Measuring the Effect of First Encounter with Source Code Entry for Instruction Set Architectures Using Touchscreen Devices: Evaluation of Usability Components

Mihael Kukec[1], Vlado Glavinic[2], and Sandi Ljubic[3]

[1] Medimurje University of Applied Sciences in Cakovec,
Bana Josipa Jelacica 22a, 40000 Cakovec, Croatia
mihael.kukec@mev.hr
[2] Faculty of Electrical Engineering and Computing, University of Zagreb,
Unska 3, 10000 Zagreb, Croatia
vlado.glavinic@fer.hr
[3] Faculty of Engineering, University of Rijeka,
Vukovarska 58, 51000 Rijeka, Croatia
sandi.ljubic@riteh.hr

Abstract. In this paper we address the possibility of writing program code for instruction set architectures using the touchscreen as the input device. Instruction set architecture is the common name for a collection of resources computer engineers use when developing code at the hardware level. One of the most important subsets among these resources are instructions which programmers use to create algorithms. Students enrolled in computer engineering curricula are trained to develop such solutions, using standard personal computers equipped with keyboard and mouse, thus providing them with a high level usability working environment. As technology progress has enabled the introduction of mobile platforms in the educational process, touchscreen based m-learning becomes a viable tool. To that end, in our previous research we developed a specific keyboard VMK that supports entry of assembly language code, which is based on mnemonic keys, with the aim to achieve a better efficiency of assembly coding. In the present paper we present the outcome of an improved empirical research targeting the comparison of VMK and the standard QWERTY keyboard. The results thus obtained show improved results of key usability attributes of efficiency and subjective satisfaction.

Keywords: Technology enhanced learning, usability, mobile devices, touchscreen keyboards.

1 Introduction

Though not lacking the computing power to leverage mobile learning principles [1–3] to execute complex learning systems [4, 5], mobile devices are restrained by inherent usability issues [6, 7] which are especially manifest when inputting text [8].

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2014, Part II, LNCS 8514, pp. 358–369, 2014.

As already noted elsewhere, virtual on-screen keyboards are "slow, uncomfortable, and inaccurate" [9] and "even expert typists have to look down at their fingers instead of feeling for the home row keys to situate their hands" [9]. Writing a small message can easily turn to a rather daunting mission hence writing computer software code on a touchscreen may seem like a rather impossible task.

Still, programming languages have considerably simpler grammar rules and a much smaller vocabulary than natural languages. Furthermore, assembly programming languages, which are the ones used at the Instruction Set Architecture (ISA) level, consist of simple commands (instructions) represented by appropriate mnemonics. Instructions are executed sequentially, in the order specified by the program, and the responsibility of the programmer is to create this logical sequence of instructions. Typically, creating an assembly language program involves the use of a text editor for inputting instruction names (operators) and related operands, each in a separate line. To reduce the amount of effort the user has to apply when writing assembly language programs on touchscreen devices, we have devised a rather simple approach [10], which is based on the reduction of the number of taps needed to input the instruction mnemonic using a virtual mnemonic keyboard (VMK), where each key represents one mnemonic (Fig. 1.). As assembly programs usually abound with mnemonic-only instructions, writing such program code usually reduces on tapping mnemonic (soft) keys mostly, eventually providing both efficiency and correctness of code entry.

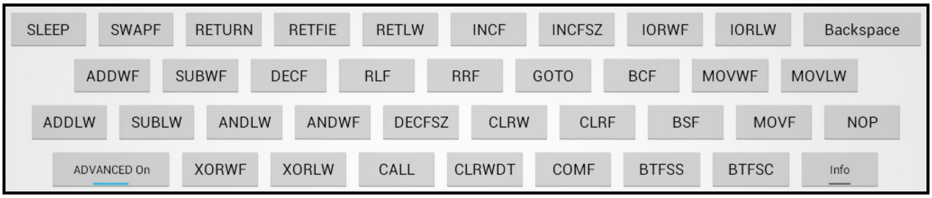

Fig. 1. Mnemonic keyboard consisting of 35 keys ("soft" buttons), with each key representing an assembly language instruction

2 Related Work

Many authors have contributed to the design of virtual keyboards and accompanying interaction styles for touchscreen devices. Virtual keyboards displayed on device screens are software generated, hence they have the ability to be transformed, modified and redesigned in many ways. The primary interaction technique for classic QWERTY virtual keyboards is direct touch. Zhai introduced a major advancement for faster writing, called *Shape Writing*, by using a sliding gesture technique [11]. Other researchers and companies developed similar techniques (e.g. *Swype* [12] and *SlideIT* [13]). The traditional concept of keyboard has also been abandoned in the Dasher project [14], which selects "flying letters" in a 2D space using a zoom-and-point interaction. To the best of our knowledge, however, no research on ISA code entry has tackled the issues we address in this paper, as all of the known solutions utilize algorithms and prediction techniques suitable for general text entry such as writing e-mails or SMSs, which makes them not exactly the most suitable for ISA code entry.

3 Design of Virtual Mnemonic Keyboard and Testing Tool

When creating a VMK (see Fig. 1.), we have based on the Microchip reduced instruction set computer architecture for PIC10, PIC12, and PIC16 microcontroller family [15], see Fig. 1. The instruction set consists of 35 mnemonics, their lengths being between 3 and 6 characters (letters) with an average of 4.6, while most of them are 5 letters long. Therefore, we could hypothesize that a VMK will decrease the time needed for assembly language program entry approximately up to a factor of 5 (i.e. replacing five taps with just one). However, there are additional aspects of human perceptual, cognitive and motoric subsystem [16] which influence the effects of using a VMK.

To evaluate our assumption, we have created a testing tool, which is an Android mobile device application, as shown in Fig. 2. and Fig. 3. The application divides the screen into two parts. The testing tool displays instructions randomly selected from a previously generated list, which the user has to type, in the upper part of the screen (approx. half of the screen real estate), while the keyboard resides in its lower half. Users entering the ISA code can switch between these keyboard layouts at every moment using a dedicated "Advanced On/Off" key, which is placed in the lower right corner of both keyboards. During the experiments, test users were however requested not to press this key unless specifically instructed to do so.

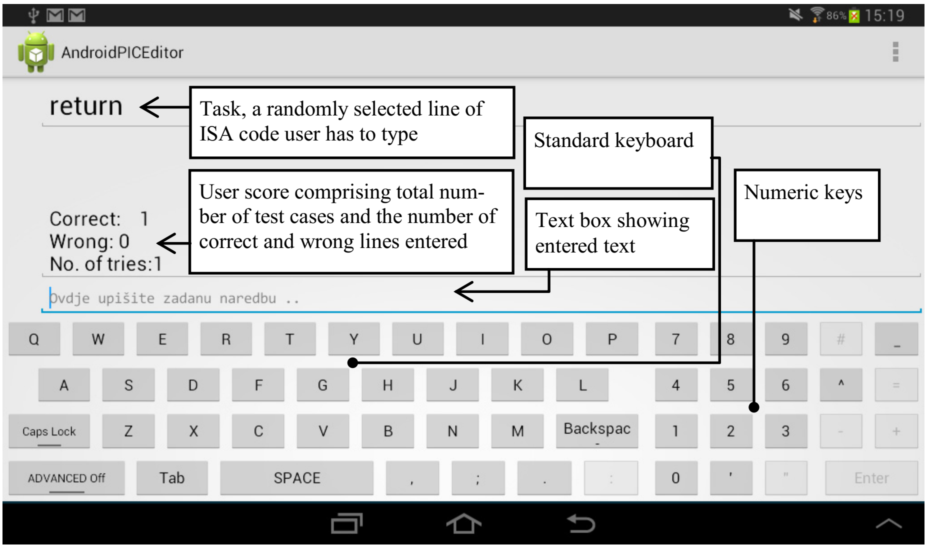

Fig. 2. Mobile device application for measuring ISA instructions reaction time and typing speed when using a standard keyboard layout. The classic QUERTY keyboard is on the left using approx. 2/3 of the available space, while its numeric part is on the right.

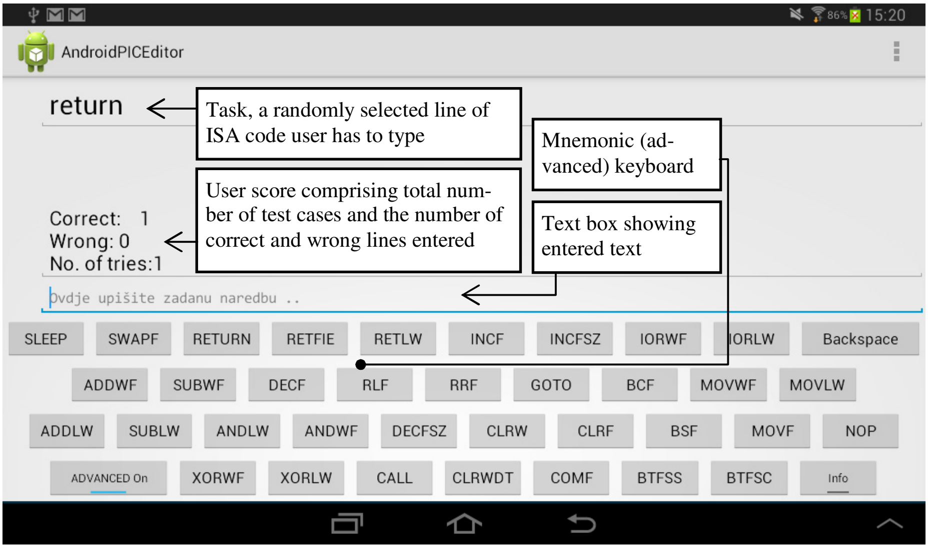

Fig. 3. Mobile device application for testing reaction time and speed of typing ISA instructions when using a mnemonic (advanced) keyboard layout. Each ISA instruction is on the keyboard represented by one key (soft button).

In the advanced mode ISA program code entry (activated with the “Advanced On/Off” key), the testing tool automatically switches between standard virtual keyboard (VSK) and VMK: the user starts typing an ISA code line using VMK, while the keyboard change from VMK to VSK is automatically performed after keying in the instruction mnemonic, as it can be seen in Fig. 4. Such behavior is necessary as each ISA code line comprises of the instruction mnemonic and (possibly) the accompanying operands which can be e.g. (binary, hexadecimal, octal or decimal) numbers, register names, labels etc.

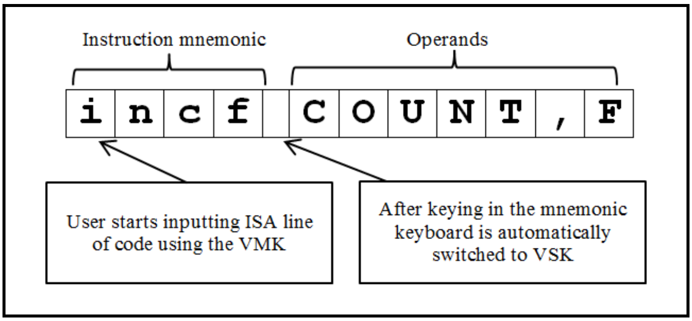

Fig. 4. Automatic switching between VSK and VMK

4 Experiment Setup and Values Measured

The initial comparison of standard desktop keyboard (DK) and standard virtual on-screen keyboard for touchscreen device (VSK) with VMK described in [10] proved that using the latter had positive effects on both the time needed to finish writing an assembly language program and the number of corrections made. Encouraged by this result we have pursued the issue further, by creating a, enhanced version of the testing application exhibiting an improved interaction method for the VMK, as described in the previous sections. The testing method has additionally been improved to reduce the effort the study group has to invest into fulfilling the assigned tasks. In our first pilot study [10] test subjects had to copy (transcribe) a 24 lines long assembly language program from paper using three different methods. This experiment setup inevitably caused unnecessary strain on test subjects because of the constant need for context changing between text on paper and its transcribed on-screen version. In the improved testing application, instructions users have to transcribe appear on the screen above the virtual keyboard she/he is using (as shown in Figs. 2 and 3), thus enabling her/him to remain focused on the task. Testing of the desktop keyboard has been dropped from this experiment.

Our study group comprised out of eight students, all male, averaging 20 years of age, that have fulfilled all the requirements for the *Computer Architecture* course at the BSc level; this allowed us to regard them as being acquainted with both the assembly language for the Microchip 14 bit instruction microcontroller family and the associated tools. The mobile device used for testing was an Android tablet computer with a 10” screen and resolution of 1280x800 pixels (Toshiba AT100-100). The study group was briefly introduced to the testing application by demonstration, while none of the participants was allowed to use it before testing actually took place. This ensured that this was their first encounter with testing application.

During the experiment, the testing application presented test subjects a continuous sequence of tasks, with each individual task consisting of one line of ISA code displayed on the screen. Tasks were randomly selected from a list of previously prepared typical lines of ISA code. Displayed tasks had to be transcribed using one of the two keyboards, either VSK or VMK.

Our primary point of interest during this experiment was to measure two values, reaction time and typing speed expressed as characters per second (CPS). Reaction time was measured as the time elapsed between the moment the task was displayed on the screen and the one the user pressed the first key in order to complete the task (i.e. the moment of pressing the “ENTER” button). The sum of reaction time and typing duration was expressed as total task execution time.

The study group was instructed to: (i) keep the tablet computer on the desk in the “landscape” orientation during the test, (ii) to use both hands for typing the ISA program code as test users would do using the classic hardware keyboard, (iii) be fast and accurate while performing the test tasks, (iv) not to rest between test tasks in order to complete as much tasks as possible. The latter request is of most importance for measuring reaction time. As a new task is displayed immediately after completing the previous one, it is very important that the members of the study group do not take a rest between tasks.

5 Results and Discussion

The study group did not have any previous experience with VMK, hence we could consider this to be the worst case scenario evaluation where learnability, satisfaction and efficiency component of usability have to be tested [17].

The testing application uses the system clock (class *android.os.SystemClock*) to retrieve time data, by executing the related system call *System.nanoTime()*. The retrieved values were converted to milliseconds and logged into a comma separated values (CSV) formatted file. After collecting data files from all of the devices used in the experiment, the CPS value was calculated for each task using ordinary spreadsheet software. Data was analyzed using IBM's Statistical Product and Service Solutions Software (IBM SPSS).

5.1 Correlation of Reaction Time and Typing Speed

The primary goal of the study is to assess the impact of users' first encounter with the VMK. Due to the lack of tactile feedback, virtual keyboards displayed on touchscreen devices create a number of difficulties to their users [8, 9], hence the reduction of the number of keystrokes per character [18] (KSPC) should help users to enter the program code in a more efficient way. Still, the effects of learnability and memorability for a new keyboard and its layout cannot be neglected. To investigate this issue further, we have examined the correlation of CPS and task reaction times for both VSK and VMK.

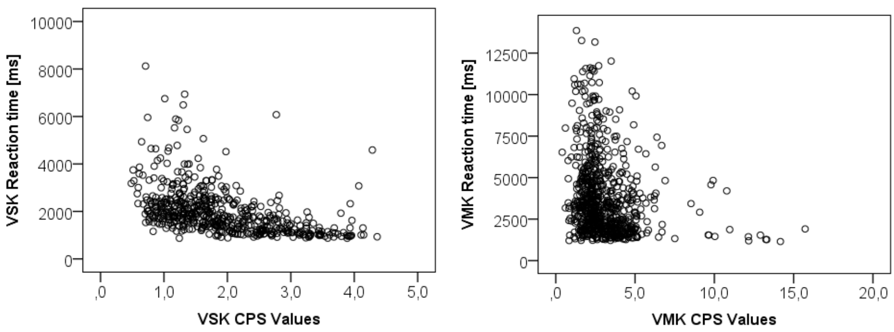

Fig. 5. Scatter plots of CPS values and reaction times for VSK and VMK

We found a moderate negative correlation between CPS and reaction time for the standard keyboard (VSK): $r(544) = -0.461$, $p<0.0005$ (see the left scatter plot on Fig. 5.). As for the mnemonic keyboard (VMK), we have found a small correlation between CPS and reaction time, $r(544) = -0.219$, $p<0.0005$ (see the right scatter plot on Fig. 5.). The coefficient of determination in the first case is 21.2%, while in the second case it is 4.8%. Both correlations are negative, which was expected, as more proficient users will have higher CPS and lower reaction time. Undoubtedly, users are more proficient in the case of VSK, hence this proficiency is observable as a higher correlation determination of VSK. On the other hand, correlation determination between CPS values and reaction times is considerably lower for VMK, since users have used the mnemonic keyboard for first time, and consequently had to apply am additional amount of effort in finding the mnemonic keys.

5.2 Comparison of Reaction Times and Speed of Typing

Data gathered from the experiment was examined before proceeding to statistical analysis. Lines of ISA code which were not typed correctly where removed from the data set submitted to analysis. It must be noted that extreme values which were outside the three interquartile (3xIQR) range were also removed. Although not directly observable from histograms (see Fig. 6.) gathered data across all combinations of factor levels violates assumption of normality, as assessed by Shapiro-Wilk's test, $p < 0.005$. However, given the robustness of the F test [19, 20] we consider the results of RM-ANOVA to be valid.

Concerning task reaction time, test subjects while using VSK reacted in 2003.92 ms, SD = 991.802 ms, on the other hand while using VMK the reaction to the task was slower: 3686.77ms, SD = 2323.744 ms. These values are depicted on Fig. 6. showing boxplots with whiskers for both reaction time and CPS values for both types of keyboards, VSK and VMK.

Repeated Measures ANOVA determined a significant difference between reaction time for VSK and VMK, $F(1, 545) = 258.164$, $p < 0.0005$. The test showed that the first encounter with VMK puts additional strain on the user trying to cope with the unfamiliar keyboard commands and new layout, which is observable in a significant increase of reaction time.

The mean CPS for VSK is 1.896, SD = 0.854. Compared to the findings described in [8] this value is rather low. The size of touchscreen button of our VSK displayed on the 10.1 inch screen was roughly 12 mm x 10 mm, while Lee and Zhai's test "soft" keyboard buttons were displayed on a touchscreen device sized 16.5mm x 10.5mm with average result of 2.6 CPS. Regarding the comparison of CPS values for VSK and VMK, RM-ANOVA determined a significant difference between 1.896 CPS, SD=0.854 for VSK and 3.057 CPS, SD=1.581 for VMK, $F(1, 545) = 223.936$, $p < 0.0005$, thus supporting our initial hypothesis which stated that VMK will have a positive effect on the efficiency of ISA code typing on touchscreen devices.

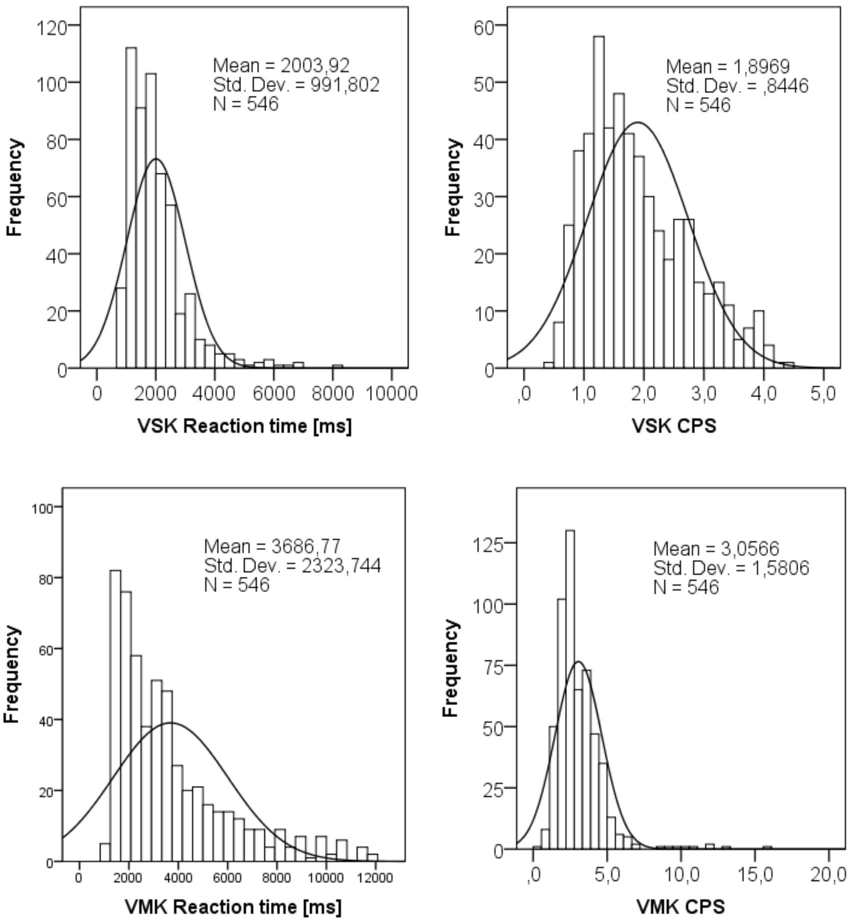

Fig. 6. Histograms of data gathered for all combinations of factor levels

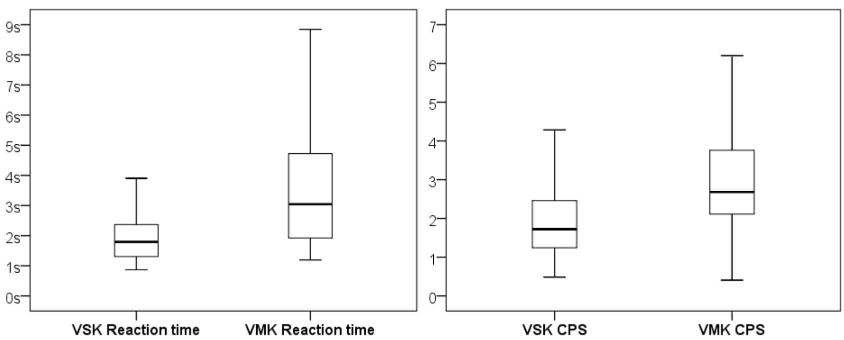

Fig. 7. Boxplot of reaction time and CPS values for VSK and VMK

5.3 Comparison of Total Task Execution Time for VSK and VMK

The initial assumptions regarding the main differences between VSK and VMK proved to be correct: RM-ANOVA showed a significant difference between reaction time for VSK and VMK. In the case of latter, the reaction time is larger by a factor of 1.84. On the other hand, owing to the reduction of KSPC for writing ISA code, VMK enables users to be significantly faster, namely by a factor of 1.61. These two factors, i.e. reaction time and CPS, are moderately negatively correlated, which is more emphasized in the case of VSK. Taking into account these two factors, it cannot easily be stated which approach (VSK or VMK) gives better overall result regarding the time needed to enter ISA code. Hence, as shown in Fig. 8, we have compared the total task time, which is calculated as the sum of reaction duration and typing duration, for both keyboards.

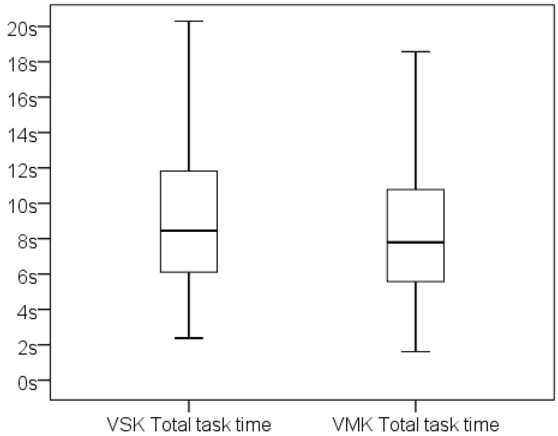

Fig. 8. Total task times for both VSK and VMK

The mean of total task times for VSK is 9.430s, SD=4.4s. For VMK, the mean of total task times is 8.575s, SD=3.8s. RM-ANOVA conducted for total task times of both conditions established a significant difference between these two mean times, $F(1, 545) = 12.579$, $p < 0.0005$.

The analysis of total task time presumes equal lengths of ISA code lines in both situations; however, as code lines (tasks) were generated randomly, we have examined instruction lengths for both of them. The mean instruction length for VSK was 11.547 characters, SD=3.046, while the one for VMK was 12.022 characters, SD=3.056. One way ANOVA showed significant difference between mean values of instruction lengths for VSK and VMK, $F(1, 1090)=6.612$, $p < 0.05$.

Taking into account all of the above findings, it follows that users enter program code 9% faster when using VMK than using the standard keyboard. This comes in spite of the fact that during the first encounter with VMK, they spend 84% more time on the average to look for the first key to type the instruction mnemonic, once they have located it, however, the reduction of KSPC will eliminate this deficiency.

Additionally, we must also note the statistically significant difference between task lengths, as in the case of VMK the respective tasks were approximately 4% longer.

5.4 Qualitative Validation of VMK

The subjective attitude towards VMK was validated through short interviews with study group members, after which students were asked to answer a questionnaire based on a seven point Likert scale. The questionnaire builds upon the theory of planned behavior (TPB) containing questions regarding: perceived ease of use, perceived usefulness, attitude towards using VMK during study projects, instructor and student readiness, subjective norm, perceived self-efficacy, learning autonomy, behavioral control, and intention [21], an contains altogether thirty items (questions). In the following we will describe student attitudes thus obtained through a short outline of questionnaire results instead of a comprehensive analysis of each item.

Two facts became evident during the short interview with the study group. Firstly, students claimed that testing was somewhat demanding: the requirement of being fast and accurate while typing on an unfamiliar keyboard put additional strain on them, thus they felt as losing pace towards the end of the test. However, the general conclusion of the study group was that they would be much faster after spending some time using VMK.

Secondly, the questionnaire results showed positive attitude towards perceived ease of use and perceived usefulness of the VMK. The prevalent grading of items in this category was within the upper part of the Likert scale (5, 6, or 7). Regarding students' opinion about the instructor's readiness to embrace m-learning, values were equally distributed around 5 which was the most used grade in this category, reflecting the students' opinion about a positive instructors' attitude towards m-learning, but with less confidence. Student readiness to embrace m-learning and subjective norm (an individual's attitude toward behavior) was mostly graded with 6 or even 7, thus emphasizing a very positive attitude towards use of mobile devices in learning. Self-confidence in using such systems is very high among the participants of the study group, and items in this category were rated mostly with grade 6 or 7, with no grade lower than 5. Furthermore, readiness to use such a system was also very high among students; again, most of the items were graded with a 6 or 7, and even fewer with grade 5. There was no grade lower that 5 in this category.

6 Conclusion

Usability is one of the main four measures of software quality, along with correctness, maintainability, and integrity [22, 23]. Inclusion of usability principles in the software engineering process in general is of utmost importance. In this respect, software systems used in learning should provide users with usable interface and effective interaction styles [24], while their subset – m-learning systems – suffer from even a greater number of additional usability problems [6, 7] inherited from mobile devices [25], that are their primary platform. The necessity of decreasing the cognitive overload

[26] for m-learning systems becomes thus increasingly prominent. The design of both the system interface and interaction methods must allow the user to focus her/his energy solely to accomplish the requested learning outcomes and create new mental structures regarding the knowledge she/he is trying to comprehend.

In this paper, we present results of quantitative and qualitative validation of a specially designed virtual mnemonic keyboard in the case of a user's first encounter with it. Using our testing system, users that had never before used such a keyboard were even more proficient than in the case of using the classic QWERTY layout, thus giving us enough confidence to state that such keyboard layout can help in the reduction of cognitive overload Indeed, test subjects involved in our experiment made similar claims, stating a positive attitude towards our keyboard design for support ISA code entry.

Acknowledgments. This paper describes the results of research being carried out within the project 036-0361994-1995 *Universal Middleware Platform for e-Learning Systems*, as well as within the program 036-1994 *Intelligent Support to Omnipresence of e-Learning Systems*, both funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Sharples, M.: The design of personal mobile technologies for lifelong learning. Comput. Educ. 34, 177–193 (2000)
2. Corlett, D., Sharples, M., Bull, S., Chan, T.: Evaluation of a mobile learning organiser for university students. J. Comput. Assist. Learn. 162–170 (2005)
3. Hwang, G.-J., Chang, H.-F.: A formative assessment-based mobile learning approach to improving the learning attitudes and achievements of students. Comput. Educ. 56, 1023–1031 (2011)
4. Holzinger, A., Kickmeier-Rust, M.D., Wassertheurer, S., Hessinger, M.: Learning performance with interactive simulations in medical education: Lessons learned from results of learning complex physiological models with the HAEMOdynamics SIMulator. Comput. Educ. 52, 292–301 (2009)
5. Hinckley, K., Zhao, S., Sarin, R., Baudisch, P., Cutrell, E., Shilman, M., Tan, D.: InkSeine: In Situ search for active note taking. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 251–260. ACM, New York (2007)
6. Holzinger, A., Nischelwitzer, A., Meisenberger, M.: Mobile Phones as a Challenge for m-Learning: Examples for Mobile Interactive Learning Objects (MILOs). In: Third IEEE International Conference on Pervasive Computing and Communications Workshops. pp. 307–311. IEEE (2005)
7. Arnedillo Sánchez, I., Sharples, M., Milrad, M., Vavoula, G.: Mobile Learning: Small Devices, Big Issues. In: Barnes, S., Montandon, L., Balacheff, N., Ludvigsen, S., de Jong, T., Lazonder, A. (eds.) Technologu-Enhanced Learning - Principles and Products, pp. 233–251. Springer, Berlin (2009)
8. Lee, S.C., Zhai, S.: The Performance of Touch Screen Soft Buttons. In: Proceedings of the 27th International Conference on Human Factors in Computing Systems CHI 2009, pp. 309–318 (2009)

9. Findlater, L., Wobbrock, J.O.: From plastic to pixels: in search of touch-typing touchscreen keyboards. Interactions 19, 44–49 (2012)
10. Kukec, M., Ljubic, S., Glavinic, V.: Improving Students' Technical Skills Using Mobile Virtual Laboratory: Pilot Study of Assembly Language Input Methods for Touchscreen Devices. In: Holzinger, A., Ziefle, M., Hitz, M., Debevc, M. (eds.) SouthCHI 2013. LNCS, vol. 7946, pp. 514–533. Springer, Heidelberg (2013)
11. Zhai, S., Kristensson, P.O.: Introduction to shape writing (2006)
12. Swype, `http://www.swype.com/`
13. SlideIT, `http://www.mobiletextinput.com/`
14. The Dasher project, `http://www.inference.phy.cam.ac.uk/dasher/`
15. Microchip 14-bit instruction set, `http://ww1.microchip.com/downloads/en/devicedoc/31029a.pdf`
16. Card, S.K., Newell, A., Moran, T.P.: The Psychology of Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale (1983)
17. Nielsen, J.: Usability Engineering. Morgan Kaufmann (1993)
18. MacKenzie, I.S.: KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques. In: Paternó, F. (ed.) Mobile HCI 2002. LNCS, vol. 2411, pp. 195–210. Springer, Heidelberg (2002)
19. Garson, D.: General Linear Models: Univariate GLM, Anova/Ancova, Repeated Measures (Statistical Associates Blue Book Series). Statistical Associates Publishers (2012)
20. Theodore, D.S.: Power of the F-Test for Nonnormal Distributions and Unequal Error Variances (1966)
21. Cheon, J., Lee, S., Crooks, S.M., Song, J.: An investigation of mobile learning readiness in higher education based on the theory of planned behavior. Comput. Educ. 59, 1054–1064 (2012)
22. Pressman, R.: Software Engineering: A Practitioner's Approach. McGraw-Hill, Inc., New York (2010)
23. Holzinger, A.: Usability engineering methods for software developers. Commun. ACM 48, 71–74 (2005)
24. Glavinić, V., Granić, A.: HCI Research for E-Learning: Adaptability and Adaptivity to Support Better User Interaction. In: Holzinger, A. (ed.) USAB 2008. LNCS, vol. 5298, pp. 359–376. Springer, Heidelberg (2008)
25. Shudong, W.S.W., Higgins, M.: Limitations of mobile phone learning. In: IEEE Int. Work. Wirel. Mob. Technol. Educ. WMTE 2005. pp. 179–181 (2005)
26. Harrison, R., Flood, D., Duce, D.: Usability of mobile applications: literature review and rationale for a new usability model. J. Interact. Sci. 1, 1 (2013)