



**HAL**  
open science

## **STAACS3: Simulation Tool for AAC with Single-Switch Scanning**

Lisa Hoiry, Cherifa Ben Khelil, Frédéric Rayar, Anaïs Halftermeyer, Jean-Yves Antoine, Mathieu Raynal

### ► **To cite this version:**

Lisa Hoiry, Cherifa Ben Khelil, Frédéric Rayar, Anaïs Halftermeyer, Jean-Yves Antoine, et al.. STAACS3: Simulation Tool for AAC with Single-Switch Scanning. 17h International Conference of the Association for the Advancement of Assistive Technology in Europe (AAATE 2023), Aug 2023, Paris, France. pp.487 - 494, <10.3233/shti230666>. <hal-04209633>

**HAL Id: hal-04209633**

**<https://hal.science/hal-04209633v1>**

Submitted on 18 Sep 2023

**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

April 2022

# STAACS<sup>3</sup> : Simulation Tool for AAC with Single-Switch Scanning

Lisa HOIRY <sup>a,1</sup>, Cherifa BEN KHELIL <sup>b</sup>, Frédéric RAYAR <sup>b</sup>,  
Anaïs HALFTERMEYER <sup>c</sup>, Jean-Yves ANTOINE <sup>b</sup>, Mathieu RAYNAL <sup>a</sup>

<sup>a</sup>*IRIT - University of Toulouse, France*

<sup>b</sup>*LIFAT - University of Tours, France*

<sup>c</sup>*LIFO - University of Orléans, France*

**Abstract.** Communication for people with motor impairments is a difficult, yet necessary, task in daily activities. To do so, soft keyboards, digital counterpart of physical keyboards, are used. Depending on their features, several designs can be considered, however their thoughtful evaluation in real-life is not feasible. Therefore, it becomes necessary to filter the possible configurations wisely, taking advantage of simulation tools. This paper presents STAACS<sup>3</sup>, a Simulation Tool for AAC with Single-Switch scanning, that allows to model and simulate the performance of any type of scanning keyboard regardless of its keys configuration, scanning strategy or prediction systems.

**Keywords.** Augmentative and Alternative Communication, Soft keyboards, Single-switch scanning, Simulation

## 1. Introduction

Nowadays, written communication via email, social networks or discussion forums is one of the main uses of digital tools (PC, Smartphone, tablets, etc.). Digital text entry has become a main activity in our daily lives. In case of motor impairment, users can use a so-called "soft keyboard", i.e. a digital representation of the physical keyboard. These soft keyboards are used with a pointing device adapted to the user's motor skills. For people with very poor motor skills, it is also possible to replace the pointing device with an automatic scrolling system where the cursor moves from key to key at regular intervals, usually from left to right and from top to bottom on the keyboard. The user selects a character by pressing a single-switch when the cursor is on the desired key. A large diversity of keyboards configurations can be imagined answering user needs. In order to evaluate the proposed keyboards during the design phase, and before assessing their relevance and usability by end users during an evaluation phase, simulation tools are a relevant solution.

In this paper, we propose a Simulation Tool for AAC with Single-Switch Scanning (STAACS<sup>3</sup>) that allows to model and simulate the performance of any type of scanning keyboard whatever a) its keys configuration, b) the scanning strategy used, c) the pre-

---

<sup>1</sup>Corresponding Author: Lisa Hoiry, Lisa.Hoiry@irit.fr.

diction systems used. Moreover, the simulation can easily be adapted to any language by modifying the text corpora used for the prediction systems and for the simulation. Finally, in order to be able to simulate all types of keyboards, we have also modified the calculation of the scan step per character (SPC). This allows us to take into account systems that use two different scan interval.

## 2. AAC with single-switch scanning

As aforementioned, a large number of virtual keyboards are available to AAC users, depending of their configurations. These configurations mainly differ by the scanning mode that is used. The use of language prediction systems (word or character) also has a significant impact on the different keyboards offered. This section describes the features that can be found in current virtual keyboards.

### 2.1. Scanning mode

Virtual keyboards offer multiple scanning modes that can be used, such as linear, row/column, half-and-half and n-ary. Figure 1 illustrates these scanning modes.

The **linear** scanning mode can be considered as the most basic approach. In this mode, the cursor systematically scans each key on the keyboard, following a left-to-right and top-to-bottom progression (Figure 1a). The selection of the desired character re-

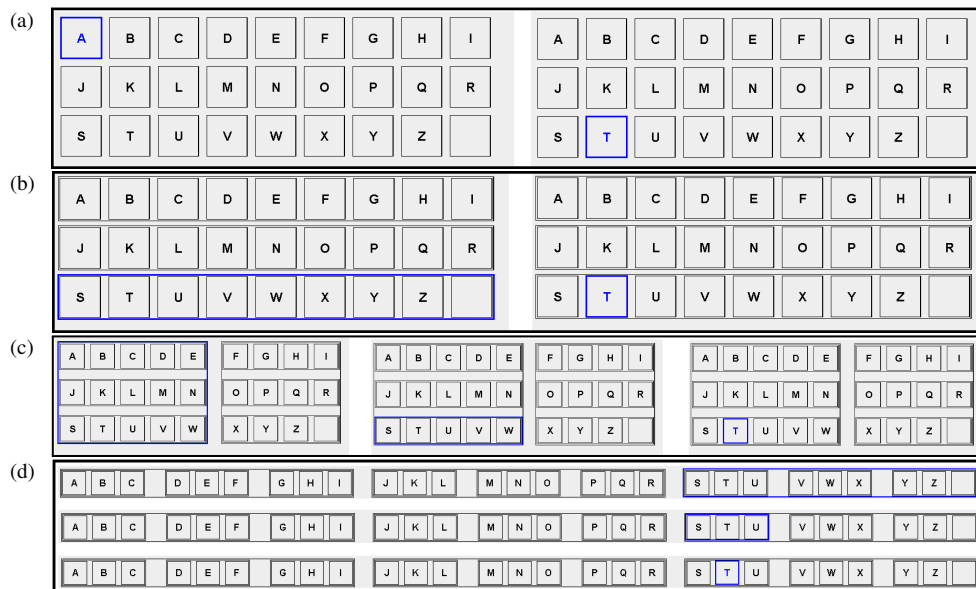


Figure 1.: Illustrations of scanning mode: (a) linear, (b) row/column, (c) half-and-half and (d) n-ary. For each scanning mode, the main steps required to select the letter "T" are presented. These views are generated by the simulation tool presented in this paper.

quired only one validation [1]. However, when the keyboard follows an alphabetic order, the typing is very slow. Indeed, the maximum number of scan steps for an alphabetically sorted keyboard is  $L$ , where  $L$  is the length of the alphabet. To overcome this limitation, keyboard with a linear scanning mode present a frequency-based organization : the most frequent letters in the considered language are presented first [1]. Another solution is to re-organise dynamically the letters ordering during text input, as explained in section 2.2.

The scanning mode that is commonly used is the **row/column** (R/C) scanning [2, 3,4]. In this scanning mode, the cursor highlights each row successively until the user validates the desired row (left of Figure 1b). When a row is selected, the cursor scans the keys (columns) of this row (right of Figure 1b). The desired letter is selected when the user validates the associated key by activating its switch. The method required two validations for one letter but the maximum number of scan steps for an alphabetically sorted keyboard is 8 [5]. Even if the number of user action is higher, the R/C scanning is faster than linear scanning [1]. Depending on the user's ability, the opposite scanning (i.e. column/row scanning) is possible.

The **half-and-half** scanning (H/H) method [6] is similar to the R/C but adds an additional first scan between two parts of the keyboard, traditionally the left and right side (left of Figure 1d). In this case, we add an additional load to the motor aspect but we divide the numbers of scan steps by two.

In  $N$ -ary search [7], at each step, the keyboard is divided into  $N$  parts. Two partition strategies can be considered: dividing the keyboard into parts with an equal number of letters or dividing the keyboard based on probability considerations. In this case, each part can contain different numbers of letters. Let's consider as an illustration the first strategy and the selection of the "t" with a ternary search keyboard ( $N = 3$ ). At the first step (top of Figure 1d), the keyboard is divided into 3 groups and the user selects the part that contains the desired letter. At the second step (middle of Figure 1d), the selected group is again divided into 3 groups and the user will also select the group that contains the "t". These steps are reproduced until the selection of the desired letter (bottom of Figure 1d). This type of scanning is the one that requires the most keystrokes to select one character [7]:  $\#steps = \lceil \log_N L \rceil$ , where  $L$  is the size of the alphabet and  $N$  is the number of groups.

## 2.2. Linguistic prediction

In order to enhance the text entry speed, some keyboards are augmented by character or word prediction modules.

**Character prediction** considers the last entered letters to predict the following one, usually by using a standard N-gram language model [8] or a lexical tree [9]. By using character prediction, we can expect to reduce the number of scan steps. Indeed, the most probable following characters are put closer to be beginning of scanning bloc. Thus theoretically improving input speed by avoiding some scan steps. However, this implies that the layout is dynamic and prevents the user from memorizing the character layout.

**Word prediction** modules offer a list of the most probable words based on either the input of previously typed words (prediction) or the letters already entered in the current word (completion). The most basic approach involves browsing a lexicon using a lexicographic tree organization [10]. However, these techniques may encounter difficulties when dealing with Out of Vocabulary Words (OOV), resulting in potential confusion.

More advanced prediction modules rely on statistical language models. They estimate the frequency of occurrence of a word according to the previous words and characters already entered in the current word. Some models offer a user adaptation, based on user's writing or even an adaption to the current theme of the input text. Word prediction enables the user to save input of certain characters by selecting a prediction by activating the switch. Moreover, this can be used by some users to reduce their spelling error. In addition to the prediction technique used, parameterisation is an important factor that enables the system to adapt to the user motor or cognitive abilities as well as its preferences (for instance factors such as the number of displayed predicted words and their positions on the layout, allowing for personalized adjustments). All of these parameter variations (scanning modes and linguistic predictions) and their combinations lead to a large number of possible scanning keyboards, not to mention the various letter organisation on the layout. Conducting experimental tests for all of these configurations, that fit most of the time very specific user needs, is not practical. Therefore, it is common practice to use simulation to reduce the design space before conducting experiments with real users.

### 2.3. Design space

In summary, in this paper we have considered the following parameters: (i) scanning mode, (ii) character/word layout per block where variations were made in the number of characters (and/or words) per block, (iii) the use (or not) of character prediction, with the possibility of dynamic rearrangement either for the entire keyboard or within each block and (iv) the use (or not) of word prediction. We are therefore interested in the Representation, Interaction and Linguistic dimensions of the DESSK framework [11], that formalize the design space of AAC and virtual keyboards. The combinations between these different parameters studied lead to a wide range of configurations that can be obtained by varying these parameters. It is not possible to test all these combinations in uses studies. For this reason, it is important to be able to evaluate the theoretical contribution of these solutions by means of a simulator.

## 3. Related works on scanning keyboards simulation

Several works have already proposed to simulate virtual keyboards. They propose to estimate the behaviour of AAC users on these devices using different metrics. Some are related to the text input speed such as character per minute (CPM) [7] or text entry rate (TER) [12], and others are related to cognitive/motor workload (such as keystroke saving and switch saving [5]). Nevertheless, these metrics are "relative" [13], and do not allow a fair comparison between different keyboard layouts. Hence, MacKenzie also proposed a new metric: the scan steps per character (SPC) [6]: it is the average number of scan steps required to write a character on the simulated keyboard.

Regardless of the simulator, the obtained results correspond to a theoretical estimation of the maximum aid that can be expected from a considered AAC system. These simulations are based on the assumption that the user has an ideal behaviour: the user promptly selects the character or prediction and enters the message without any spelling errors. The calculation of the SPC metric in [13] adds an complementary degree of theorizing. Indeed, it is not based on the writing of real texts, but on the input of isolated

words whose influence on the final SPC is mitigated by their frequency of occurrence in a reference corpus. Additionally, it should be noted that the simulator does not have the capability to simulate a keyboard with character prediction. Moreover, with just the general SPC we can't calculate the word per minute (WPM) [14] for keyboards with different scanning interval (SI) for row and column. This formula considers only a single SI, which means we cannot apply the formula for keyboards like one in [4], and thus a direct comparison between the results of the simulation and the user study is not feasible.

#### 4. STAACS<sup>3</sup>

In order to address the limitations of MacKenzie's simulator, we propose in this paper a Simulation Tool for AAC with Single-Switch Scanning (STAACS<sup>3</sup>)<sup>2</sup>. To facilitate the usage of STAACS<sup>3</sup>, text input systems are described in a simple XML format that allows to define the keys layout, the characters' arrangement, the scanning strategy, and the way to use the results of the prediction algorithms (words and/or characters).

*Metric* The results are given in the form of a CSV file, with the number of scan steps per word. The SPC value is also computed, following [13], on these data. However, the SPC does not allow to compute the theoretical WPM if the SI is different for row and column. In STAACS<sup>3</sup>, log files include  $SPC_{row}$  and  $SPC_{column}$ , representing the SPC values for rows and columns, respectively, so that theoretical WPMs can be calculated as close as possible to reality. In this case, we therefore consider the following formula to compute the WPM:

$$WPM = \frac{1000 \times 60}{5 \times (SPC_{column} \times SI_{column} + SPC_{row} \times SI_{row})} \quad (1)$$

*Language adaptation* STAACS<sup>3</sup> has been implemented to allow easy adaptation to different languages by simply modifying the test corpus and the prediction module. For example, to apply the simulator in French we have changed the prediction module to Predict4All<sup>3</sup> implemented in Life Companion<sup>4</sup>. We have also changed the test corpus to a corpus composed with the 2000 most frequent french words from [15] associated with their frequency of occurrence. According to the criteria of [16] this corpus is representative to the French language.

*Character prediction* In addition to the word prediction already offered by Mackenzie, STAACS<sup>3</sup> allows the integration of character prediction. This allows dynamic characters rearrangement, which can be global for the keyboard or within each block. This option of STAACS<sup>3</sup> allows to compare different keyboards with dynamic character rearrangement.

*Visualisation* Finally, STAACS<sup>3</sup> provides a graphical user interface that allows the user to visualise the keyboard layout and test its behaviour before simulating its theoretical performance. This allows the designer to check that the XML description corresponds to his intention. The images in figure 1 were generated with the proposed simulator.

<sup>2</sup>STAACS<sup>3</sup> is open-source: <https://github.com/MathieuRaynal/keyboardSimulator>

<sup>3</sup><https://github.com/mthebaud/predict4all>

<sup>4</sup><https://lifecompanionaac.org/>

## 5. Results and discussion

### 5.1. Results

STAACS<sup>3</sup> allows a wide range of keyboard configurations to be simulated based on variations in the parameter combinations mentioned in the section 2.3. The values considered for these parameters are given in Table 1.  $N$ -ary search can be reasonably declivable with  $N$  between 2 and 4 [7]. Keyboard layouts can take the form of a diagonal matrix (Figure 1a in [13] or a series of 1 to 5 lines balanced in terms of the number of keys (example with 4 lines in Figure 1a). All their designs are adaptable with an alphabetical or probabilistic order (i.e. letters are ordered according to their frequency in the desired language). Character prediction gives the possibility of dynamic rearrangement either globally for the whole keyboard or within each line. Finally, with word prediction, we can propose keyboards with 1 to 9 suggested words, either into the letter grid or as a list at the cardinal point (i.e. to bottom, top, left or right of the keyboard). The combination of their values leads to the wide range of scanning keyboards, not all of which are relevant.

Parameter name	Values		Count
Scanning mode	linear, R/C, H/H, N-ary [2-4]		6
Layout	diagonal matrix	alphabetical order	12
		probabilistic order	
	line [1-5]	alphabetical order	
		probabilistic order	
Character prediction	no		3
	yes	local dynamic rearrangement	
		global dynamic rearrangement	
Word prediction	no		46
	yes	position (inside, on the side)	
		number of predictions [1-9]	

Table 1.: Non-exhaustive list of parameter's values.

In Table 2 we present some keyboard simulations (with an SI of 750ms) to show the different possible analysis with STAACS<sup>3</sup>.

To begin with, we reproduced the performance of MacKenzie's simulator by simulating keyboards (Figure 1a, b and c in [13]) from [4,5]. These simulations are carried out under the same conditions as in [13] i.e. with BNC-1 and Phrases corpus[16] and we obtain exactly the same SPC results.

Schadle et al. [17] showed that character prediction improves the average number of scanning steps by about 90% for a linear scan with a keyboard of 64 keys. With the simulator, we evaluated the contribution of character prediction with a keyboard of 27 keys (i.e. the 26 letters of the Latin alphabet and space). We obtain an SPC of about 3 scan step with prediction and 10 without. Similarly, STAACS<sup>3</sup> allows us to compare and order scanning modes such as between RC and  $N$ -ary [7] or between linear and RC [17].

Finally, thanks to the addition of  $SPC_{row}$  and  $SPC_{column}$ , we can compare the result of the simulation with the performance in the user study of a keyboard with two SI. The simulation of the keyboard from [4] was reproduced, on Phrases corpus, with the same

Name	Scanning mode	Word prediction	Character prediction	Layout	SPC	WPM
a	linear	no	no	Figure 1a	10.23	<b>1.56</b>
b	linear	no	yes	Figure 1a	3.36	<b>4.76</b>
c	R/C	no	no	Figure 1b	7.49	<b>2.14</b>
d	H/H	no	no	Figure 1c	6.83	<b>2.34</b>
e	N-ary	no	no	Figure 1d	6.39	<b>2.50</b>

Table 2.: Some scanning keyboards simulations' results, obtained with STAACS<sup>3</sup>.

conditions as in user study ( $SI_{row}=250$  ms and  $SI_{column}=750$ ms) to calculate the wpm. We obtain 5.75 wpm ( $SPC_{row}=2.001$  and  $SPC_{column}=2.34$ ). In [4], the wpm range from 4.4 to 6.6, and up to 7.4 wpm with a faster user.

## 5.2. Discussion

We can see that simulation gives a good idea of potential performance of a user with a given keyboard configuration. Indeed, the simulation allows to obtain results close to a study with users. This previous result proves that simulating keyboards is a good way of filtering out different ideas, but it doesn't substitute for a user study especially with motor disabled users. Indeed, STAACS<sup>3</sup> returns an single result where a user study shows the range of possible performances. Thus, several explanations can be highlighted.

First, it is interesting to note that the data simulated by STAACS<sup>3</sup> corresponds to optimal theoretical results: the user selects the character or the prediction as soon as possible and the message is entered without spelling errors. All these things are not systematically encountered when observing the behavior of real users, particularly with motor disabled users. Therefore, we will consider future experiments with real users, in order to collect as much data as possible on human behaviour during text input in order to model such behaviour and incorporate it into the simulation.

Furthermore, for the time being, we are only simulating word input, apart from the fact that it does not fully exploit word prediction, it also does not represent a real interaction with a scanning keyboard as text input by real users might. Finally, in this version of STAACS<sup>3</sup> we do not yet integrate accents or punctuation although they are important, especially in French. This constitutes an important bias because we only considered 27 keys (without word prediction keys), against 64 in the case of a full French keyboard [17].

## 6. Conclusion

In this paper we present STAACS<sup>3</sup>, a scanning keyboard simulator that allows to evaluate several configurations with an easy adaptation to different languages. We investigated the wide range of scanning keyboards (at least 8000) that can be simulated with STAACS<sup>3</sup>, and the usefulness and limits of the simulator.

In future works, we will consider future experiments with real users, in order to create a model of typing time that takes into account a theoretical model of human behavior. In future work, we will consider experiments with real users to create a predictive model that takes into account human behaviour. This model will then be integrated into

STAACS<sup>3</sup> to improve the relevance of the proposed results and to get even closer to the results obtained by users.

## Acknowledgements

This work was funded by the French national research agency (ANR) in the framework of the AAC4All project (ANR-21-CE19-0051) - <https://www.aac4all.org>

## References

- [1] Venkatagiri H. Efficient keyboard layouts for sequential access in augmentative and alternative communication. *Augmentative and Alternative Communication*. 1999;15(2):126-34.
- [2] Felzer T, MacKenzie IS, Beckerle P, Rinderknecht S. Qanti: a software tool for quick ambiguous non-standard text input. In: *Computers Helping People with Special Needs: 12th International Conference, ICCHP 2010, Vienna, Austria, July 14-16, 2010, Proceedings, Part II 12*. Springer; 2010. p. 128-35.
- [3] Sandnes FE. Can Automatic Abbreviation Expansion Improve the Text Entry Rates of Norwegian Text with Compound Words? In: *Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*; 2018. p. 1-7.
- [4] Koester HH, Levine SP. Learning and performance of able-bodied individuals using scanning systems with and without word prediction. *Assistive Technology*. 1994;6(1):42-53.
- [5] Leshner G, Moulton B, Higginbotham DJ. Techniques for augmenting scanning communication. *Augmentative and Alternative Communication*. 1998;14(2):81-101.
- [6] Sandnes FE, Eika E, Medola FO. Multi-switch scanning keyboards: A theoretical study of simultaneous parallel scans with QWERTY layout. In: *Universal Access in Human-Computer Interaction. Methods, Technologies, and Users: 12th International Conference, UAHCI 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part I 12*. Springer; 2018. p. 287-98.
- [7] Poláček O, Míkovec Z, Slavík P. Predictive scanning keyboard operated by hissing. In: *Proceedings of the 2nd IASTED International Conference Assistive Technologies*. Citeseer; 2012. p. 862-9.
- [8] Wandmacher T, Antoine JY, Poirier F, Départe JP. Sibylle, an assistive communication system adapting to the context and its user. *ACM Transactions on Accessible Computing (TACCESS)*. 2008;1(1):1-30.
- [9] Raynal M, MacKenzie IS, Merlin B. Semantic keyboard: Fast movements between keys of a soft keyboard. In: *Computers Helping People with Special Needs: 14th International Conference, ICCHP 2014, Paris, France, July 9-11, 2014, Proceedings, Part II 14*. Springer; 2014. p. 195-202.
- [10] Badr G, Raynal M. WordTree: results of a word prediction system presented thanks to a tree. In: *Universal Access in Human-Computer Interaction. Applications and Services: 5th International Conference, UAHCI 2009, Held as Part of HCI International 2009*. Springer; 2009. p. 463-71.
- [11] Raynal M, Badr G, MacKenzie IS. DESSK: description space for soft keyboards. In: *Human-Computer Interaction. Technological Innovation: Thematic Area, HCI 2022, Held as Part of the 24th HCI International Conference, HCII 2022*. Springer; 2022. p. 109-25.
- [12] Mankowski R, Simpson RC, Koester HH. Validating a model of row-column scanning. *Disability and Rehabilitation: Assistive Technology*. 2013;8(4):321-9.
- [13] MacKenzie IS. Modeling text input for single-switch scanning. In: *Computers Helping People with Special Needs: 13th International Conference, ICCHP 2012, Linz, Austria, July 11-13, 2012, Proceedings, Part II 13*. Springer; 2012. p. 423-30.
- [14] Mackenzie IS, Felzer T. SAK: Scanning ambiguous keyboard for efficient one-key text entry. *ACM Transactions on Computer-Human Interaction (TOCHI)*. 2010;17(3):1-39.
- [15] Ben Khelil C, Rayar F, Antoine JY, Hoiry L, Raynal M, Halftermeyer A. What you need is what you get: adapting word prediction of Augmentative and Alternative Communication aids to youth language. In: *25th HCI International Conference, HCII 2023*. Springer; 2023. .
- [16] MacKenzie IS, Soukoreff RW. Phrase sets for evaluating text entry techniques. In: *CHI'03 extended abstracts on Human factors in computing systems*; 2003. p. 754-5.
- [17] Schadle I, Le Pevedic B, Antoine JY, Franck P. SibyLettre: Système de prédiction de lettre pour l'aide à la saisie de texte. In: *TALN 2001 Recital 2001 (Tours, 2-5 juillet 2001)*; 2001. .