



**HAL**  
open science

## DESSK: DEscription Space for Soft Keyboards

Mathieu Raynal, Georges Badr, I. Scott MacKenzie

► **To cite this version:**

Mathieu Raynal, Georges Badr, I. Scott MacKenzie. DESSK: DEscription Space for Soft Keyboards. International Conference on Human-Computer Interaction (HCI 2022), Jun 2022, Gothenburg, Sweden. pp.109-125, <10.1007/978-3-031-05409-9\_9>. <hal-03736479>

**HAL Id: hal-03736479**

**<https://hal.science/hal-03736479v1>**

Submitted on 22 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# DESSK: description space for soft keyboards

Mathieu Raynal<sup>1</sup>, Georges Badr<sup>2</sup>, and I. Scott MacKenzie<sup>3</sup>

<sup>1</sup> ELIPSE team, IRIT lab, University of Toulouse, France  
`mathieu.raynal@irit.fr`

<sup>2</sup> TICKET LAB, Antonine University, Baabda, Lebanon  
`georges.badr@ua.edu.lb`

<sup>3</sup> Electrical Engineering and Computer Science, York University, Toronto, Canada  
`mack@cse.yorku.ca`

**Abstract.** We present DESSK, a description space for soft keyboards. DESSK provides a framework for any soft keyboard to be graphically described along four dimensions – context, representation, interaction, and linguistic system – each with multiple criteria. DESSK begins with a blank working template which is populated with labels and symbols to characterize the dynamics and interactions of the keyboard. The goal is to provide a framework within which any soft keyboard can be placed and to serve as a theoretical basis to situate soft keyboards in relation to proposed or existing systems.

**Keywords:** soft keyboards, description space, interaction models, descriptive model

## 1 Introduction

Early soft keyboards were identical representations of physical keyboards: The character layout was similar and each soft button corresponded to a physical key. These keyboards were originally created to allow people with motor disabilities to enter text. The keyboards were "soft" or "virtual", meaning they were rendered in graphics on a display. Interaction proceeded either through single-switch scanning [9] or using eye gaze [12].

Soft keyboards are also standard on virtually all desktop systems. Interaction can use any pointing device, such as a mouse, trackball, joystick, or even an eye tracking system. With the emergence of pen-based computing, smartphones, and tablets, these text entry systems are more common as they replace physical keyboards. And new interaction devices, such as smartwatches, joysticks, or virtual reality headsets, also support text entry. Moreover, because these systems are created in software, they offer more possibilities for interaction than a physical keyboard. Thus, they are able to evolve dynamically, adjusting to the user's input, and they often interact with other software components or use several interaction modalities.

These possibilities result in a wide variety of text entry systems using soft or virtual keyboards. Although contexts of use are diverse, the goal is the same:

supporting and improving text entry speed and accuracy. Nevertheless, there is no theoretical framework to situate this work in relation to proposed or existing systems. Several reviews and workshops reflect on text input [44, 50, 23, 18, 27]; however, given the diversity of systems, interaction contexts, or interaction modalities, these reviews focus on one or a few specific criteria – e.g., accessibility according to a motor impairment [45] or visual impairment [55, 46], mobile use [24], or 3D virtual environments [10] – and compare the systems according to this prism.

The purpose of this paper is to describe soft keyboards generally, whatever the intended context and constraints. We present different types of soft keyboards according to the elements that characterize them, including their context of use, the interaction modalities, and the software components coupled to the keyboard. For this, we present a Description Space for soft keyboards, called DESSK, which encompasses the criteria to describe any type of soft keyboard.

The goal of DESSK is to describe each soft keyboard from a system point of view: that is, to represent the way the user interacts with the keyboard, to describe the internal functioning of the keyboard, and how it evolves as the user types. Moreover, beyond functional aspects, we contextualize soft keyboards by including the intended environment.

DESSK only concerns soft keyboards. We define soft keyboards as any text entry system composed of several interactive zones to produce text strings. On a soft keyboard, the interactive zones appear on a display and are generally materialized by buttons or edges to which one or several characters (or codes) are linked. DESSK does not take into account gesture recognition or handwriting recognition systems or voice dictation systems used for text entry.

## 2 DESSK: A descriptive model

In the space of modeling, DESSK is an example of a *descriptive model* [20]. Descriptive models are tools for thinking. They breakdown a problem space into constituent parts and offer a visual depiction of the problem space. With this, they empower researchers to understand and think in different ways about the problem space and, importantly, to develop a deeper understanding of current phenomenon and to inspire new possibilities. This is in contrast to *predictive models* or *analytic models*, which are tools for predicting or quantifying [19].

Descriptive models are everywhere. Often, researchers describe the components of a problem space without framing their efforts as a model. That’s what we do! The process is natural and unconscious, and usually just serves to present current practice as prelude to a new idea for empirical study. However, it is also the case that ”describing the components of a problem space” is a contribution in itself, if done in a thorough, constructive, and illustrative manner. And the HCI literature is replete with descriptive models presented in this way. Examples include Johansen’s quadrant model for groupware [15], Buxton’s three-state model for graphical input [6], MacKenzie and Castellucci’s frame model for visual attention [21], or Card et al.’s model human processor [7].

The problem space of DESSK is soft keyboards. What are the interaction and contextual parts of soft keyboards? Can these be organized in a comprehensive framework that encompasses all soft keyboards? Is there an appropriate visual organization for this framework? These are the sort of questions we considered in developing our description space for soft keyboards, or DESSK for short. The components in DESSK are presented in the following sections.

### 3 Description criteria

Our description space is defined along four dimensions. These are now described with reference to the blank working template shown in Figure 1. The template will be populated with additional details later using specific examples of soft keyboards.

Context	User	
	Display	
	Device	
Representation	Visibility	
	Language	
	Cardinality	
	Layout	
Interaction	Navigation	
	Validation	
Linguistic system		

Fig. 1: Working template for description space of soft keyboards (DESSK)

Briefly, the first dimension, **Context**, concerns the context of use of the soft keyboard; that is, characteristics of the user and environment for which the system has been designed. The second dimension, **Representation**, describes the visual appearance of the soft keyboard – the way the keyboard is structured and its graphic representation. The third dimension, **Interaction**, focuses on how the user interacts with the system, both to navigate on the keyboard, and to validate desired zones. Finally, the last dimension, **Linguistic system**, defines the language components used to facilitate the input with soft keyboards. Each of these dimensions is broken into criteria presented as follows.

### 3.1 Context of use

Soft keyboards were initially designed for people who cannot use a physical keyboard to enter text on their computer. However, since the emergence of smartphones and more generally touch screens, the soft keyboard has become an essential tool for most users. The design of a soft keyboard generally responds to a specific problem, often related either to the user's abilities or to the characteristics of the device on which the keyboard is used. Context of use therefore includes at least three criteria: User, Display, and Device.

**User** This indicates whether the keyboard is designed for a particular profile of people. This can include the abilities of the user (due to a disability or the context of interaction), but also the expertise of the user (novice vs. expert).

For users' abilities, we distinguish soft keyboards intended for people without major constraints, those designed for motor-disabled people [36], and those studied for people with a visual impairment [4, 43].

On the other hand, user performance evolves when using a soft keyboard [25]. Users are initially "novices", but with time learn the new layout and gradually become "experts". Some keyboards have been designed more specifically to help users get started with this new keyboard [26]. Conversely, other keyboards provide additional interactions or shortcuts to increase the text entry speed when users gain expertise with the keyboard [8, 13].

**Display** The first text input systems were mainly used on a traditional desktop computer where the soft keyboard was displayed on the screen. But the uses have greatly diversified over the last twenty years with the democratization of new display surfaces such as smartphones, smartwatches, interactive TVs, head-mounted displays, etc. The diversity of these display surfaces shows the importance of this criterion when designing a text entry system. For example, the layout may differ if displayed on a watch screen compared to a mixed-reality headset.

For the different possible values of this criterion, we have chosen to distinguish between soft keyboards displayed on traditional screens coupled to a computer, and those presented on touch screens. Usually the keyboards displayed on a computer screen are those used by motor-challenged people who use the keyboard to enter text on the computer.

With touch screens, we group together those on devices such as smartphones, tablets, or even interactive tables that are used with the finger or a stylus. On the other hand, we distinguish the very small touch screens used, for example, by smartwatches because the very small display surface brings about a specific problems of keyboard display and interaction.

**Device** Beyond the display surface, the other important criterion when designing the system is the interaction modality available to the user to interact with the soft keyboard. An interaction modality, as defined by Nigay and Coutaz

[32], combines the physical device with the interaction language. This criterion determines which physical device (mouse, joystick, Wii Remote, touch screen, etc.) is used to interact with the soft keyboard. As we will see in the "interaction" dimension, input generally includes subtasks: navigation and validation. The interaction language used can be different for these two subtasks. This is why the interaction language is described in the "navigation" and "validation" criteria of the "interaction" dimension.

Many devices have been used to interact with a soft keyboard. The most widely used are certainly the touch screens that can be found in many situations: interactive terminals, interactive tables, or personal devices such as smartphones, tablets, or smartwatches. These screens have the particularity of being both the input device and the display surface. We distinguish two types of touch screens: those of smartwatches which are very small, and others with a larger footprint.

Some soft keyboards are also designed for entertainment systems. In this context, devices are varied: gamepad [14], joystick [52], or remote controller (like Wii [16]).

Finally, as noted in the introduction, soft keyboards are also the main way to interact and communicate for people with a motor impairment. The interaction device is then adapted to the motor skills of the person. People who still have little motor skills use pointing devices such as a joystick [53], trackball [51], while the most paralyzed use specific devices such as an eye tracker [49], muscle contractions [11], vocal input (non verbal) like humming [35] or hissing [34].

In some scenarios, physical keys are used to move a cursor on the soft keyboard [5, 39]. For soft keyboards with single-switch scanning, the cursor moves automatically from zone to zone and, when the cursor is on the right zone, the user validates it through some input mechanism [48].

Finally, we could also take physical keyboards into account in this criterion. Some of them use unconventional layouts or interactions (such as OrbiTouch (by Keybowl, Inc.) or DataHand [17]).

**Summary** Table 1 summarizes the possible values for the three criteria of the "Context of use" dimension. It should be noted that the set of possible values is not restricted to those given, but is malleable as per new technologies and applications that arise.

Table 1: Possible values for each criteria of the "Context of use" dimension

Criteria	Possible values
User	able-bodied, motor impairment, visual impairment, novice, expert
Display	all, screen, projection, touch screen (smartphones, tablets), small touch screen (smartwatches), interactive TV, head-mounted display
Device	mouse, joystick, touch screen (smartphones, tablets), small touch screen (smartwatches), remote control (Wii), eye tracker, muscle contractions, vocal input (non verbal), hissing, tongue

### 3.2 Representation

The second dimension of our description space concerns the structure and the graphical representation of a soft keyboard. We use four criteria to describe the representation.

A soft keyboard can sometimes be decomposed into several text input systems. For example, a soft keyboard that includes word prediction or work completion is a combination of two input systems: the keyboard on the one hand and the word list on the other hand. Values are assigned to following criteria separately for each part of the keyboard.

**Visibility** Some parts of the keyboard are not permanently visible. For example, the POBox system [29] displays a list of the most probable words but only after the first character of the word is entered. We therefore distinguish between those parts that are permanently visible and those that are visible only occasionally.

**Language** The language determines the information that appears in the interactive zones. This can be characters usable on the system, undefined words (for predicted words lists), or codes to represent or extend the character set. For example, EdgeWrite [54] uses codes representing the four corners of a square. All characters have a representation using a sequence of these codes.

**Cardinality** Cardinality corresponds to the number of interactive zones on the input system. We write "N" when there are as many interactive areas as there are characters. If there are less zones than elements, the number of zones is given. This can be the case of a soft keyboard using a code (such as EdgeWrite [54] or H4-Writer [22]), or when several characters are associated with the same zone. Examples of the latter are ambiguous keyboards, such as a phone keypad, where multiple letters are associated with each key.

**Layout** Here we describe how the different interactive zones are arranged in relation to each other. This can be any grid such as the Qwerty layout, a horizontal or vertical list, a circular layout [28], or a square layout [33].

**Summary** Table 2 summarizes the possible values for the four criteria of the "Representation" dimension. As with the Context-of-use dimension, the set of possible values may vary as per new technologies and applications that arise.

### 3.3 Interaction

Whatever the type of elements produced (codes, characters, or words), the production of these elements occurs in two stages: first, navigating to the desired interactive zone and then validating or confirming the zone.

Table 2: Possible values for each criteria of the "Representation" dimension

Criteria	Possible values
Visibility	none, occasionally, permanent
Language	characters, words, codes
Cardinality	full, number
Layout	none, grid, list, square, circle

**Navigation** The navigation phase is only present for systems using a pointer or cursor to select the interactive zones. For soft keyboards on smartphones, connected watches, or other devices using a touch screen, there is no pointer: The user directly accesses the zone, for example, using a finger or stylus. In this case, there is no navigation phase. If a pointer or cursor is present, its movement depends on the user's ability to manipulate a pointing device or its equivalence. For users with a motor impairment, navigation uses a cursor that is moved, typically without using a pointing device. One possibility is the use of non-verbal voice input: the user vocalizes sounds which map to virtual arrow and select keys [45].

More commonly, the cursor or hot spot is moved automatically from zone to zone on a soft "scanning keyboard" (aka "single-switch scanning"). When the cursor is on the right zone, the user validates it through some input mechanism. There are two types of movement: *continuous movement* of a pointer using a pointing device, and *discrete movement* of a cursor from zone to zone using directional keys or other discrete actions to realize and control the direction of movement. We therefore distinguish three types of navigation: automatic, direct, or indirect. Each type can be continuous or discrete.

**Validation** Finally, the validation of an interactive zone produces the element linked to this zone. For soft keyboards on smartphones or other touch screens, the most common validation consists in directly tapping the desired zone. Pressing a finger on the screen, the user validates the zone under the finger. During an interaction by a gesture stroke, the interactive zones are generally validated by crossing: Each zone crossed by the trace is considered validated. In this case, a language model makes it possible to determine the desired word from the sequence of zone crossings [56]. It is also possible to perform a gesture on the desired zone to validate it. This technique of validation by gesture is sometimes used for zones where there are several characters. The use of gestures to validate the zone allows, in the same action, removing the ambiguity on the desired character [31].

For soft keyboards used with a pointing device, validation is usually done with a button on the pointing device. Similarly, for keyboards where the selection of the zone is done by single-switch scanning, the user validates the selection using a switch or any other input mechanism that produces a discrete action to indicate validation of the selection [48].

**Summary** Table 3 summarizes the possible values for the three criteria of the "Interaction" dimension. As before, the set of possible values is not restricted to those given, but can expand as new interaction techniques emerge.

Table 3: Possible values for each criteria of the "Interaction" dimension

Criteria	Possible values
Navigation	couple (automatic, direct, indirect) $\times$ (discrete, continuous)
Validation	discrete, continuous

### 3.4 Linguistic system

Finally, the last dimension of our description space determines whether our soft keyboard uses algorithms based on linguistic knowledge. The process combines actions previously performed with linguistic knowledge (rules, statistics, etc.) to complete or modify a string of characters already entered. Alternatively, the process may dynamically modify part of the keyboard to help the user in his text input [30].

These systems are of different types. The most well-known are word prediction algorithms that propose the most probable words according to the given prefix [3]. Prediction systems can also propose characters that are most likely to succeed the prefix already entered.

Soft keyboards can also use a deduction system, whereby information coming from the user's input is used to deduce the intended word. This information can include, for example, the ordered sequence of the zones previously hovered over or validated. From this information and linguistic knowledge, the system will then deduce the word best matching this information.

**Summary** Table 4 summarizes the possible values for the three criteria of the "Context of use" dimension. Again, the set of possible values may expand as per new technologies and applications that arise.

Table 4: Possible values for each criteria of the "Context of use" dimension

Criteria	Possible values
Linguistic system	characters prediction, words prediction, words deduction

## 4 Putting DESSK into practice

Our description space is presented in the form of a table with one criterion per line, as shown above in Figure 1. In populating the description space, the objective is to show the dynamics of the soft keyboard and possible interactions between the different parts of the keyboard. We use green circles to represent the possible productions, with C, W, or S displayed in these circles to infer that the system produces characters, words, or strings. Blue lines represent the interactions between the different criteria of the soft keyboard: The circle at one end indicates which criterion generates the interaction while the diamond at the other end indicates the criterion impacted by this interaction. Green and blue rectangles, respectively, define the information that is sent to the linguistic system, and the modification that is made on the soft keyboard in reaction to the information sent by the linguistic system.

The modifications can be of different types: The most common consist of modifying the set of words displayed in the list [3] or the set of characters displayed on additional keys [37], or altering the character positions on a "scanning keyboard" [48]. It is also possible to modify the key sizes [1] or shapes [2] to facilitate access to the most probable characters. Other modifications have also been tried, for example, changing the font to highlight the most probable characters to assist searching for novice users [26], or modifying the transfer function of the pointing device to improve navigating to keys [38].

Modifications to the keyboard can also occur depending on user interactions. For example, the FishEye keyboard is a full keyboard designed to be displayed on a smartphone or PDA and used with the finger or a stylus [41]. When the user (or stylus) touches the screen, the keys around this point magnify to make them easier to read (see Figure 2a).



(a) Implementation [42]

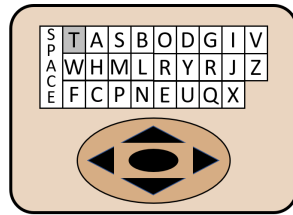
Context	User	able-bodied
	Display	touch screen
	Device	
Representation	Visibility	permanent
	Language	characters
	Cardinality	full
	Layout	grid <span style="float: right;">key size </span>
Interaction	Navigation	direct – continuous <span style="float: right;"></span>
	Validation	discrete <span style="float: right;"></span>
Linguistic system	none	

(b) DESSK description

Fig. 2: FishEye Keyboard

The user can then move his finger (or stylus) on the screen until hovering over the desired character. When he lifts his finger, the character is validated. Figure 2b is the description with DESSK of the FishEye Keyboard.

Another example is F.O.C.L., a soft keyboard designed in the late 90s to be implemented on a cell phone, pager, or other mobile device [5]. The user moves a cursor from key to key using four directional keys. After each character is entered, the character layout changes so that the most likely characters are as close as possible to the cursor. Figure 3 shows the description of this keyboard in our space. We can observe that the characters entered are added to the text produced (green circle with a C) and sent, in parallel, to the character prediction system which returns the list of characters ordered according to the probability of each one to be entered. With this information, the system updates the character layout.



(a) Implementation [5]

Context	User	all
	Display	screen
	Device	physical keys
Representation	Visibility	permanent
	Language	characters
	Cardinality	N
	Layout	grid <span style="float: right;">layout</span>
Interaction	Navigation	indirect – discrete <span style="float: right;">↻</span>
	Validation	discrete <span style="float: right;">C</span>
Linguistic system	characters prediction	<span style="float: right;">character</span>

(b) DESSK description

Fig. 3: FOCL system.

If the soft keyboard uses regions to enter elements, they will be presented in additional columns. See Figure 4. For example, a soft keyboard with a word list includes two regions: the soft keyboard and the list of predicted words. The border between the regions describes the relationship between these two parts. On the one hand, the two systems are used in parallel (dotted border) whereby the user switches between the keyboard and the prediction list during the input of a word. Or, the two systems are used separately and sequentially (solid border). For example, in the DUCK keyboard [40], the user enters an approximate set of characters for a word and then selects the correct word from a set of words proposed by the deduction system.

The example in Figure 4 is SIBYLLE [48] which is an assistive communication system for people with motor disabilities who cannot use a pointing device. Navigation on the soft keyboard is done by automatic switch scanning. The user validates the selection with a contact that is activated when the cursor is on the desired key. SIBYLLE is a complete text entry system which proposes

all the characters and functionalities proposed on a standard keyboard, but also complementary keys which propose additional functionalities. It includes regions for character input, a word list, a numeric keypad, and a menu bar. Two linguistic prediction systems are included, a character prediction system which rearranges characters after each character entered, and a word prediction system which proposes a list of the most probable words.

The representation of SIBYLLE in DESSK (Figure 4) presents only the part of SIBYLLE that evolves dynamically during the input; that is, the character input block whose layout is rearranged after each character input, and the word list (on the left) that is updated after each new character entered.

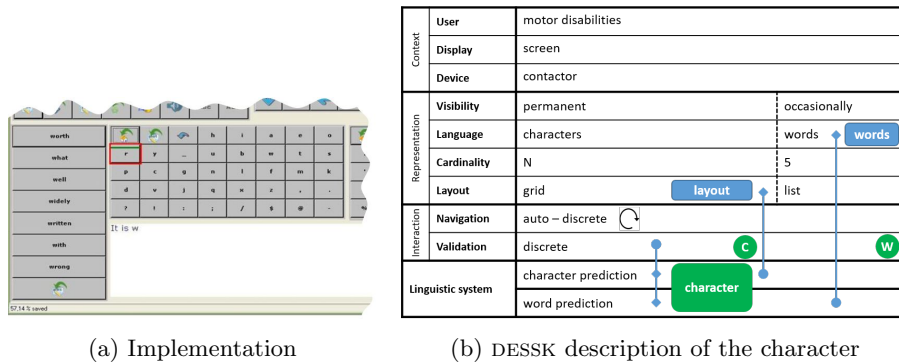


Fig. 4: SIBYLLE system.

For many soft keyboards, zone validation produces a character or code. In the case of code-based keyboards, a sequence of codes is produced before being sent to a deduction algorithm that determines the character corresponding to the sequence. This sequence is produced by validating several interactive zones. To show this in our description space, we use a black rounded arrow between the selection and validation criteria. The number of repetitions necessary to achieve the sequence is given as an interval near the arrows. For example, in EdgeWrite [54], each character is coded by a sequence of corners the user has passed through (see Figure 5a). The right part of Figure 5 shows the description of EdgeWrite in DESSK .

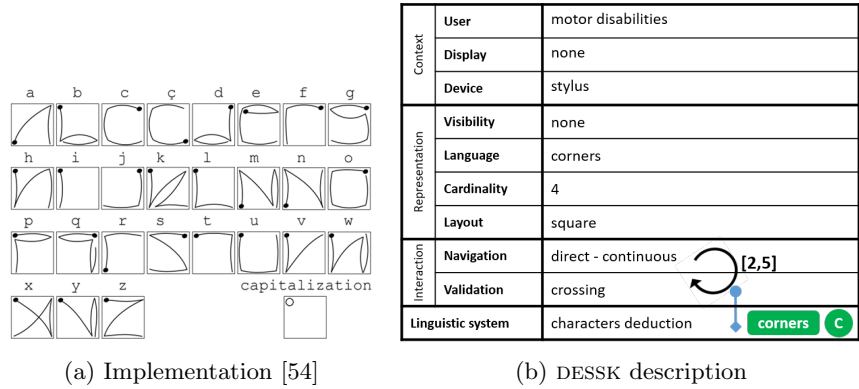


Fig. 5: EdgeWrite system.

Finally, the rounded arrow used with the validation criteria represents a repetition of the validation phase. This type of repetition is used in particular to remove an ambiguity when several characters are on the same interactive zone. For example, the selection of a character on a multi-tap soft keyboard is done by clicking once, twice, or three times on the button containing the desired character [47]. Figure 6 represents the operation of a multi-tap keyboard with our description space.

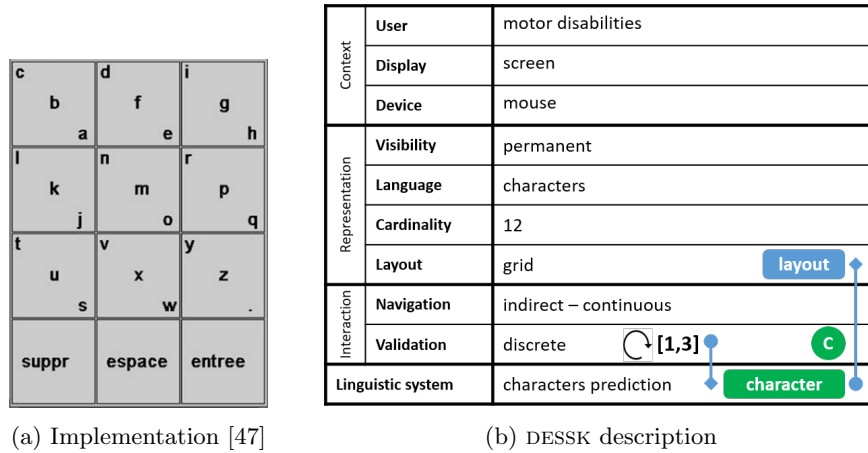


Fig. 6: Multi-tap keyboard.

## 5 Conclusion

We present DESSK, a description space for text entry with soft keyboards. The space is divided into four regions, Context, Representation, Interaction, and Linguistic system. The regions include sub-regions which are populated with labels and symbols to describe the operation of the keyboard, including dynamic and linguistics features and the context of use.

In order to serve the community working in text entry, we have designed and developed a website dedicated to text entry systems: <http://text-entry.com/>

This website is primarily a showcase of the various existing soft keyboard entry systems and aims to bring together as many research prototypes as possible. Beyond a simple catalog of existing solutions, the goal of our website is also to allow the exploration of this set of systems according to the desired characteristics. To do this, our website includes all the criteria from our description area to allow all solutions to be filtered according to these criteria. The description in our description space is available for each system presented on our page. All of these descriptions are available on our site <http://text-entry.com/>.

Our goal in proposing this descriptive model is to provoke thought about the problem space for the design of text input systems. We built our model from a large set of keyboards found in the literature. However, our descriptive model (certainly) has limits; some systems may be difficult to map into the DESSK model. Our description space will evolve over time depending on the feedback we get and the new systems we learn about.

## References

1. Al Faraj, K., Mojahid, M., Vigouroux, N.: Bigkey: A virtual keyboard for mobile devices. In: Proceedings of the International Conference on Human-Computer Interaction – HCI '09. pp. 3–10. Springer, Berlin (2009)
2. Aulagner, G., François, R., Martin, B., Michel, D., Raynal, M.: Floodkey: Increasing software keyboard keys by reducing needless ones without occultation. In: Proceedings of the 10th WSEAS international conference on Applied computer science. pp. 412–417. World Scientific and Engineering Academy and Society (WSEAS), ACM, New York (2010)
3. Badr, G., Raynal, M.: WordTree: Results of a word prediction system presented thanks to a tree. In: International Conference on Universal Access in Human-Computer Interaction (LNCS 5616). pp. 463–471. Springer, Berlin (2009)
4. Banubakode, S., Dhawale, C.: Survey of eye-free text entry techniques of touch screen mobile devices designed for visually impaired users. *Covenant Journal of Informatics and Communication Technology* **2**(1) (2013)
5. Bellman, T., MacKenzie, I.S.: A probabilistic character layout strategy for mobile text entry. In: Proceedings of Graphics Interface '98 – GI '98. pp. 168–176. Canadian Information Processing Society (CIPS), Toronto (1998)
6. Buxton, W.A.S.: A three-state model for graphic input. In: Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction – INTERACT'90. pp. 449–456. Elsevier, Amsterdam (1990)

7. Card, S.K., Moran, T.P., Newell, A.: The psychology of human-computer interaction. Erlbaum, Hillsdale, NJ (1983)
8. Chen, X., Grossman, T., Fitzmaurice, G.: Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In: Proceedings of the 27th annual ACM Symposium on User interface software and technology – UIST ’14. pp. 615–620. ACM, New York (2014)
9. Damper, R.: Text composition by the physically disabled: A rate prediction model for scanning input. *Applied Ergonomics* **15**(4), 289–296 (1984)
10. Dube, T.J., Arif, A.S.: Text entry in virtual reality: A comprehensive review of the literature. In: Proceedings of the 21st International Conference on Human-Computer Interaction – HCI ’19 (LNCS 11572). pp. 419–437. Springer, Berlin (2019)
11. Felzer, T., MacKenzie, I.S., Beckerle, P., Rinderknecht, S.: Qanti: A software tool for quick ambiguous non-standard text input. In: International Conference on Computers for Handicapped Persons – ICCHP ’10. pp. 128–135. Springer, Berlin (2010)
12. Frey, L.A., White, K., Hutchison, T.: Eye-gaze word processing. *IEEE Transactions on Systems, Man, and Cybernetics* **20**(4), 944–950 (1990)
13. Isokoski, P.: Performance of menu-augmented soft keyboards. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI ’04. pp. 423–430. ACM, New York (2004)
14. Isokoski, P., Raisamo, R.: Quikwriting as a multi-device text entry method. In: Proceedings of the Third Nordic conference on Human-Computer Interaction – NordiCHI ’04. pp. 105–108. ACM, New York (2004)
15. Johansen, R.: Groupware: Future directions and wild cards. *Journal of Organizational Computing and Electronic Commerce* **1**(2), 219–227 (1991)
16. Jones, E., Alexander, J., Andreou, A., Irani, P., Subramanian, S.: GesText: Accelerometer-based gestural text-entry systems. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems – CHI ’10. pp. 2173–2182. ACM, New York (2010)
17. Knight, L.W., Retter, D.: Datahand: design, potential performance, and improvements in the computer keyboard and mouse. In: Proceedings of the Human Factors Society Annual Meeting. vol. 33, pp. 450–454. SAGE Publications Sage CA: Los Angeles, CA (1989)
18. Kristensson, P.O., Brewster, S., Clawson, J., Dunlop, M., Findlater, L., Isokoski, P., Martin, B., Oulasvirta, A., Vertanen, K., Waller, A.: Grand challenges in text entry. In: Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems – CHI ’13, pp. 3315–3318. ACM, New York (2013)
19. MacKenzie, I.S.: Motor behaviour models for human computer interaction. In: Carroll, J.M. (ed.) *HCI models, theories, and frameworks: Toward a multidisciplinary science*, pp. 27–54. Morgan Kaufmann, San Francisco (2003)
20. MacKenzie, I.S.: *Human-computer interaction: An empirical research perspective*. Elsevier, Amsterdam (2013)
21. MacKenzie, I.S., Castellucci, S.J.: Eye on the message: Reducing attention demand for touch-based text entry. *International Journal of Virtual Worlds and Human-Computer Interaction* **1**, 1–9 (2013)
22. MacKenzie, I.S., Soukoreff, R.W., Helga, J.: 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-Writer. In: Proceedings of the 24th annual ACM Symposium on User Interface Software and Technology – UIST’11. pp. 471–480. ACM, New York (2011)

23. MacKenzie, I.S., Tanaka-Ishii, K.: Text entry using a small number of buttons. In: MacKenzie, I.S., Tanaka-Ishii, K. (eds.) *Text entry systems: Mobility, accessibility, universality*, pp. 105–121. Morgan Kaufmann, San Francisco (2007)
24. MacKenzie, I.S., Tanaka-Ishii, K.: *Text entry systems: Mobility, accessibility, universality*. Elsevier, Amsterdam (2010)
25. MacKenzie, I, S., Zhang, S.X.: The design and evaluation of a high-performance soft keyboard. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. pp. 25–31. ACM, New York (1999)
26. Magnien, L., Bouraoui, J.L., Vigouroux, N.: Mobile text input with soft keyboards: Optimization by means of visual clues. In: *International Conference on Mobile Human-Computer Interaction*. pp. 337–341. Springer, Berlin (2004)
27. Majaranta, P., Riih , K.J.: Twenty years of eye typing: Systems and design issues. In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications – ETRA’02*. pp. 15–22. ACM, New York (2002)
28. Mankoff, J., Abowd, G.D.: Cirrin: A word-level unistroke keyboard for pen input. In: *Proceedings of the ACM Symposium on User Interface Software and Technology – UIST’98*. pp. 213–214. ACM, New York (1998)
29. Masui, T.: POBox: An efficient text input method for handheld and ubiquitous computers. In: *International Symposium on Handheld and Ubiquitous Computing*. pp. 289–300. Springer, Berlin (1999)
30. Merlin, B., Raynal, M.: SpreadKey: Increasing software keyboard key by recycling needless ones. In: *10th European conference for the advancement of assistive technology in europe (AAATE 2009)*. pp. 138–143. IOS Press, Amsterdam (2009)
31. Nesbat, S.B.: A system for fast, full-text entry for small electronic devices. In: *Proceedings of the 5th International Conference on Multimodal Interfaces – ICMI’03*. pp. 4–11. ACM, New York (2003)
32. Nigay, L., Coutaz, J.: A design space for multimodal systems: concurrent processing and data fusion. In: *Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems*. pp. 172–178. ACM, New York (1993)
33. Perlin, K.: Quikwriting: Continuous stylus-based text entry. In: *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST’98*. pp. 215–216. ACM, New York (1998)
34. Pol cek, O., Mikovec, Z., Slav k, P.: Predictive scanning keyboard operated by hissing. In: *Proceedings of the 2nd IASTED International Conference Assistive Technologies*. pp. 862–869. ACTA Press, Calgary, Canada (2012)
35. Polacek, O., Mikovec, Z., Sporcka, A.J., Slavik, P.: Humsher: a predictive keyboard operated by humming. In: *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. pp. 75–82. ACM, New York (2011)
36. Polacek, O., Sporcka, A.J., Slavik, P.: Text input for motor-impaired people. *Universal Access in the Information Society* **16**(1), 51–72 (2017)
37. Raynal, M.: Keyglasses: Semi-transparent keys on soft keyboard. In: *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. pp. 347–349. ACM, New York (2014)
38. Raynal, M., MacKenzie, I.S., Merlin, B.: Semantic keyboard: Fast movements between keys of a soft keyboard. In: *International Conference on Computers Helping People with Special Needs – ICCHP ’14 (LNCS 8548)*. pp. 195–202. Springer, Berlin (2014)
39. Raynal, M., Martin, B.: Slidekey: Impact of in-depth previews for a predictive text entry method. In: *International Conference on Computers Helping People with Special Needs – ICCHP ’20*. pp. 363–370. Springer, Berlin (2020)

40. Raynal, M., Roussille, P.: DUCK: A DeDUCTive soft Keyboard for visually impaired users. In: *Harnessing the Power of Technology to Improve Lives*, pp. 902–909. IOS Press, Amsterdam (2017)
41. Raynal, M., Truillet, P.: Fisheye keyboard: Whole keyboard displayed on pda. In: *International Conference on Human-Computer Interaction*. pp. 452–459. Springer, Berlin (2007)
42. Raynal, M., Vinot, J.L., Truillet, P.: Fisheye keyboard: Whole keyboard displayed on small device. In: *UIST 2007, 20th annual ACM Symposium on User Interface Software and Technology – UIST '07*. pp. pp–65. ACM, New York (2007)
43. Siqueira, J., de Melo Soares, F.A.A., Ferreira, D.J., Silva, C.R.G., de Oliveira Berretta, L., Ferreira, C.B.R., Félix, I.M., da Silva Soares, A., da Costa, R.M., Luna, M.M.: Braille text entry on smartphones: A systematic review of the literature. In: *2016 IEEE 40th annual computer software and applications conference – COMPSAC'16*. vol. 2, pp. 521–526. IEEE, New York (2016)
44. Soukoreff, R.W., MacKenzie, I.S.: Recent developments in text-entry error rate measurement. In: *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems – CHI'04*. pp. 1425–1428. ACM, New York (2004)
45. Sporka, A.J., Felzer, T., Kurniawan, S.H., Poláček, O., Haiduk, P., MacKenzie, I.S.: Chanti: Predictive text entry using non-verbal vocal input. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems – CHI'11*. pp. 2463–2472. ACM, New York (2011)
46. Tinwala, H., MacKenzie, I.S.: Eyes-free text entry with error correction on touch-screen mobile devices. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction – NordiCHI'10*. p. 511–520. ACM, New York (2010)
47. Vigouroux, N., Vella, F., Truillet, P., Raynal, M.: Evaluation of aac for text input by two groups of subjects: Able-bodied subjects and disabled motor subjects. In: *8th ERCIM Workshop, User Interface for All, Vienne, Autriche*. pp. 28–29. Springer, Berlin (2004)
48. Wandmacher, T., Antoine, J.Y., Poirier, F., Départe, J.P.: Sibylle, an assistive communication system adapting to the context and its user. *ACM Transactions on Accessible Computing (TACCESS)* **1**(1), 1–30 (2008)
49. Ward, D.J., Blackwell, A.F., MacKay, D.J.: Dasher: A data entry interface using continuous gestures and language models. In: *Proceedings of the 13th annual ACM symposium on User interface software and technology – UIST'00*. pp. 129–137. ACM, New York (2000)
50. Wigdor, D., Balakrishnan, R.: A comparison of consecutive and concurrent input text entry techniques for mobile phones. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems – CHI'04*. pp. 81–88. ACM, New York (2004)
51. Wobbrock, J., Myers, B.: Trackball text entry for people with motor impairments. In: *Proceedings of the ACM SIGCHI conference on Human Factors in computing systems – CHI'06*. pp. 479–488. ACM, New York (2006)
52. Wobbrock, J.O., Myers, B.A., Aung, H.H.: Writing with a joystick: a comparison of date stamp, selection keyboard, and edgewrite. In: *Proceedings of Graphics Interface – GI'04*. pp. 1–8. CIPS, Toronto (2004)
53. Wobbrock, J.O., Myers, B.A., Aung, H.H., LoPresti, E.F.: Text entry from power wheelchairs: Edgewrite for joysticks and touchpads. In: *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility – ACCESS'03*. pp. 110–117. ACM, New York (2003)

54. Wobbrock, J.O., Myers, B.A., Kembel, J.A.: EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In: Proceedings of the 16th annual ACM Symposium on User Interface Software and Technology – UIST’03. pp. 61–70. ACM, New York (2003)
55. Ye, L., Sandnes, F.E., MacKenzie, I.S.: QB-Gest: Qwerty bimanual gestural input for eyes-free smartphone text input. In: Proceedings of the 22nd International Conference on Human-Computer Interaction – HCI’20 (LNCS 12188). pp. 223–242. Springer, Berlin (2020)
56. Zhai, S., Kristensson, P.O.: Shorthand writing on stylus keyboard. In: Proceedings of the ACM SIGCHI conference on Human factors in Computing Systems – CHI’03. pp. 97–104. ACM, New York (2003)