E-Assiste: A Platform Allowing Evaluation of Text Input Systems

Mathieu Raynal, Sébastien Maubert, Nadine Vigouroux, Frédéric Vella, Laurent Magnien

E-Assiste: A Platform Allowing Evaluation of Text Input Systems

Mathieu Raynal *Sébastien Maubert* *Nadine Vigouroux* *Frédéric Vella* *Laurent Magnien*

Institut de Recherche en Informatique de Toulouse (IRIT)
Dialogue InterAction Multimodalité Accessibilité et Nouvelles Technologies (DIAMANT)
118 route de Narbonne
31062 Toulouse Cedex
FRANCE
+33 (0)5 61 55 63 14
{raynal, maubert, vigourou, vella, lmagnien}@irit.fr

Abstract

In this article we describe the E-Assiste platform which is intended for the evaluation of text entry systems. This platform enables to easily evaluate a text entry system, taking advantage of the already existing agents, such as the presentation of the text for a text copy task. Furthermore, E-Assiste captures all the events in such a way that it's possible to analyze them after the experimentation. So that the analysis is easier, we defined a communication protocol, so all the sent messages have the same pattern.
To show the usability and the utility of our platform, we bring in this article two use cases of text entry evaluation with E-Assiste.

1 Introduction

For people with motor impairments, text input is important for their access to a computer and so for their independence in doing daily tasks. They often use a keyboard which is displayed on the screen (called soft keyboard) and interact with it just with a pointer (pointing device, mouth stick, head stick, laser pointer, etc.). Moreover, with the emergence of the mobile and keyboardless devices – as for example, personal digital assistants (PDA) -, more and more research is also carried out on the soft keyboards (MacKenzie & Soukoreff, 2002).

But one of the actual needs is to have tools and methods to compare text entry themselves. In fact, as it is defined in (MacKenzie & Soukoreff, 2002, p.154):

> *"An evaluation is valuable and useful if the methodology is* ***reproducible*** *and results are* ***generalizable****. "****Reproducible****" implies that other researchers can duplicate the method to confirm or refute results. … . "****Generalizable****" implies that results have implications beyond the narrow context of the controlled experiment."*

However, each one designs its own evaluation tool as (MacKenzie & Zhang, 1999) or (Isokoski, 2004) to calculate the parameters that interest him – which are often the text input speed and the error rate. Moreover the user's performances are often compared just with the use of a reference keyboard: an AZERTY or QWERTY soft keyboard (Vigouroux, Vella, Truillet & Raynal, 2004). Furthermore, designers do not use the same reference keyboard: for example, (Vigouroux et al., 2004), (Isokoski, 2004) and (Masui, 1998) compare their system with a reference keyboard but in each case, this reference keyboard has a different layout.

In order to compensate this problem, we designed the E-Assiste platform: a distributed and generic experimentation platform allowing to evaluate any type of text input system, i.e. both soft keyboards which are intended for disabled motor people - such as Keystrokes[1] -, and text entry devices on mobile systems – such as Menu-Augmented Soft Keyboard (Isokoski, 2004).

[1] http://www.assistiveware.com/keystrokes.php

We begin this article with a brief background on the evaluation methods and the factors which are important to design a valuable and useful experiment. We report the concepts selected for the design of the E-Assiste platform and the metrics available for the text entry research. Then we describe the E-Assiste architecture and its communication protocol. Finally we report two experimentations which used E-Assiste.

2 Background

We describe the most used approaches at the present time to evaluate text input systems. We classify them in two main categories:

- User's performance modelling which is based on predictive laws ;
- Experimentations with subjects in real use context.

2.1 User's performance modelling

User's performance modelling is widely used because it permits to compare easily performance on several systems. Moreover, it is used to find a limit under of the system's performance because the models used to simulate the performance are intended for expert user. Soukoreff and MacKenzie's model (Soukoreff & MacKenzie, 1995) are the most frequently used. This model predicts user's performance by adding the predicted movement times – calculated according to Fitts' law (Fitts, 1954) - between each digraph. Each of these is weighed by the frequencies of occurrence of the digraph P_{ij}. Then, the average time in seconds for typing a character is:

$$\overline{MT} = \sum_{i=1}^{27}\sum_{j=1}^{27}\frac{P_{ij}}{IP}\left[\log_2\left(\frac{D_{ij}}{W_j}+1\right)\right]$$

With D_{ij} which is the distance between the key associated to the characters *i* and *j* respectively. W_j is the size of the key associated to the character j. IP is the Fitts' index of performance. It defines if the simulation is for an expert user or a novice user.

Another model is the GOMS model (Card, Moran & Newell, 1983) based on Fitts' law (Fitts, 1954) and Hick-Hyman's law (Hick, 1952) (Hyman, 1953). This model gives the time prediction task according to a scenario. These psychophysics laws have been used on various soft keyboards.

2.2 Experimentation methods

2.2.1 What do we have to estimate?

When we evaluate a text input system, two values are mainly studied: the speed and the accuracy. The speed is measured by all designers in one of these two ways: it is the number of characters tapped per second (cps) or the number of words per minute (wpm).

But the calculation of error rate depends more on the method used to evaluate the text input system. In the case of a copy task, Soukoreff and MacKenzie proposed a set of measures to study them: minimum string distance (MSD) (Soukoreff & MacKenzie, 2001) which is based on the distance of Levenshtein (Levenshtein, 1966) and keystrokes per characters (KSPC) (MacKenzie, 2002).

2.2.2 How to evaluate a text input system?

The evaluation of a text input system consists generally in a copy task by several subjects. This task generally consists in copying patterns (Isokoski, 2004), a set of short phrases (MacKenzie & Zhang, 1999) or a text (Vigouroux et al. 2004). It can also be a text creation to the initiative of the subject. However MacKenzie in

(MacKenzie & Soukoreff, 2002) shows that there are more inconveniences in the creation than in a copy task. Indeed, it is more difficult to control the narrow context of the experiment.

The copy task mainly raises one problem: it is called the Focus of Attention (FOA) which was defined by Mackenzie in (MacKenzie & Soukoreff, 2002) as being the attention required by the task. To try to mitigate this problem, several text presentation models are proposed.

The most used are:

- To display on the screen the text to copy – called source text - (Matias, MacKenzie & Buxton, 1996). The text entered by the subject is displayed between each line of the copy text paragraph;
- To present small phrases to the subject who must remember it before entering it (MacKenzie & Zhang, 1999). For this type of exercise, Mackenzie proposes a set of small phrases (MacKenzie & Soukoreff, 2004).
- To dictate the texts through an audio channel (Ward, Blackwell & MacKay, 2000).

3 E-Assiste platform

We designed our platform (called E-Assiste) to make experiments while being as little intrusive as possible, and nearest as possible to *in vivo* experimentation. For this reason, subjects use the text input system in their living environment. Indeed, to prove the efficiency of a text input task on a mobile device, it is preferable that this one should be done in a real mobility context. In the same way, disabled motor people often have a pointing device adapted to their disability: so it is essential for them to have this one to carry out the text entry task in appropriate conditions.

The objective of E-Assiste is to capture and save all the events (called traces) produced by the subject during the experiment. Indeed, these traces are centralized and once the text entry task is finished, it is possible to analyze them, i.e. to calculate, for example, the error rate, the number of characters typed per second ... This evaluation process is made in a transparent way for the subject.

3.1 Architecture

To achieve this goal, we built a distributed platform (Figure 1), i.e.:

- On the one hand, we have an E-Assiste part in our laboratory. This part called server saves the interaction traces of the different experimentations controlled through the experiment design.
- On the other hand an E-Assiste part called client is on the subject's device – this device can be a computer, a mobile device if it is an experiment on mobility. This client contains the text input system (defined by layout representation, device pointing, predictive system) which must be evaluated, and a system to display the source text.

Finally, E-Assiste is designed to be as modular as possible. We have defined software agents as being a part of server or client – for example soft keyboards, prediction systems, or analysis tools - which can be changed according the goal of experiment. The agents communicate only by standardized textual messages thus we can replace easily an agent by another if this one respects this communication by messages. On this way, for each experimentation design, it is possible to change as well the soft keyboard to evaluate, as the device that displays the source text and/or the analysis agents. This modularity can be important according to the use context. For example, the text to be copied could be presented differently if the task is on a PC or a PDA.

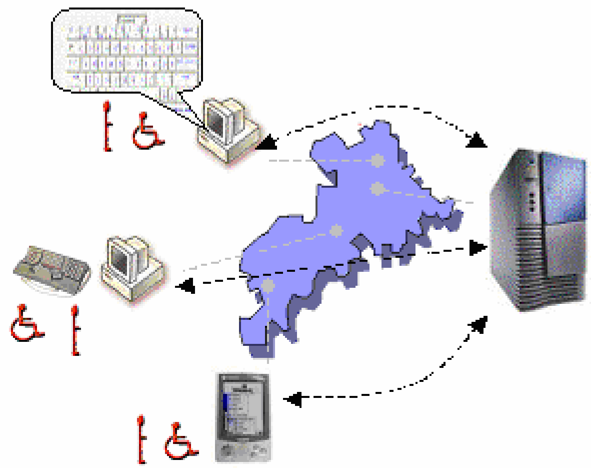


Figure 1 : E-Assiste architecture

The interaction traces between subject and system - mouse events (moved, pressed or released), character entered or deleted, list of characters predicted – and the communication between the different agents on the client are sent in real time to the server according to the communication protocol. The events collected by the server can be displayed in real time and it is possible to analyze them at the end of experiment. The analysis is also modular: each designer can create his own analysis agent which can be connected to the server if it respects the communication protocol.

3.2 The inter agent communication

3.2.1 The communication protocol

We defined a generic communication protocol between the text entry system and the platform. This protocol describes all events which could be interesting for the analysis of the user's behaviour: mouse events, keyboarded characters, etc. The communication between the different agents is also taken into account, such as characters sent by the soft keyboard to a prediction system and vice versa words or characters returned by a prediction system to the soft keyboard. The management event of the experiment is described too in this protocol: words or text that subject must keyboard in a copy task, exercise beginning, etc. For each event, we defined required and optional arguments.

Any text input system which would be integrated into the platform should respect this protocol. Thus, the server part always receives the same type of message, which simplifies the event analysis and permit to compare the different text input systems.

In this protocol we describe two methods for event transmission:

- On the one hand, by text messages sent on the Ivy software bus (Buisson et al., 2002) for real-time communication;
- On the other hand, by a markup language for data backup.

3.2.2 The Ivy software bus

For real-time communication, we use the Ivy software bus (Buisson et al., 2002). This technology enables to make different agents communicate only by text messages. These agents can be programmed with different languages, run on different operating systems and on distant computers. Each agent can send messages on the bus and subscribe to the message patterns (defined by regular expressions) that it wants to receive.

In our case, each event which occurs on an agent is sent on the software bus according to the format defined in the communication protocol. The server subscribes to all message patterns described in this protocol. In this way it receives all messages from all agents. Agents also subscribe to some messages. For example, prediction system agents subscribe to messages sent when characters are keyboarded.

3.2.3 The KTML language

In addition to the real-time communication via Ivy, we save event data progressively in a file, according to a description format called Keyboard Trace Markup Language (KTML), based on eXtended Markup Language (XML)[2].

Such a file is created at the beginning of each experimentation session and is saved at the end of this one on the subject device. This one permits us to have a backup if the client/server communication fails during the experiment. The server application can also backup its received data in this file format. Thus, we could load this backup on a future experiment and then compare these saved results with the new experiment.

3.2.4 A use example

The following table shows what happens about sent and saved messages, when a character is keyboarded on a text entry system.

Events	Ivy messages	KTML messages
Left mouse button is pressed on coordinates (198,151), at time 50718 ms.	E-Assiste:MousePressed x="198" y="151" t="50718" type="left"	<MousePressed x="198" y="151" t="50718" type="left"/>
Character 'd' is keyboarded at time 50728 ms.	E-Assiste:TextInput t="50728" string="d" type="fixe"	<TextInput t="50728" string="d" type="fixe"/>
Left mouse button is released on coordinates (198,151), at time 50890 ms	E-Assiste:MouseReleased x="198" y="151" t="50890"	<MouseReleased x="198" y="151" t="50890"/>

The argument t is the time that passed since the experiment beginning, in milliseconds. This argument is required for all messages. It enables us to reconstitute exactly all what happened during an experiment.

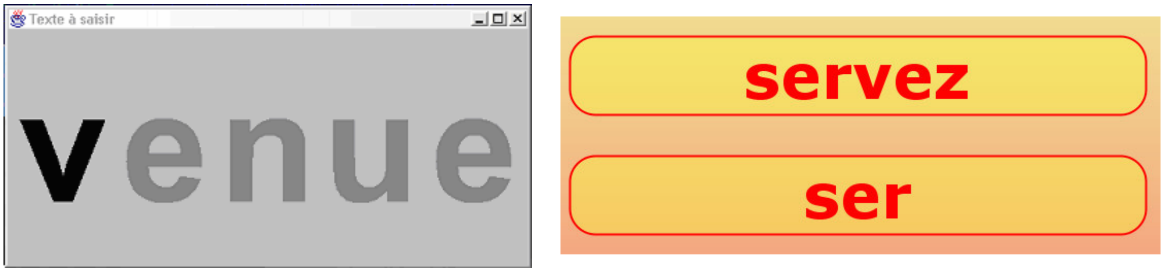
3.3 Basic agents

3.3.1 Presentation strip of source text for a copy task

In the current version of E-Assiste, we propose both text creation and text copy tasks. For the copy task, we have designed 2 types of source text presentation (Figure 2):

- The first is used for a text copy (Figure 2 a.). It is a strip which displays the N first characters of the text the subject must type. After each character entered by the subject the text moves to the left if the keyboarded character is similar to the character at the left of the strip. In case of an error, there is a visual and audio feedback, and the strip does not move while the subject enters an erroneous character;
- The second has been designed for a copy task of isolated words (Figure 2. b). The source word is presented on a line, and the result of the user keyboarding appears on the lower line. The text entry errors are not displayed on the result line and in the same way that the former, there is a visual and audio feedback.

[2] http://www.w3.org/XML/



a) **b)**

Figure 2 : Source text presentation for a text copy task

3.3.2 Metrics

We have an analysis agent which enables us to calculate the basic metrics at the end of an experiment. According to this one, we calculate the time of the experiment, the text entry speed (in cps or wpm) and the number of text entry errors.

Furthermore, as we work mainly on the problem of movement minimization for the disabled motor people, we added two measures of distance:

- The distance really crossed by the cursor of the pointing device;
- The "optimal" distance that is calculated according to the straight lines which separate each click on the keyboard.

3.3.3 Other functionalities

We added graphical agents (on the right of Figure 3) on the server to interpret more easily the experiment results. These agents are:

- A histogram of all the click times corresponding to entered characters. Click time is measured between the press and the release of the pointing device button.
- A curve giving the speed of the cursor of the pointing device during the experiment time;
- A re-play agent of the experiment which permits designer to know the subject behaviour.

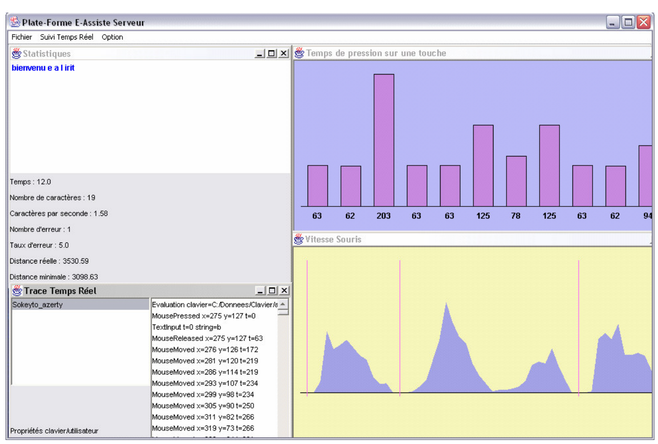


Figure 3 : Data displayed on the server screen

On the client, we added an agent which can interpret a SOKEYTO file and build the corresponding keyboard. This file is the result of the design of a keyboard with the SOKEYTO platform (Vella, Vigouroux & Truillet, 2003). This platform enables each one to design his own keyboard according to his needs. Thus, after the design of his keyboard a subject can evaluate it with E-Assiste.

4 Experiments

Several case studies are in progress through the E-Assiste platform (comparison layout, transparency augmented keys).

4.1 Evaluation of ANNIE keyboard

4.1.1 Presentation of ANNIE keyboard

The subject daily used the AZERTY virtual keyboard named Clavicom. But, he felt the fatigue with this keyboard after a long typing task. To overcome this problem, he designed the ANNIE keyboard (Figure 4) with the SOKEYTO platform. To reduce this effect, he positioned the most frequent letters in the centre of the soft keyboard. We also point out how he organized the arrangement of the characters around the space key (first round the vowels, second round the most frequent consonants).

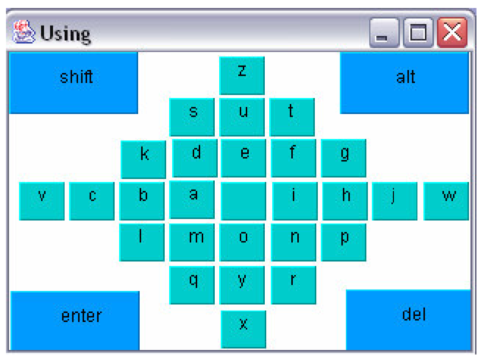


Figure 4 : Annie Keyboard

4.1.2 Case study: comparison between ANNIE and AZERTY keyboard layouts

We realize the experiment with E-Assiste. The SOKEYTO file of the Annie Keyboard was interpreted by the basic SOKEYTO interpreter on the E-Assiste client. During the experiment, we use the presentation strip for the text copy task (Figure 2.a). Two motor disabled subjects (both suffering of Spinal Muscular Atrophy) had to type a text, which are 1367 characters long (including spaces). The hypothesis was: the subject input text more quickly with ANNIE keyboard than AZERTY keyboard.

According to the data collected on the E-Assiste server, we observed that the subject 1 is better with ANNIE keyboard than with AZERTY keyboard. Indeed, he typed 13.59 % more quickly with ANNIE keyboard. But for the subject 2, we observed the opposite effect for speed input text: 15.52% more quickly with AZERTY keyboard. These results could be explained by the ability of the subject 1 to memorize his own layout. However they browsed more optimal distance with ANNIE keyboard than with AZERTY keyboard (52.38% for subject 1 and 18.10% for subject 2). But, the subjects 1 and 2 both made more error with ANNIE keyboard (18.51% for subject 1 and 31.25% for subject 2).

4.2 KeyGlass experimentation

4.2.1 Description

KeyGlass system (Raynal, 2004) is based on the addition of 4 keys over a virtual keyboard (Figure 5). These keys are automatically displayed - after each keyboarded character - around the key which has just been typed. The associated characters are proposed according to the word that the user is tapping. Not to disturb the user, the added keys are semi-transparent. Thus the user always keeps a visual reference on the keyboard.

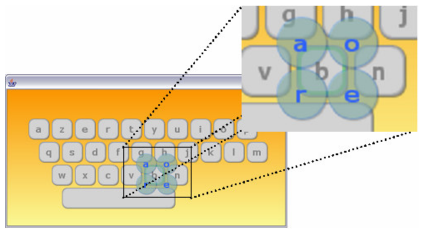


Figure 5 : KeyGlass principle: added keys after keyboarding the character 'b'

4.2.2 Experimentation

We realized the experiment of KeyGlass with E-Assiste. We used the presentation strip of word copy task (figure 2.B). Moreover, KeyGlass system integrates a system of character prediction, thus we added an analysis agent to the E-Assiste server. This one enables us to know the rate of good character prediction. Furthermore we know the rate of added characters which are used by the subject.

5 Futur works

The experiments we realized with E-Assiste are in laboratory contexts. We hope to realize experiments *in vivo* for motor handicapped people and experiments in mobility. In this context, we will adapt our text presentation agents so that this can be easily used on PDA for example.

For the analysis tools, Soukoreff and MacKenzie propose in (Soukoreff & MacKenzie, 2003) an analysis tools for the text-entry error rate. We shall connect this agent to E-Assiste as soon as they will have given it to the community as it is written in their article.

6 Conclusion

E-Assiste is a platform which enables each designer of text input device to estimate its text entry system. This platform is conceived so as to be reusable by all. It enables a designer to evaluate his system in a classic way without conceiving the necessary agents for the evaluation (source text presentation or analysis tools). However if this one needs a particular analysis, he can easily add a agent according to the communication protocol.

We so hope with our platform to enable the mutualisation of experiments and to have a corpus of results for the various text entry systems. E-Assiste enables us to compare all the systems evaluated with it.

References

Buisson, M., Bustico, A., Chatty, St., Colin, F-R., Jestin, Y., Maury, S., Mertz, Ch., Truillet, Ph. (2002). Ivy: un bus logiciel au service du développement de prototypes de systèmes interactifs. *Proceedings of the 14th French-speaking conference on Human-Computer Interaction – IHM'02,* 223-226, ACM Press.

Card, S. K., Moran, T. P., & Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of the movement. *Journal of experimental psychology*, 47, 381-391.

Hick, W. E. (1952). On the rate of gain of information. Quarterly *Journal of Experimental Psychology*, 4, 11-26.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45, 188-196.

Isokoski, P. (2004). Performance of Menu-Augmented Soft Keyboards. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '04*, 423–430. New York: ACM.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals, *Soviet Physics-Doklady*, 10, 707-710.

MacKenzie, I. S. (2002). KSPC (keystrokes per character) as a characteristic of text entry techniques. *Proceedings of the Fourth International Symposium on Human Computer Interaction with Mobile Devices*, 195-210. Heidelberg, Germany: Springer-Verlag.

MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI'03*, 754-755. New York: ACM.

MacKenzie, I. S., & Zhang, S. Z. (1999). The design and evaluation of a high performance soft keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*, 25-31. New York: ACM.

MacKenzie, I. S., & Soukoreff, R. W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*. *Lawrence Erlbaum Associates (2002).* 17, 147-198.

Masui, T. (1998). An Efficient Text Input Method for Pen-based Computers. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI'98*, 328-335. New York: ACM.

Matias, E., MacKenzie, I. S., & Buxton, W. (1996). One-handed touch-typing on a QWERTY keyboard. *Human-Computer Interaction*, 11, 1-27.

Raynal, M. (2004). Keyglasses : des touches semi-transparentes pour optimiser la saisie de texte, Rencontre Jeunes Chercheurs en Interaction Homme Machine, 93-96.

Soukoreff, R. W., & MacKenzie, I. S. (1995). Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard. *Behaviour & Information Technology*, *14*, 370-379.

Soukoreff, R. W., & MacKenzie, I. S. (2003). Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI'03,* 113-120. New York: ACM.

Soukoreff, R. W., & MacKenzie, I. S. (2001). Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic. *Extended Abstract of the ACM Conference on Human Factors in Computing System – CHI'01*, 319-320. New York: ACM.

Soukoreff, R. W, & MacKenzie I. S. (2004). Recent developments in text-entry error rate measurement. *Extended Abstracts of the ACM conference on Human Factors in Computing Systems – CHI'04, 1425-1428.* New York: ACM.

Vella, F., Vigouroux, N., & Truillet, Ph. (2003). Environnement de conception de clavier virtuel : SOKEYTO (SOftware KEYboard TOolkit), *Proceedings of the 16th French-speaking conference on Human-Computer Interaction – IHM'03,* 181-182, ACM Press.

Vigouroux N., Vella F., Truillet, Ph., & Raynal, M. (2004). Evaluation of AAC for Text Input by two groups of subjects: Able-bodies Subjects and Disabled Motor Subjects. *Adjunct Proceedings of the 8th ERCIM UI4All*.

Ward, D., Blackwell, A., & MacKay, D. (2000). Dasher -- a data entry interface using continuous gestures and language models. *Proceedings of the UIST'00 Symposium on User Interface and Software Technology*. New York: ACM.