International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.6 (2014), pp.159-170
http://dx.doi.org/10.14257/ijmue.2014.9.6.16

Exception Handling Education Approach Incorporating Role-Changing Brainstorming Technique in HCI Design Process

Jongwan Kim

Division of Computer and Information Technology, Daegu University, Korea
jwkim@daegu.ac.kr

Abstract

Human Computer Interaction (HCI) is a discipline dealing with the study, planning, and design of interactions between humans and computers. The prevalent use of digital devices is increasing the need for research and education on HCI. Paying a close attention to HCI is important as poorly designed user interfaces can lead to unexpected problems. This work is focused on a new education method geared towards reducing errors and exceptions while developing a program, incorporating role-changing brainstorming techniques during HCI design process. According to the proposed approach, two evaluation elements, including the ability to find errors and exceptions and the ability to present alternative solutions when faced with these errors, are considered. Through a role-changing brainstorming based design exploration, a low-fi prototype was developed to set up an assessment method incorporating quantitative and qualitative factors for a team-based HCI project.

Keywords: *Human computer interaction, HCI, exception handling education, role-changing brainstorming, design exploration process*

1. Introduction

Education in computer science is shifting from desktop computers to mobile devices, lectures to projects, and individual learning to collective learning. This trend is a direct result of the advent of ubiquitous computing technology which has allowed an individual to possess more than one device. Human Computer Interaction (HCI), which puts an emphasis on the interface between human and his or her computing device, is becoming an important field in research and education in computer science related fields.

HCI involves the study, planning, and design of the interaction between humans and computers. In the HCI curriculum research work supported by the ACM SIGCHI in 1992 [1], Hewett *et al.* defined HCI as a discipline concerned with the design, prototyping, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. It is often regarded as the convergence of computer science, psychology, behavioral sciences, arts, design and several other fields.

Every computer science education involves the design and the realization of a problem. A computer scientist solves the problem based on the algorithm design and its implementation. When each programmer develops software, he or she analyzes the demands of a given problem, codes a program and tests whether it works properly or not, then follows the users' requests. This research is focused on providing a HCI education plan that minimizes the users' exceptional input processes or errors that cannot be taken into account when developing a program.

The prevention of errors occurring from using software, mobile apps, or computing devices can help users by interpreting and recovering errors. Capstone design, a team project based on

ISSN: 1975-0080 IJMUE

enhancing problem solving abilities, has been introduced recently as a subject for senior students in many universities. The main reason why this subject has been introduced is because it prepares the students for the enterprise field where team work and creative problem solving is required.

In December 2003, the Federation of Korean Industries investigated 'What Industries Look for in a University Curriculum'. After surveying approximately 200 CEO's from various industries, they found out that problem solving, human relations, basics of business administration, leadership are the skills business corporations look for when hiring new employees. Among all the skills, problem solving skills such as brainstorming was considered the most important of all. This is a new study that suggests the incorporation of brainstorming technique in HCI education in terms of exception handling.

This paper is organized as following. Section 2 will present related works to this research, Section 3 will propose an educational method, Section 4 will conduct practices based on the proposed method and develop an evaluation method, and Section 5 will conclude this work.

2. Related works

With the advent of programming language, many endeavors to reduce programming errors or exceptions have performed during several decades. HCI topic also stimulates many researchers and educators to participate in this effort. As a creative problem solving methodology, brainstorming technique started getting the spotlight for a team to carry out the project.

2.1. HCI Education

HCI is a discipline studying, planning, and designing an interaction between humans and computers. There are several researches done in this area. It has been estimated that around 48% of work on a project goes into the design and implementation of the user interface [2]. Ceaparu et al. found that lost time associated with frustrating computer experiences wasted overall about 50% of a user's time spent on a computer and then pointed out that user interface (UI) design and implementation is the most important in software development [3]. Therefore, proper analysis, good design, usability evaluation as well as better understanding of human behavior are crucial in successful UI [4].

Grudin explained their historical, conceptual and cultural differences in three different disciplines: human factors and ergonomics, information system, and HCI [5]. He described the development of computer interface in three stages with the development of computers. A computer device started out as a giant machine provided only to computer system operators at the early mainframe stage. Soon after, the size of the computers shrunk and industrial managers began to use them for their businesses. Nowadays, computers and hand-held devices are widely distributed in any individuals' homes, workplaces, pockets and palms.

Peslak reviewed the literature in all areas of computer science and information systems and suggested the importance of HCI. He then determined twelve interface topics important to engineering and technology students. They are interface requirements, GUI design, interface analysis, assessment and evaluation, design psychology, human behavior and abilities, implementation, collaboration and communication, costs/benefits, testing, prototyping, and computer use disability [4]. He developed custom exercises related to the twelve topics and ordered 32 students to do the exercises. The experimental results showed that the developed assignments were statistically meaningful.

Klemmer presented CS147 Introduction to Human-Computer Interaction course (Fall 2012) in his open online lecture site [6]. In this course, students will learn how to design

technologies that bring people pleasure rather than frustration. Paper prototyping and low-fi prototyping solutions to design UI will be made and students will learn how to evaluate them. Students will furthermore learn how to conduct fieldwork with people to help them get design ideas. After students finish these stepwise practices, they will know how to develop human-centered design skills, which is essentially the main goal of his course [6]. In general, UI development process, Usability goals, User-centered design, UI design discovery, Rapid prototyping, Evaluation, and Programming implementation are included in the HCI curriculum [7].

Bringing down common errors and exceptions when developing software is a crucial part of current HCI education, but it is not specific enough. Therefore, this research seeks to provide a detailed HCI education plan that reduces common software development errors or exceptions.

2.2. Processing Errors and Exceptions

Dictionary definition tells us that an error is an act or condition of ignorant or imprudent deviation from a code of behavior. In computer science, a computer error occurs when a computer experiment releases an incorrect result. Preventing these computer errors from occurring is the key to assure users to use computers comfortably. Developing error-free software is impossible in reality. Therefore, it is important to ensure that software is made so that users are able to interpret errors while using their devices. When developing software, error message that clearly indicates the problem and constructively suggests a solution is required. In many cases, however, developers tend to focus on indicating the problem to the users in the view of them, but not on how to solve the problem itself.

Object-oriented programming language Java is a representative language that acquires the concept of exception handling. Exceptions happen when a program is executed. Exceptions include errors occurring when a program is divided by 0, when an index of an array is a negative number or is bigger than the size of the array, when the name of the reference file is different from the original file, or when the required file does not exist. When these errors occur, exception handling allows programs to continue running by catching exceptions.

In Java language, the key difference between an exception and an error in programming is that an exception is predictable while an error is unpredictable. Java's error class refers to critical errors that cannot be processed by users when they occur. Thus, they have to go through a Java interpreter, which will print an error message and stop the program. On the other hand, Java's exception class helps users prevent exceptional situations beforehand. Take the following as an example. Figure 1 is a screenshot of a simple java program that converts temperatures from Fahrenheit to Celsius, which shows how 90 degrees Fahrenheit is converted to 32 degrees Celsius. Figure 2 shows an error message that occurred from the user typing in the letter 'u' instead of a number as shown in Figure 1. Just from Figure 2, ordinary users having no knowledge of programming may not know what they have done wrong. To prevent such typing errors occurring from users, an example execution result occurring from exception handling is shown in Figure 3. Figure 3 indicates how users need to type a number instead of a letter in the Fahrenheit section through an indicating message instead of the error message in the Figure 2.

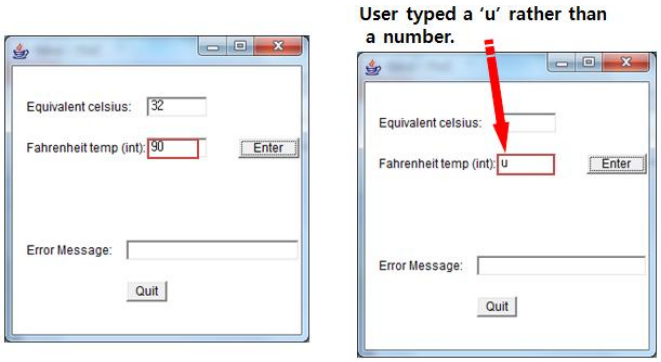


Figure 1. A Java Program Converting Degrees from Farenheit to Celsius

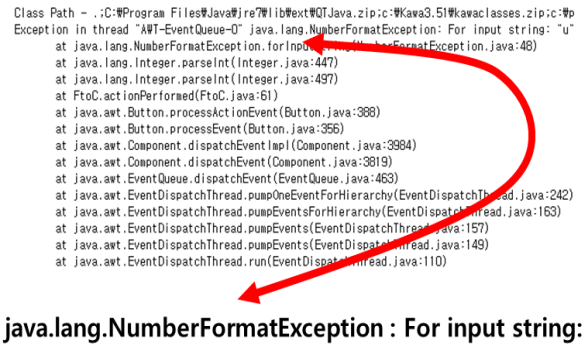


Figure 2. An Error Message Occuring from the Figure 1 Java Program

A stable system is developed so that no frustrating error messages will pop up. In this respect, it is important to prevent errors prior to them occurring. In order to prevent errors, menus or commands should be very precise. Data entry field should have appropriate default value. We would like to solve this problem with a new approach.

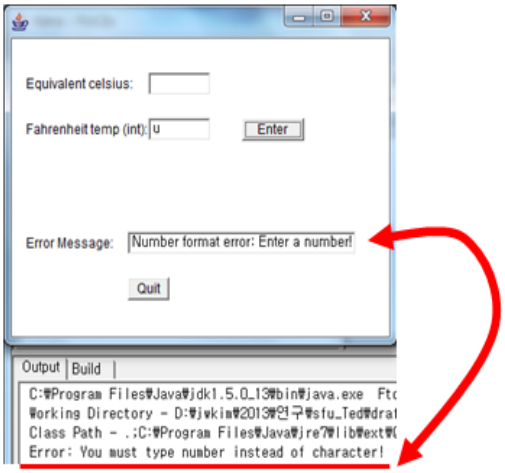


Figure 3. Execution Result After an Exception Handling for the Number Format was Performed

2.3. Problem Solving Method

Alex Osborn, a partner in an advertising agency, developed brainstorming as a method for creative problem solving in 1939. He was frustrated by his employee's inability to develop creative ideas for advertisement campaigns and began to host group-think sessions. From the sessions, he noticed the improvement in quality and quantity of ideas produced in groups compared to those produced by employees individually. Osborn subsequently published his book *Applied Imagination*, which widely distributed his brainstorming theory [8]. Today the term of brainstorming is recognized as a problem solving approach for most of group ideation sessions.

Brainstorming is performed when 3 to 10 people gather together to freely discuss on a topic. Note that a participant is not allowed to criticize another participant's opinion. It is a series of process where gathered thoughts are polished and judged into an idea. The method was first introduced in businesses, but is now being used in university classes after being recognized as an effective technique to produce ideas. Participants start brainstorming by using materials such as post-it notes or a whiteboard to think and discuss about a topic and present ideas. There are four rules behind brainstorming. First, ideas generated should not be criticized or critiqued. Second, ideas should be accepted no matter how ridiculous they are. Third, the more ideas generated, the better. Finally, new ideas should be generated from old ideas with the help of combining or hitchhiking.

The following are the solutions to when ideas get stuck during their generation process [9]. First, choose either success or failure. By imagining oneself in a bright future, as compared to a gloomy failure, ideas are said to be generated much faster. Second, put two unrelated ideas together. Sometimes, connecting two different ideas may result in a better idea. Third, stimulate associations. One may come up with an entirely new idea by imagining words or images. Fourth, apply the SCAMPER method.

SCAMPER is a method to utilize a check list in the brainstorming process. SAMPER was originally developed by Osborn, but Bob Eberle was the one whom organized these questions into a technique and gave it the name. SCAMPER is a mnemonic that stands for: Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, and Reverse or Rearrange. One uses the tool by asking questions about existing products, using each of the seven prompts above. An excellent example developed using the SCAMPER technique is Sony's global hit product Walkman, born from popular existing devices – radio, cassette recorder, and speaker. Walkman was the most popular music player ahead of Apple's iPod.

We seek to incorporate brainstorming when working on team projects in the HCI curriculum to minimize errors occurring from UI development. This kind of approach has not been reported to our knowledge yet but is expected to be an effective method of teaching.

3. Exception Handling Education based on Brainstorming Technique

After the publication of HCI curriculum research work supported from the ACM SIGCHI in 1992, many progresses have been made on HCI education. In particular, Teaching and Learning Human-Computer Interaction: Past, Present, and Future work is a milestone recently sponsored by the SIGCHI Executive Committee [10]. Churchill et al. insisted that HCI education should include problem domain specific methods and tools familiar with computer science, psychology, and engineering. They studied core topics of HCI education through international survey responses from more than 30 countries, and personal and focus group interviews. The survey respondents and interviewees wanted a unified theoretical perspective and a common curriculum and indicated that HCI should include interdisciplinary contents. Although both theory and practice are essential in HCI teaching, hands-on projects and

practical experience such as internship are equally important. Currently the ACM and IEEE computer society are revising their joint Computer Curriculum documents for 2013. During the Contextualized Curriculum for HCI workshop in the 2012 CHI conference, workshop participants discussed and formulated a context for the HCI component of the undergraduate curriculum in terms of the current teaching practices of HCI educators [11].

The above shows that HCI education is interdisciplinary, involving a combination of many studies, theory, and practice. This research focuses on proposing a brainstorming-based education plan that strengthens the ability to prepare some exceptions during a UI design process while students are working on team project.

Bourguet has classified uncertainty and error handling methods in a pervasive computing environment, without the requirement of user's conscious interaction with the system into thirty different error handling strategies [12]. According to this classification, an error handling strategy can either be the responsibility of the machine due to multimodal interface or that of the system user. He divided machine error handling strategies into error reduction by design and error reduction by context, and then identified fourteen user prevention error handling strategies that can be used in ubiquitous computing environment [12]. Here, the main focus of this work would be reinforcing error prevention in HCI education. The aim of the UI design education method is to create a general error prevention plan despite what the condition may be. In other words, the plan is not limited to kind of the device, whether it being a desktop computer or a portable mobile device, or its UI type, passive or active.

Commercialized computers such as ENIAC or MARK I from the mid-1940s used to take up around 20m by 2m office space. However, computers used nowadays range from large scale computers to handheld computers, such as tablets and smart phones. Hence, the day has come where anyone can easily gain access to and use a computer. UI design in this environment is not directed towards experts, but for all of us, requiring more caution during its development. Shneiderman identified "Eight Golden Rules of Interface Design" as a guide to good interaction design [13]. The most recent rules are to Strive for consistency, Cater to universal usability, Offer informative feedback, Design dialogs to yield closure, Prevent errors, Permit easy reversal of actions, Support internal locus of control, and Reduce short-term memory load.

Out of all the above UI design principles, our main focus will be on error and exception prevention. The most common UI design mistake occurs when there is no specific reason as to how or why an error has occurred. The mistake continues to be a problem as the user continuously makes the same mistakes over and over again, or wait indefinitely for the computer to fix the problem by itself if the system does not provide any feedback. The following example shows an error occurring from receiving no feedback for the user's mistakes. In Figure 4, for instance, there might not be any reaction after a user presses the big white button in the center or the green button on the bottom left corner of the screenshot. This is when a manual guide comes in handy, as it tells us to drag the small green call icon located on the bottom left corner of the screen upright. This error occurs because a program designer has forgotten the fact that users are not program designers or developers who know exactly what needs to be done in a certain situation. It is crucial to be able to think in the users' perspective as this is HCI's most important principle of all – in fact, the first letter H signifies Human, or user.

Norman specifically developed a Design Model for system users in his well-known book, The Design of Everyday Things [14], which focuses on a set of ideas on how a system is organized and functioned. Figure 5 is a concept map of Norman's Design Model. Good conceptual model is the key to joyful product design. Errors can occur in a system when the design model and the customer model do not correspond to each other.

Figure 4. Smart Phone Example Showing a Call Receiving Mistake

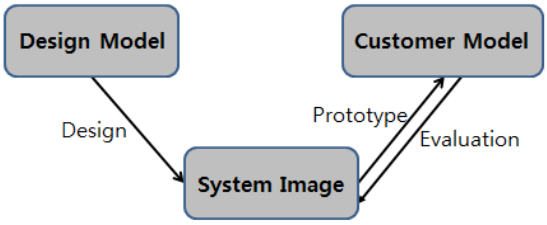


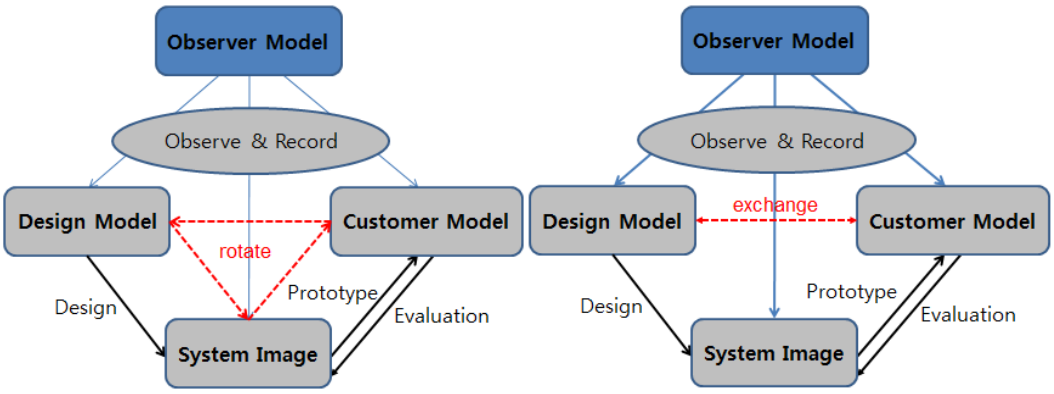
Figure 5. Norman's Design Model

In order to prevent these errors from occurring, it would be effective for team members to change being the designer's and the user's role while brainstorming. From now on, this article will focus on proposing role-changing model based design exploration process. A team would consist of 4 members, gathered together to form a design model to reduce errors. Person A develops design concepts, person B makes the prototype of a system based on the system concept, person C tests the prototype based on customer model, and person D observes and records the entire process. Design A creates the system's specification by incorporating the system's aim of accomplishment. System B makes the prototype by sketching the conceptual specification and makes up a story board. Customer C tests the usability of the system's prototype. Observer D, whom is the key member of the research, watches and records the roles of A, B, C, determines the strengths and the weaknesses of the system, and then writes a report on the findings.

In this work, two strategies to implement the proposing idea are applied. In the first strategy, we rotate participants' roles, except observer D. This would mean that the role-playing would have to be done 6 (= 3!) times, and participants need to brainstorm each time. This method of brainstorming based on dynamic role-playing would be more effective in making an error-free system where designers would better understand the customers' needs. Figure 6 outlines the proposed design exploration process of a design model that incorporates the observer model. Figure 6 (a) shows the conceptual diagram of the role-rotating design model. But this model gives a burden to students who should act as designer, developer, and customer regardless of their knowledge and skills. In general, system developer B would have its expertness compared to designer and customer, thus we fix system developer B and exchange only designer A and customer C in its role. This second approach reduces design

time and simplifies the whole process. While two students A and C exchange their roles until no improvement, a student playing customer can suggest his or her idea to resolve some problems. The second approach will be shown in Figure 6 (b). Observer D observes the behaviors of A and C, then records problems of the prototype. If there is a problem, D facilitates A, B, and C to discuss the problem, then solve it to ensure that the prototype runs smoothly. The above processes will be repeated until there are no signs of errors in the system. In a class situation, this prototype could be tested as students have learned the skills to perform both the tasks of designers and customers. Thus, this role-changing technique can be used in the brainstorming phase to help fix design problems.

Building a prototype is not limited to a paper sketch or a story board. Making a video is another way of turning down a user's frustration while using a system. The growing prevalence of smart phones and tablets has allowed anyone who desires to easily create a video, for instance, that outlines the method of using a trial product or prototype. Another way to improve user's understanding of a product would be to provide an improper example of using the product as well as a proper example. Most user manuals include a successful example along with trouble shooting methods. This research hopes to further guide the users by providing examples of error and exception handling and their solution plan found during role-changing brainstorming based design search process.



(a) Role-rotating model (b) Role-changing model

Figure 6. Proposed HCI Design Exploration Process with Observer Model

4. Practice and Evaluation

According to the proposed HCI education plan, students need to be evaluated quantitatively and qualitatively, individually and collectively. Team projects, in particular, involve private assessments as well as group assessments. Private assessments include individual assignment scores, test scores, and amounts of participation in group projects. Group assessments include creativity, complexity, completeness, presentation skills, and peer evaluation by other group members. Be sure to have a standard for marking the creativity component as it could be subjective towards the evaluator. It is important to recognize the potential to foster creativity through education and the measure of assessing the creativity aspect in class. Novelty, quantity, flexibility, feasibility, design, and usefulness were suggested to be the components of creativity in HCI education. All of these components as well as the above 8 assessment factors were taken into account and their effectiveness was tested by applying them on a team project based course for 4 years. Applying these factors were proven to be effective based on the feedback of students taking the course.

Among the above 8 evaluating factors, the four factors (individual assignment scores, test scores, degree of participation in group projects, and level of completeness in a project) are evaluated quantitatively. The remaining four elements (creativity, level of complexity in a project, presentation skills, and peer evaluation by other group members) are evaluated qualitatively. The main focus of this research is to minimize UI errors and handling of exceptions, and the ability to discover errors and exceptions and the ability to present solution ideas need to be incorporated. The appropriate assessment of the system in projects in HCI classroom is based on total ten factors. Figure 7 shows a low-fi prototype developed through a role-changing brainstorming based design exploration process. The author and several other Korean computer science faculties have participated in the Human-Computer Interaction: User Interface Design, Prototyping, & Evaluation class in Seoul, 2013 [7]. The goal of this group project was to design a new smartphone app for mass transit users in Seoul. A low-fi prototype was set to be designed by the end of the last lecture day in a 5 day course. Assuming that smartphones with contactless card technology were used, our team members interviewed several subway passengers and watched their behaviors as they went through the ticket stations in a subway station. A low-fi paper prototype was then developed; the proposed role-changing brainstorming based design exploration process was contributed to improve prototype development process.

Figure 7. Low-fi Prototype Sample Developed Through a Role-changing Brainstorming Based Design Exploration Process

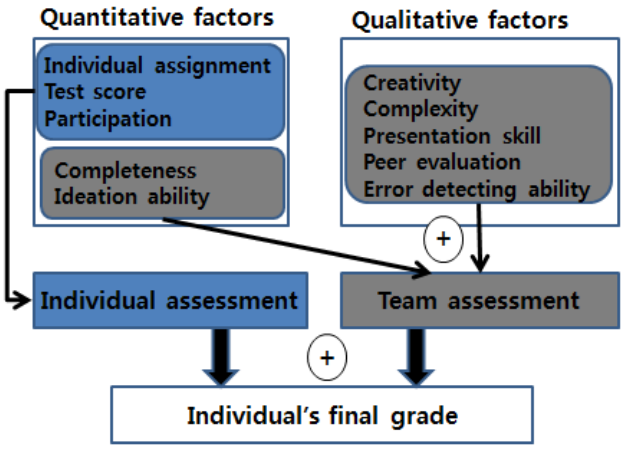


Figure 8. Assessment Method to Combine Quantitative and Qualitative Factors for a Team Based HCI Project

For every quantitative and qualitative assessment, both individual and team assessment has been incorporated. As shown in Figure 8, the first three quantitative factors are assessed individually, while the remaining two quantitative factors and all the five qualitative assessments evaluate teams. The summation of individual assessment and team assessment scores determine each individual's grade. Every subject requires a fair grading policy for assessment. Since most HCI courses put an emphasis on both individual efforts, including individual assignments and written exams, and team-cooperative efforts, a fair assessment method has been developed. This method is not limited to HCI courses; it can be implemented to any course as a standard assessment method to grade students based on their individual efforts while working on group projects.

5. Conclusion

With the prevalence of smartphones and tablets, research and education on HCI has been continually emphasized. However, its badly designed user interfaces have led to unexpected problems. In this paper, we focused on an innovative education method to reduce the occurring of exceptions and errors. To achieve this goal, we proposed an error reducing and exception handling education approach that incorporates role-changing brainstorming technique in HCI design exploration process. This proposed role changing design exploration model may not be effective in real industries. No particular skill is required for a designer to be a customer. Conversely, a customer does not have the necessary skills to act as a designer all of the sudden. In a classroom setting, however, students can practice being both the designer and the customer to reduce the occurrence of errors. The skills the students have acquired in class can be applied in their workplace after graduation. This research focuses on using role-changing brainstorming based design exploration process to develop a low-fi prototype. To evaluate prototypes developed from this process, a new HCI project assessment method has been proposed. For a better evaluation of students, two more elements – the ability to find error and exception and the ability to present alternative solution – have been added to the existing eight evaluating elements as shown in Figure 8. The effectiveness of newly proposed role-changing brainstorming and evaluation method needs to be further validated by conducting more trials in classrooms and receiving feedback from participating students.

Acknowledgments

This research was supported by the Daegu University Research Grant, 2013. The author would like to thank Prof. James Landay from University of Washington for his HCI lecture in Seoul and the faculties that performed "Seoul Unified Smartphone App Development" project. He also thanks the School of Computing at the Simon Fraser University for allowing him the opportunity to spend his sabbatical year.

References

[1] T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong and W. Verplank, "ACM SIGCHI Curricula for Human-Computer Interaction", http://sigchi.org/cdg/cdg2.html, (**1996**).
[2] B. A. Myers and M. B. Rosson, "Survey on user interface programming", Proc. of SIGCHI'92: Human Factors in Computing Systems, **(1992)** May 3-7, pp. 195-202, Monterrey, CA.
[3] I. Ceaparu, J. Lazar, K. Bessiere, J. Robinson and B. Shneiderman, "Determining causes and severity of end-user frustration", International Journal of Human-Computer Interaction, vol. 17, no. 3, **(2004)**, pp. 333-356.
[4] A. Peslak, "A Framework and Implementation of User Interface and Human-Computer Interaction Instruction", Journal of Information Technology Education, vol. 4, **(2005)**, pp. 189-205.

[5] J. Grudin, "Three Faces of Human-Computer Interaction", IEEE Annals of the History of Computing, vol. 27, no. 4, **(2005)**, pp. 46-62.
[6] S. Klemmer, CS147 Introduction to Human-Computer Interaction, http://hci.stanford.edu/courses/cs147/2012/, **(2012)**.
[7] J. Landay, Human-Computer Interaction: User Interface Design, Prototyping and Evaluation, Seoul, Korea, **(2013),** January 14-18, http://homes.cs.washington.edu/~landay/teaching/korea2013/schedule.html.
[8] A. Osborn, "Applied imagination: Principles and procedures of creative problem solving 3rd Ed.", Charles Scribner's Sons, New York: NY, **(1963)**.
[9] K. Park, Y. Park, B. Kim, H. Chang and Y. Kim, "Capstone Engineering 2nd Ed.", Infinity Books, Korea, **(2010)**.
[10] E. Churchill, "A. Bowser and J. Preece, "Teaching and Learning Human-Computer Interaction: Past, Present, and Future", Interactions, vol. 20, no. 2, **(2013)**, pp. 44-53.
[11] S. Fincher, P. Cairns and A. Blackwell, "The Contextualized Curriculum for HCI", The Contextualized Curriculum: A CHI 2012 workshop, **(2012)** May 5-10, Austin, TX, http://www.cs.kent.ac.uk/people/staff/saf/chi2012/TheContextualisedCurriculum_EA_v2.2.pdf.
[12] M.-L. Bourguet, "Uncertainty and Error Handling in Pervasive Computing: A User's Perspective", book chapter in Ubiquitous Computing, Eduard Babkin Ed., INTECH, **(2011)**.
[13] B. Shneiderman and C. Plaisant, "Designing the User Interface: Strategies for Effective Human-Computer Interaction 5th Ed.", Addison-Wesley, Reading: MA, **(2010)**.
[14] D. Norman, "The design of everyday things", Basic Books, New York: NY, **(1990)**.

Author

Jongwan Kim, He received the BS, the MS, and the PhD degree in Department of Computer Engineering from Seoul National University, Korea, in 1987, 1989, and 1994, respectively. He has been with Daegu University since 1995 and is currently a professor. His research interests include artificial intelligence, internet dysfunction, human computer interaction, and computing education.