigued by the possibility to apply the orthogonal paradigm to languages which do as much as possible approximate perfect transparency. In particular, we were fascinated by the possible application to Slavic languages such as Croatian and Serbian, which have two almost perfect transparent orthographies, derived from a clever choice of the two alphabets used, i.e., the modified Gaj’s Latin alphabet [5] for Croatian and the Srpska Ćirilica for Serbian. Moreover, as the Croatian alphabet is also one of the two scripts used for the Bosnian and Serbian languages, and a reduced version is used for the Slovenian language, Croatian has been chosen as target language for the study which is described in the present paper.
To this aim, the characteristics of Croatian have been analysed and an orthogonal keyboard layout has been designed according to the frequency statistics of the language. The suitability of the proposed layout has been then tested with results that are comparable with those obtained with Italian and Spanish keyboards.

In Section 2, the main characteristics of the orthogonal keyboard paradigm are outlined, while Section 3 outlines the statistical analysis of the language and the design of the Croatian orthogonal keyboard layout. Section 4 outlines the experimental results and conclusions.

2. The Orthogonal Pseudo-syllabic Model

The orthogonal text entry paradigm is based on the definition of a keyboard layout which makes it possible to input dominant and subordinate p-syllables. The dominant p-syllables are directly selected without using any shift key, while the subordinate ones need the selection of specific shift keys. Both types are accessible without the need to perform a complete visual exploration of the orthogonal keyboard, because the text units are assigned to the keyboard keys so that their horizontal and vertical positions are easily inferable from the graphemic structure of the related p-syllabic entity. In fact, each column of the keyboard is assigned to a specific consonant grapheme \( \hat{C}_i \), which can be either a single consonant or a sequence of consonants denoting (in a transparent language) a single phoneme. Instead, each row is assigned to a specific vowel grapheme \( \hat{V}_i \), which can be either a single vowel or semi-vowel or a sequence of them. In this way, the p-syllable that contains both the vowel grapheme of row \( x \) and the consonant grapheme of column \( y \) can be accessed by the direct composition of the consonant and vowel positions, which are independent of each other (orthogonal). Anyway, only one action is needed to input dominant or subordinate p-syllables (provided that the related shift key is concurrently selected). All other syllables and pseudo-syllables can be selected with a sequence of dominant or subordinate p-syllables, so requiring two or (sometimes) more actions.

Given a language \( L \), we start by defining the complete consonant and vowel sets \( \hat{C}_S_L \) and \( \hat{V}_W_L \), which contain the most used consonant and vowel graphemes \( \hat{C}_i \) and \( \hat{V}_i \), respectively (\( n_{\hat{C}_S_L} = |\hat{C}_S_L| \) and \( n_{\hat{V}_W_L} = |\hat{V}_W_L| \) are the set cardinalities). The two sets also contain the empty grapheme \( \epsilon \), so that the \( x,y \) grapheme composition can build purely vowel and consonant p-syllables too. So, the complete p-syllable set \( \hat{P}_S_L \) (with cardinality \( n_{\hat{P}_S_L} = |\hat{P}_S_L| \) contains any possible string \( \hat{C}_i \hat{V}_i \) and \( \hat{V}_i \hat{C}_i \). The second ones are subordinate p-syllables and are selectable by means of the forward/reverse shift (FR-Sh).

As already stated, this scheme allows users to access the target p-syllable without the need of visually scanning the whole keyboard or memorizing it. This implies a reduction of the required cognitive load and search time, due to the smaller number of the items to be considered. Therefore, also non-expert users can obtain better access times than with non-ordered p-syllable layouts. It is worth noting that this approach is completely different from the typical approach found in literature, which is based on the selection of several single-letter keys to build syllabic entities [9].

In many cases, the direct use of the complete p-syllable set would yield a too large keyboard layout; when this is the case, the size of the orthogonal array must be reduced by applying one or more optimisation steps to generate the final orthogonal keyboard layout. First, it can be verified if it is convenient to eliminate columns or rows whose p-syllables have low marginal frequencies in the target language (column deletion (CD) and row deletion (RD)). Second, it can be verified if it is convenient to fold columns containing p-syllables with low marginal frequencies, by assigning them to other columns (column folding (CF)), and merge rows containing p-syllables with low marginal frequencies, by assigning them to other rows (row folding (RF)). The selection of a subordinate key belonging to one of the folded entities is done by selecting one of two specific shifts (CF-Sh for CF optimisation or RF-Sh for RF optimisation), which are unique for all the folded columns and folded rows. Another possible optimisation concerns merging together clusters of two or more columns if each column contains a subset of rows with rarely used p-syllables, and the subset belongs to rows different from those of
all the other columns of the same cluster (Column merging (CM)).

When CD and RD optimisations are applied, there is a reduction in the consonant and vowel graphemes, thus two optimised sets for consonant and vowel graphemes, $\widehat{CS}_L$ and $\widehat{WV}_L$, and, consequently, an optimised p-syllable set $PS^o_L$ are obtained.

Is it worth noting that in defining the orthogonal keyboard layout only the optimisation steps should be applied which are strictly needed to obtain a suitable keyboard size for the target language. As it will be shown in the next Section, Croatian orthography does not require an extensive application of the optimisation steps. In fact, the definition of the orthogonal keyboard will require only the CF optimisation.

3. The Croatian Orthogonal Keyboard

As already stated, we have chosen Croatian because of its almost perfect transparent orthography. In fact, the native Croatian alphabet is composed of 30 graphemes, with a one-to-one correspondence with the phonemes of the language: $a, b, c, Ć, Ć, d, dž, d, e, f, g, h, i, j, k, l, lj, m, n, nj, o, p, r, s, š, t, u, v, x, z, ŵ; i.e.$:

a) 22 one-letter graphemes of the English alphabet (in fact, q, w, x and y do not belong to the Croatian alphabet);

b) 5 specific one-letter consonants with diacritics ($č, Ć, ď, š$ and $ž$);

c) 3 specific two-letter consonants (digraphs) which denote specific sounds and are treated in all effects as separate letters ($dž$, $lj$, and $nj$) (they correspond exactly to one-letter graphemes of the Serbian Cyrillic alphabet).

This alphabet is also called Gaj’s Latin alphabet, after the Croatian linguist Ljudevit Gaj who in 1830 set the Croatian alphabet starting from the Latin alphabet [5]; the only difference from the actual script consisted in the use of a further digraph $dj$ in place of the actual letter $ď$. It is worth noting that the Croatian alphabet is also one of the two scripts used for the Bosnian and Serbian languages; moreover, a reduced version is used for the Slovenian language.

Concerning the missing Latin letters ($q, w, x$ and $y$), although they are used only to write foreign words, their increasing presence in modern texts (such as newspaper articles, textbooks, and so on) has suggested not to discard them from the orthogonal layout. Instead, a notable characteristic of the Croatian language is that doubled consonants are very rarely used; for this reason, they have not been included in the orthogonal layout. Finally, the double grapheme $qu$ has been used in place of $q$ because in almost all the words (both Croatian and foreign) the simple consonant $q$ is followed by $u$. Therefore, the starting (complete) consonant and vowel sets have been defined as follows:

$$\widehat{CS}_{hr} = \{ e, b, c, Ć, Ć, d, dž, ď, e, f, g, h, j, k, l, lj, m, n, nj, o, p, r, s, š, t, u, v, x, y, z, ŵ \}$$

$$\widehat{WV}_{hr} = \{ e, a, e, i, o, u \}$$

Because $|\widehat{CS}_{hr}| = 30$ and $|\widehat{WV}_{hr}| = 6$, the resulting keyboard would be too wide for an efficient use. The reduction of the keyboard width has been done by identifying a suitable set of less frequently used p-syllables and then applying a folding optimisation step which sets them as subordinate p-syllables of the orthogonal layout. To this aim, we have generated a detailed statistic of the p-syllable frequencies starting from the data present in the Croatian Language Repository - newspapers sub-corpus [8], which gives the relative frequencies of 710,177 different words in texts taken from Croatian newspapers [1]. In particular, we have taken into account the first 120,000 most frequent words, whose relative frequencies constitute more than 97% of the total word occurrences.

Because the dominant p-syllables have the structures $Č$, $Ŵ$, or $ČV$, we have extracted the frequency statistics concerning the 34 single graphemes $G = Č$ or $Ŵ$, and the 145 double graphemes $ČV$, by computing, for each p-syllable, the sum of the weighted relative frequencies of the words in which the p-syllable is present. In other words, the relative frequency of each word has been taken $n$ times, where $n$ is the number of occurrences of the p-syllable in that word; in this way, a more realistic statistics of real frequencies of the graphemes has been obtained. The statistics concerning the single and double grapheme p-syllables are outlined in Table 1 and Table 2, respectively.
Keystroke Saving in a Language with Highly Transparent Orthography

Table 1. Single grapheme % statistics.

<table>
<thead>
<tr>
<th>G</th>
<th>%G</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>59.64</td>
</tr>
<tr>
<td>i</td>
<td>50.84</td>
</tr>
<tr>
<td>o</td>
<td>46.90</td>
</tr>
<tr>
<td>e</td>
<td>43.90</td>
</tr>
<tr>
<td>n</td>
<td>29.69</td>
</tr>
<tr>
<td>r</td>
<td>26.98</td>
</tr>
<tr>
<td>t</td>
<td>24.57</td>
</tr>
<tr>
<td>s</td>
<td>22.65</td>
</tr>
</tbody>
</table>

The CF optimisation defines the final orthogonal layouts for the dominant and the subordinate p-syllables that require the selection shift $CF-Sh = \text{on}$. As the dominant p-syllables are selectable directly in the layout, they have been chosen according to the frequencies outlined in Table 1 and Table 2, which show very low frequencies for p-syllables involving the consonants y, w, dž, x and q. In fact, the relative frequencies in words of these consonants are strictly less than 0.15%, whereas all other consonants have values greater than 1%. This consideration has suggested to select as consonants for the dominant p-syllables the 24 consonants with frequencies exceeding 1%, i.e., all but the 5 consonants above. In this way, the size of the orthogonal keyboard is $(n_y \times n_x) = (25 \times 6)$, and the 25 columns of the orthogonal keyboard are assigned to the dominant consonants, lexicographically ordered one after the other (the empty grapheme $\epsilon$ is put at the beginning, so corresponding to the column of the vowels): $\epsilon, b, c, \tilde{c}, d, d\tilde{z}\tilde{z}, f, g, h, j, k, l, lj, m, n, nj, p, r, s, \tilde{s}, t, v, z, z\tilde{z}$.

All other p-syllables, containing the consonants y, w, dž, x and qu, are set as subordinate p-syllables (whose selection requires that $CF-Sh = \text{on}$) and are allocated to 5 columns.

To render the cognitive task of the user as easy as possible, these 5 consonants have been lexicographically ordered one after the other; moreover dž and qu have been allocated to columns whose dominant consonants are lexicographically close to their original position in the unfolded configuration:

Table 2. ĈV % statistics.
$d\tilde{z} \Rightarrow \text{col } 6 \ (d), \ qu \Rightarrow \text{col } 18 \ (p),$
\[ \w \Rightarrow \text{col } 23 \ (v), \ x \Rightarrow \text{col } 24 \ (z), \]
y \Rightarrow \text{col } 25 \ (\tilde{z}).

This CF mapping yields a subordinate layout in which only 5 columns are assigned; this means that 19 columns are potentially available for the allocation of $19 \times 6 = 114$ additional strings (added words). As our aim was to explore the potentiality of the orthogonal paradigm, this additional improvement has not been addressed in the present work. However, further gains in terms of keystroke saving could be easily obtained by allocating, as added words, a suitable set of syllables and words chosen among the most frequent words of the language, by using the data provided, for example, by the Croatian Language Repository [1].

The final keyboard configurations are outlined in Tables 3 to 5. Table 3 and 4 outline the layouts of the Croatian orthogonal keyboard for the dominant p-syllables and the reverse subordinate p-syllables with FR-Sh = on, while Table 5 outlines the layout for the subordinate p-syllables with CF-Sh = on.

It is worth noting that the use of dominant and subordinate p-syllables with FR-Sh = on makes it possible to insert the graphemic sequences VCCV and CVVC with only two actions. This is very important because in the Croatian language, as well as in almost all Slavic languages, the first graphemic structure is very frequent.

### Table 3. Croatian orthogonal keyboard for dominant p-syllables.

|     | b | c | Ć | Ć | d | f | g | h | j | k | l | lj | m | n | nj | p | r | s | š | t | v | z | ţ |
| a   | ba | ca | ča | da | dř | fa | ga | ha | j | ka | la | lja | ma | na | nj | a | ra | ra | sa | ša | ta | va | za | ža |
| e   | be | ce | če | de | dře | fe | ge | he | je | ke | le | lje | me | ne | nje | pe | re | re | se | še | te | ve | ze | že |
| i   | bi | ci | či | di | dři | di | fi | gi | hi | ji | ki | li | lji | mi | ni | nji | pi | ri | si | ši | ti | vi | zi | ži |
| o   | bo | co | čo | do | dřo | fo | go | ho | jo | ko | lo | ljo | mo | no | njo | po | re | ro | so | šo | to | vo | zo | žo |
| u   | bu | cu | ču | du | dřu | fu | gu | hu | ju | ku | lu | lju | mu | nu | nju | pu | ru | su | šu | tu | tu | vu | vz | žu |

### Table 4. Croatian orthogonal keyboard for subordinate p-syllables (FR-Sh = on).

|     | b | c | Ć | Ć | dř | f | g | h | j | k | l | lj | m | n | nj | p | r | s | š | t | v | z | ţ |
| a   | ab | ac | ač | ad | dřaf | ag | ah | aj | ak | al | aj | am | an | anj | ap | ar | as | aš | at | av | az | až |
| e   | eb | ec | eč | ed | dřed | ef | eg | eh | ej | ek | el | elj | em | en | enj | ep | er | es | eš | et | ev | ez | ež |
| i   | ib | ic | ič | id | dřid | if | ig | ih | ij | ik | il | ilj | im | in | inj | ip | ir | is | iš | it | iv | iz | iž |
| o   | ob | ic | ič | id | dřid | if | ig | ih | ij | ik | il | ilj | im | in | inj | ip | ir | is | iš | it | iv | iz | iž |
| u   | ub | uc | uč | ud | učdř | uf | ug | uh | uj | uk | ul | ulj | um | un | unj | up | ur | us | uš | ut | uv | uz | už |

### Table 5. Croatian orthogonal keyboard for subordinate p-syllables (CF-Sh = on).

|     | b | c | Ć | Ć | dř | f | g | h | j | k | l | lj | m | n | nj | qu | r | s | š | t | w | x | y |
| a   | ba | ca | ča | dža | dřa | fa | ga | ha | ja | ka | la | lja | ma | na | nj | a | qua | ra | sa | ša | ta | wa | xa | ya |
| e   | be | ce | če | dže | dře | fe | ge | he | je | ke | le | lje | me | ne | nje | que | re | se | še | te | ve | ze | že |
| i   | bi | ci | či | dži | dři | fi | gi | hi | ji | ki | li | lji | mi | ni | nji | qui | ri | si | ši | ti | wi | xi | yi |
| o   | bo | čo | čo | džo | do | fo | go | ho | jo | ko | lo | ljo | mo | no | njo | quo | ro | so | šo | to | wo | xo | yo |
| u   | bu | ču | ču | džu | du | fu | gu | hu | ju | ku | lu | lju | mu | nu | nju | quu | ru | su | šu | tu | wu | xu | yu |
4. Experimental Results and Conclusions

The Croatian orthogonal keyboard has been evaluated by using the following parameters. \( NW, N_\alpha, N_O, N_{Sp} \) are the numbers of words, alphanumeric, non-alphanumeric, and space characters in the text; \( N_C = N_\alpha + N_O \) is the total number of characters in the text; \( K_\alpha \) is the number of alphanumeric keys typed on the keyboard; \( S_{Sh} \) and \( S_K \) are the numbers of capital letters and \( FR-Sh \) shifts.

The typing gains are measured by 3 types of KPSC (keystrokes per character) and KSR (keystroke saving rate) parameters.

\[
KSPC_\alpha = \frac{K_\alpha}{N_\alpha} \text{is the mean number of the alphanumeric keys to input an alphanumeric character, and}
\]

\[
KSPC_\beta = \frac{(K_\alpha + N_O - N_{Sp})}{(N_C - N_{Sp})} \text{is the mean number of the non-space keys to input a non-space character, and}
\]

\[
KSPC_\gamma = \frac{(K_\alpha + N_O)}{N_C} \text{is the mean number of keys to input a whichever character, and}
\]

\[
KSR_\alpha = 100 \times (1 - KSPC_\alpha) \text{is the percentage of the alphanumeric keys saved by using the orthogonal keyboard.}
\]

\[
KSR_\beta = 100 \times (1 - KSPC_\beta) \text{is the percentage of the non-space keys saved by using the orthogonal keyboard.}
\]

\[
KSR_\gamma = 100 \times (1 - KSPC_\gamma) \text{is the percentage of the keys (of any type) saved by using the orthogonal keyboard.}
\]

Finally, \( KSPC_\gamma = \frac{(K_\alpha + N_O)}{N_C} \text{is the mean number of keys to input a whichever character, and}
\]

\[
KSR_\gamma = 100 \times (1 - KSPC_\gamma) \text{is the percentage of the keys (of any type) saved by using the orthogonal keyboard.}
\]

To test the keyboard layout, some Croatian texts have been automatically analysed to find how much improvement can be obtained with respect to the classical, letter-by-letter approach. The texts were downloaded from the electronic versions of the popular Croatian magazine Globus [6].

The obtained results (outlined in Table 6), show significant keystroke savings (greater than 40.5%) for alphanumeric characters, and only slightly lower values (greater than 38.8%) for non-spaces characters. This means that the additional use of clever algorithms for the automatic addition of spaces could easily improve the already good (greater than 32.2%) keystroke savings obtained for all characters. These results are impressive, especially if they are compared with the only slightly better keystroke savings obtained with Italian and Spanish [2]. In fact, these two Neolatin languages are structurally characterised by very high frequencies of CV sequences, whereas in Croatian (and in almost all Slavic languages) triple and quadruple consonants sequences occur very frequently. Probably, these high gains can be ascribed to the fact that in Croatian most of the more frequent words are monosyllabic words with CV structure. In fact, according to the used Croatian Corpus, the ten most frequent words are, in order: je, u, i, se, da, na, za, su, a, od [1].

The obtained keystroke savings are comparable with those obtained, for English and highly inflected languages with one-suggestion word prediction tools [19, 15], which, as stated in Section 1, require only a moderate cognitive load. This is also the case with the orthogonal keyboard, which is a text entry method that requires a lower cognitive load than non-orthogonal 2-D

<table>
<thead>
<tr>
<th>Text</th>
<th>( NW )</th>
<th>( N_C )</th>
<th>( N_\alpha )</th>
<th>( N_O )</th>
<th>( N_{Sp} )</th>
<th>( S_{Sh} )</th>
<th>( S_K )</th>
<th>( K_\alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>globus-1</td>
<td>1341</td>
<td>8357</td>
<td>6724</td>
<td>1633</td>
<td>1328</td>
<td>152</td>
<td>278</td>
<td>3995</td>
</tr>
<tr>
<td>globus-2</td>
<td>1223</td>
<td>7098</td>
<td>1514</td>
<td>5584</td>
<td>1197</td>
<td>169</td>
<td>134</td>
<td>3258</td>
</tr>
<tr>
<td>globus-3</td>
<td>1181</td>
<td>6701</td>
<td>5237</td>
<td>1464</td>
<td>1163</td>
<td>158</td>
<td>159</td>
<td>3077</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Text</th>
<th>( KSPC_\alpha )</th>
<th>( KSR_\alpha )</th>
<th>( KSPC_\beta )</th>
<th>( KSR_\beta )</th>
<th>( KSPC_\gamma )</th>
<th>( KSR_\gamma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>globus-1</td>
<td>0.5941</td>
<td>40.58</td>
<td>0.6117</td>
<td>38.82</td>
<td>0.6734</td>
<td>32.65</td>
</tr>
<tr>
<td>globus-2</td>
<td>0.5834</td>
<td>41.65</td>
<td>0.6058</td>
<td>39.41</td>
<td>0.6723</td>
<td>32.76</td>
</tr>
<tr>
<td>globus-3</td>
<td>0.5875</td>
<td>41.24</td>
<td>0.6099</td>
<td>39.00</td>
<td>0.6776</td>
<td>32.23</td>
</tr>
</tbody>
</table>

(b)

Table 6. KSR Statistics for the Croatian orthogonal keyboard.
Besides the reduction of the cognitive load due to the characteristics of the orthogonal scheme, the question may be raised if the same can be said for an additional reduction due to the transparency level of the target language. Certainly, in a perfect transparent language such as Croatian, as well as almost always in languages such as Italian and Spanish, there is a one-to-one correspondence between phonemes and graphemes. For this reason, the translation task is very simple, and the related cognitive load is low, whereas in English the same task is much more complex, thus requiring a higher cognitive load. But, one may wonder if this reduction could be accurately measured as a function of language transparency, i.e., by means of some direct or indirect objective measure [17]. At first sight, a possible indirect and objective measure could be the typing speed, since there is an intuitive correlation between WPM and cognitive load. However, there are also other important factors, such as differences in orthography, grammatical rules and inflection of the languages involved, which do not allow us to tell if this correlation can be accurately modelled or not. Thus, the answer to this interesting issue would require specific and detailed experiments, which were not among the aims of the present work.

In conclusion, the present study has shown that the orthogonal paradigm is well suited to obtain high keystroke savings for a highly transparent Slavic language such as Croatian. This is a promising result, also because we can argue that this result can be automatically extended to Serbian, due to its almost perfect alphabetic correspondence and language similarity with Croatian. Thinking along the same lines, it would be useful to explore the application of the orthogonal paradigm to Cyrillic alphabets, because of their use in Serbian, Bulgarian, Macedonian, Russian and many other Eastern languages. Another interesting target language is Polish, which is not only a highly transparent language, but is also the most widespread Slavic language written with the Latin alphabet.

References


Keystroke Saving in a Language with Highly Transparent Orthography


Received: October, 2009
Revised: February, 2010
Accepted: March, 2010

Contact addresses:
Francesco Curatelli
DIBE–University of Genova
Via Opera Pia 11/A
16145 Genova, Italy
e-mail: curatelli@unige.it

Chiara Martinengo
DIMA–University of Genova
Via Dodecaneso 35
16146 Genova, Italy

FRANCESCO CURATELLI is a professor of electronics at the University of Genova – Department of Biophysical and Electronic Engineering (DIBE), Genova, Italy. His research activities include the study and development of CAD tools, techniques and algorithms for hardware/software real-time systems, educational methodologies for teaching mathematics to impaired children, human-computer interaction and assistive technology.

CHIARA MARTINENGO is a researcher in mathematics at the University of Genova – Department of Mathematics (DIMA), Genova, Italy. Her research activities include the study of several arguments in commutative algebra and algebraic geometry, educational methodologies for teaching mathematics to impaired children, human-computer interaction and assistive technology.