Usability and User Satisfaction of Hardware-Software Interfacing Visualization Kit for Novice Learning Programming

Ahmad Zamzuri MOHAMAD ALI*, Siti Rosminah MD DERUS

Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, Tanjong Malim, 35900, Perak, Malaysia

Abstract

Com port is the interface of choice for most of the projects that require communication between a computer and external device or peripherals. This study has developed an experimental tool that serves as a practical hands-on learning kit for novices in exploring and mastering hardware-software interfacing in visual environment. After exploring the developed experimental tools, learners could understand the use of computer to configure, control and connect external devices via com port in visual environment. Based on the questionnaire feedback, the developed tool seems very useful and helpful for beginners, which have minimum skills in electrical and electronic engineering in understanding hardware-software interfacing programming better.

Keywords: hardware-software interfacing; programming; visualization

1. Introduction

The ICT is a bustling new industry with programming, database, software engineering, project management, networking and creative multimedia appears to be the most-wanted skills (Ungku Harun Al’Rashid, 2004). Research findings show that industries’ requirements for software engineers are high, and this indicated that programming personals are critical needs for industries nowadays (Ahmad Rizal, Nurliana & Yahya, 2008). However, the programming skills needed by the industries were lacking in most of the local graduates, which

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: zamzuri@fskik.upsi.edu.my

© 2013 The Authors. Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of The Association of Science, Education and Technology-TASET, Sakarya Universitesi, Turkey.

Available online at www.sciencedirect.com
forces industries to hire software engineers from abroad to fulfil their requirement (Knowledge Worker Exchange, 2004). Due to these, institutes of higher learning strive to plan, design and develop or revise programming courses in their curriculum to fulfil the current industries’ needs. Question arises, were the students really being prepared to meet the needs, especially in hardware-software interfacing programming areas? Basically, university’s aims are mainly to produce graduates with highly-developed intellectual, rather than practical capabilities (Moreira, 2004). The lectures in university were very theoretical and even weekly laboratory sessions were very textbook-based (Moreira, 2004). It is also found that only two to four percent preferred hands-on programming, while the majority preferred to do technical writing, software analysis and design (Moreira, 2004). Unavailability of adequate teaching and learning aids for students to experience proper practical skills might be among the reason contributing to these drawbacks. Therefore, efforts are needed to develop tools specifically designed to fulfil the requirements of beginner programmers (Pears et al., 2007).

2. Research Objective

The objective of this study is to design and develop program visualization and test its effectiveness among novice learners. The initial phase of this study was only focused on design and development of the program visualization kit for hardware software interfacing programming, specifically for beginners. The study was focused on hardware software interfacing, since most of the previous studies found on program or algorithm visualization were mainly emphasized on software and very rare or almost none on hardware software interfacing programming, specifically for beginners. Research findings have shown; educational aids developed without considering the educational principles may and may not work well. It is unarguable on the existence of abundance of hardware-software interfacing teaching aids in the commercial market. However, most of these kits were not grounded on educational research outcomes. Furthermore, most of the kits emphasized more on microcontroller technology, which is unsuitable for novice programmers in mastering hardware-software interfacing fields.

3. Development Methodology

The designing, developing and testing of hardware-software interfacing kit (ComPort V1.0) process was mainly grounded on ADDIE Instructional Design framework, which consists of the analysis, design, development, implementation and evaluation phases.

3.1 Analysis

The big picture of the complete instructional design process was established during this phase as follows:

3.1.1 Identify the content

A first step in exploring hardware software interfacing is learning how to get most from a port with the everyday application and peripherals, which include find, configure and handle (Axelson, 2000). Computer ports are mainly differentiated into two categories that are parallel communication ports and serial communication ports. Fundamentally, a port is a set of signal lines that microprocessor uses to transmit and receive data with other components. Most computer ports are digital, where each signal or bit is 0 or 1. Parallel port transfers multiple bits concurrently, while serial port transfers a bit at a time. Most personal computers come with one parallel port and a number of serial ports. On newer personal computers and laptops, parallel port is mostly not included, since nearly all the newer peripherals use serial port. Therefore, the serial port appears to be the port of choice in this study.
There are many types of serial ports available, ranging from the ubiquitous RS-232 port to a newer RS-485, USB, IEEE-1394 and IrDA interfaces (Axelson, 2000). Almost all the computers have at least one RS-232 port. This interface is useful when the computer, and the peripherals need to be connected are physically far apart. RS-232 links are slow. Along with each byte, the transmitting devices normally add a start and stop bit. Even at 115 200 bits per second, which is a typical maximum rate for a serial port, the data transfer rate with one start and stop per byte is just 11 520 bytes per second. RS-485 is another serial communication choice in computers, which can use cables as long as 4000 feet and allows up to 32 devices to connect to a single pair of wires. Other interfaces similar to RS-232 and RS-485 are RS-422 and RS-423. A better option for serial interfacing is Universal Serial Bus (USB), a project of a group that includes Intel and Microsoft (Axelson, 2000). A single USB port can have up to 127 devices communicating at either 1.5 Megabits per second or 12 Megabits per second over a 4-wire cable. The USB standard also describes both the hardware interface, and software protocols. Newer computers may have numerous USB port built-in that can be used without any added hardware and software drivers. The IEEE-1394 high-performance serial bus, also known as Firewire, is another serial interface of choice. It allows up to 63 devices to connect with a computer, with transmission rates of up to 400 Megabits per second. The 6 wire cables can be as long as 15 feet, with daisy chains extending to over 200 feet. The interface is popular for connecting digital audios and video devices. The Infrared Data Association (IrDA) interface is a wireless serial communication over distances of 3 to 6 feet. The link transmits an infrared signals at up to 115 200 bits per second. It is intended for short-range communication where cabled interface is inconvenient. Newer computers and peripherals now have IrDA interfaces built-in. In conclusion, USB is the serial interfacing of choice in this study, based upon its ability, speed and availability in most of the computer system nowadays.

Notably, serial port is harder to interface than the parallel port for parallel communication. Any peripherals connected to the serial port need the serial transmission converted to parallel for parallel interfacing. This can be done using UART. On the software side, there are many more registers need to be attended to than on a Standard Parallel Port (SPP). This study intended to develop the serial to parallel conversion hardware and the software compatible to it. Since the kit is for novice, the conversion circuit and program were not the main concern to be introduced to the learners. The main concern would be on ways to access the hardware in software. In conclusion, the main objective of this study is to develop a hardware interface to facilitate hardware visualization and software interface to facilitate program and algorithm visualization.

### 3.1.2 Identify the target learners

Programming tools were generally developed to meet professional preference's requirement; they often have extensive sets of concepts and features that are problematic for novices. For that reason, efforts are needed to develop tools specifically designed to fulfil the requirements of beginner programmers. Thus, the target learners of the kit developed would mainly be secondary and tertiary level students who have very minimum skills in electronic and electrical engineering, and hardware software interfacing programming.

### 3.1.3 Identify the environment, delivery and instructional strategies

No matter how well the visualization tool was designed, is educationally ineffective if students were not engaged actively in the learning process. Therefore, lab activities which engage various active learning approaches in promoting problem-solving skills will be the delivery and instructional strategy. This will be done in following phase.

### 3.1.4 Identify the suitable software and hardware

The main concern for beginners should be on understanding the basic programming concepts instead memorizing the rigid syntaxes and rigid commands; simpler command and syntax should be the language of choice for them.
Consequently, after gaining experience with these simpler languages, students can move to more general-purpose, commercially available languages more effectively. Therefore, Visual Basic and VB.net would be the language of choice in this study, since it’s the simplest and preferable visual programming environment in most of the institutions.

For communicating with computer, the PIC microcontroller would be the choice. The PIC microcontroller uses a component called Universal Synchronous Asynchronous Receiver Transmitter (USART) for serial to parallel conversion. This component will be configured as a Full-Duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. As pointed out earlier, USB would be the port of choice for hardware software communication.

3.1.5 Identify the constraint and quality assurance

Rapid changes in software technology might be a challenge in ensuring the long term relevancy of the kit developed. For that reason, the kit and the lab activities developed do not stress much on specific programming language. The main objective would be on assisting learners in mastering the fundamental skills of hardware software interfacing programming in visual environment.

Enough hardware visualization kit for usability testing on the larger number of samples appeared to be another challenge. The challenge was due to funding constraint. However, this challenge could be tackled by employing presentation or collaborative learning strategy in gathering feedback on usability and performance.

3.2 Design and development

As pointed out in analysis phase, the main concern of the kit developed would be on assisting novice to understand the fundamental of hardware software interfacing. Programming in hardware software interfacing often involves number systems. Binary and hexadecimal appears to be an easy-to-read way of expressing the bit and byte oriented values for addressing in hardware software interfacing programming. Binary is also known as base 2 since there can only be two values that are 0=OFF and 1=ON. Binary representations are useful in seeing the value of each bit in a byte. Numbers representing 0 are not counted. Meanwhile, numbers representing 1 are counted. Hexadecimal or base 16 is another major numbering system, which is easier to work on large numbers. The numbers are counted from 0 to 9 and followed by letters A to F. The letters A through F represent decimal values 10 through 15, respectively. Hexadecimal numbers have either h suffix or 0x prefix. Each character in hexadecimal number represents 4 bits that make hexadecimal numbers a convenient way to express 8 or 16 bit numbers. In particular, the content of ComPort V 1.0 was mainly focused on assisting novice learners in mastering these skills before proceed to programming part, particularly in visual programming environment.

Sweller, Van Merrienboer & Paas (1998) claimed, “Learners often view worked example, rather than explanatory texts, as the primary and most natural source of learning material” (p. 274). For that reason, text explanation was minimized and the worked example approach will be emphasized more. Mostly, this will be tackled in the lab manual which will be designed in the following study. However, the approach was also imposed minimally in the kit developed. Basically, worked example refers to description of how to solve problem, or in other word, states and steps needed for the solution (Caspersen & Bennedsen, 2007). Worked example approach will be followed by engagement and problem-solving approach in sequence to assist novice learners’ schema acquisition and automation in a more systematic manner. Considering the limitation of working memory, the above-mentioned tasks would be segmented and presented in different tabs with minimizing the extraneous load in mind. The first tab is brief introduction on addressing; the second tab is binary output and input manipulation; the third tab is hexadecimal output and input manipulation; the fourth tab is algorithm visualization, which consists of one
worked example, and learners could practice their own programming as well. The final tab is brief explanation on interfacing in Visual Basic.

Beside the standard VB commands, customized commands were also developed to minimize the complexity of hardware software interfacing programming (example: `out` to manage output, `inp` to manage input, `goto` to manage looping, `delay` to manage timer). Since, the main concern of the kit is to assist novice programmers in understanding the fundamental concepts, simplified commands are expected to minimize intrinsic load throughout the learning processes. A VB component was also developed for actual interfacing in Visual Basic or VB.net environment. Basically, the software architecture is depicted in Figure 1.

![Software architecture](image)

**Fig. 1. ComPort v 1.0 software architecture**

The design and development of software and hardware visualization were carefully grounded on educational principles gained from literature. Besides the layout, typography, colour and font, and other basic principles such as prevent distraction elements, avoid cluttered information, consistency, several other principles were also employed to improve cognitive economy as proposed in Naps et al. (2003) and Khuri (2001). The completed hardware visualization and software visualization are as in Figure 2 and Figure 3.
3.3 Implementation and evaluation

Throughout the development process, expert review from both academia and industry was conducted on attribute related to pedagogy and technical as well. Usability study was conducted once the design and development processes were wholly completed. The usability and user satisfaction questionnaire, which was adapted from
The Post-Study System Usability Questionnaire (PSSUQ), was used to obtain feedback from the participants. Originally, PSSUQ is a research instrument that was developed for use in scenario-based usability evaluation at IBM. The modified version of the questionnaire for this study consists of 21 items, which were divided into seven categories namely design/layout, functionality, ease of use, learnability, satisfaction, outcome/future use, and errors/system reliability. The questionnaire’s items are 7-point scales with 1 for strongly agree, followed by next numbers in sequence with the end point 7 for strongly disagree.

The research participants were 35 university students, whose ages ranged from 19 to 21, and were enrolled in a Diploma in Games course. The study was conducted during the final session of their first programming course in the curriculum. Therefore, participants have basic knowledge in structured programming. However, they have no experience in hardware software interfacing programming. Moreover, the participants also lack of skills or none at all, in electronic and electrical engineering.

Enough hardware visualization kit for usability testing on a larger number of samples appeared to be a challenge. Adding to this, the participants were beginners, it is impossible for them to explore the kit on their own due to no prior knowledge about the content. Mainly, without the lab manuals with worked examples, that will only be developed in the second phase of the study. Nevertheless, these challenges were tackled by employing presentation learning method in gathering feedback on usability and satisfaction.

The study was performed in the computer lab session that lasted for one hour. The features of ComPort V 1.0 were presented in projected presentation condition. Detailed explanations with several worked examples were given before moving to practice and discussion session. The participants went through the presentation session for about 15 minutes; they were allowed to ask question and discuss throughout the session as well. The discussion session last for around 15 minutes. The session was ended with participants answering the questionnaire.

### 4. Results and Discussion

Cronbach alpha for usability and user satisfaction questionnaire was .95, indicating that overall the scales had acceptable reliability.

**Design/Layout:** Overall in design and layout aspects, the learners liked the interface of the kit (M=2.60, SD=1.26), organization of the information presented was clear (M=2.49, SD=1.22), and the interface was pleasant to be used (M=2.51, SD=1.33).

**Functionality:** The kit has all the functions and capabilities that the learners expected (M=2.60, SD=1.41), the information retrieved from the kit was effective in helping them to complete related tasks (M=2.4, SD=1.59), and the learners agree that all the features in the kit functions well (M=2.60, SD=1.58). Additional individualized testing is needed to further affirm this claim. This will be done in following phase of the study.

**Ease of use:** The learners agree that the kit was simple to use (M=2.17, SD=1.69), the information needed was easy to find (M=2.20, SD=1.47), the information provided was clear (M=2.14, SD=1.29), and overall; the kit was easy to use (M=2.34, SD=1.64).

**Learnability:** The learners agree that it was easy to learn to use the kit (M=2.22, SD=1.51). They found that, there was too much information to read before able to use the kit (M=2.97, SD=1.65). Even so; they agree the information provided was easy to understand (M=2.56, SD=1.27). According to the written comments by learners, the information appearing in the addressing table, need to be simplified to avoid confusion. Taking this
into consideration, in the following study, the addressing table will be simplified with only the related information will be displayed in each respective tab.

**Satisfaction:** In general, the learners felt comfortable using the kit \( M=2.57, \ SD=1.48 \), they enjoyed exploring it \( M=2.29, \ SD=1.32 \), and overall they were satisfied with the kit \( M=2.51, \ SD=1.58 \).

**Outcome/Future use:** The learners agree that they could become productive quickly using the kit \( M=2.17, \ SD=1.40 \), they were convinced that the kit could improve their programming skills \( M=2.31, \ SD=1.71 \), and based on their current experience using the kit; they would use it regularly \( M=2.37, \ SD=1.23 \).

**Errors/system reliability:** The learners agree that whenever they made mistake using the kit, they could recover easily and quickly \( M=3.17, \ SD=3.36 \). They agree that the kit gave error messages that clearly told them how to solve the problem \( M=2.86, \ SD=1.62 \).

Overall outcome of the questionnaire clearly indicates learners’ satisfaction of ComPort V 1.0. However, the outcome of this questionnaire is only limited to projected presentation use of the kit. Further study needs to be conducted on personalized learning condition. By experiencing more challenging hands-on practices, the finding may differ. As discussed in earlier section, for this to take place, more organized lab manuals is needed. Further study on performance will also be conducted in the following research.

5. Conclusions

No doubt, programming appeared to be among the challenging course to be taught and learned, specifically if it involves novices. Combination of adequate teaching strategies, customized teaching aids and suitable language selection, unquestionably plays the potential role in eliminating the problems faced by novices in learning programming. Therefore, this study series intended to design and develop teaching kits to assist programming learning, focusing on hardware software interfacing programming for novice. The first phase of study was mainly concentrated on design and development of hardware visualization and software visualization. The design and development processes were carefully grounded on educational principles to ensure the maximum effectiveness of learning. The outcome of usability and user satisfaction questionnaire clearly indicates learners’ liking of the kit developed. Notably, no matter how well the visualization tool was designed, is educationally ineffective if students were not engaged actively in the learning process. Therefore, lab activities which engage various active learning approaches in promoting problem-solving skills will be the aim of second phase of the study.

**Acknowledgement**

The authors wish to acknowledge the support of MOSTI (e-Science Fund) and the Research Management Centre, Universiti Pendidikan Sultan Idris, who awarded research grants for this study.

**References**


Ahmad Zamzuri Mohamad Ali and Siti Rosminah Md Derus / Procedia - Social and Behavioral Sciences 103 (2013) 1252 – 1260


