INTELLIGENT CODING TOOL FOR PHYSICAL INTERFACE

Chaivoraporn, P[1] and Teeravarunyou, S[1]

[1]School of Architecture and Design
King Mongkut's University of Technology Thonburi
126 Phachauthit Road
Thungkru
Bangkok, Thailand 10140
Corresponding author's e-mail: ouan_jung@hotmail.com, isaknyou@kmutt.ac.th

Abstract: Usability evaluation is an increasingly important part of the user interface design process. However, gathering and analyzing the necessary data are time-consuming tasks; make inefficient use of human resources and; expose the data to subjective evaluation. There are computer-assisted tools available that could gather and record data; however, they still require human to code data. The aim of this study is to design and develop a computer data-coding assistant that could automatically code data based on relevant product's functional architecture and time-motion study. An extensive literature review was carried out to identify relevant design criteria and framework for this tool. Functional architecture of each product being tested can be fed into the system to provide framework for the system to code data received from usability trials. The tool is designed to be implemented as a user tactile tracking system that automatically tracks and records user's movements in relation to the interface. An expert walkthrough was carried out to evaluate the tool using scenario-based interaction together with questionnaires and in-depth interviews. The comments made were that the tool should increase the time consuming performance, effectiveness and the reliability of the data recorded. However, a number of recommendations were made relating to system interface design.

1. INTRODUCTION

Usability findings can vary widely when different evaluators study the same user interface, even if they use the same evaluation technique (Molich et al. 1998, 1999; Nielsen 1993). According to Molich, the first and second comparative user testing studies in 1998 and 1999 demonstrated less than a 1% overlap in findings among four and eight independent usability testing teams for evaluations of two user interfaces. This result implies a lack of systematicity or predictability in the findings of usability evaluations. Furthermore, usability evaluation typically only covers a subset of the possible actions users might take. Similar to the usability testing, there are many obstacles, which cause inaccuracy of coding data. For example, unobvious behavior occur an error from the human themselves. The behavior of using product is unique. For instance, user behavior of operating a washing machine should be totally different from the operating of handheld device. Other products may require different types of coding behavior. As a result, this research will be a research study on modern and high technology product like a digital camcorder as the pilot study. The result of study could be also applied to other industrial operations such as an assembly line.

How can systematicity of results and fuller coverage in usability assessment be achieved? One solution is to increase the number of usability teams evaluating the system and to increase the number of study participants (Souza, 2002). An alternative is to computerize assistance some aspects of usability evaluation, such as the capture, coding, or analysis. From the problems mentioned above, three explicitly objectives are setup. The aim of this research is to create a tool that it should:

- Reducing time of usability evaluation process. The new tool should decrease the time spent on usability evaluation and consequently the cost.
- Increasing consistency of the uncovered errors. The tools can consistently detect deviations from these models. It is also possible to detect usage patterns that suggest possible errors, such as immediate task cancellation.
- Reducing need of coding skill from observers. Automating some aspects of evaluation, such as the coding or analysis activities, could aid observers who do not have expertise in coding data.

2. RESEARCH METHODOLOGY

The methodology is based on experimental research with two phases of procedures. First phase of research was conducted to obtain necessary data for tool implementation. The second phase was tool implementation and experiment methods including of data collection, scope of variable, questionnaire and interview.

This research consists of a preliminary study through observation on the real situation of usability evaluation to address the problem. Although there is a lot of research about tools related to usability coding, the physical user interface is less. Almost of them were based on the computer interface. For this reason, the implementation of tool for the physical interface is demanded. It can be used for cognitive ergonomics as well. Reviewing and analyzing on available framework need to be established. The next step after seeing the whole picture of the tool was revealed. The research involves a study of usability evaluation in which professional inspection, observation, and semantic questionnaire. Then retrospective interview questions were developed in order to obtain information from participants. Two professionals and five experienced usability evaluator were conducted immediately after participant demonstrated and completed the questionnaire. All of activities were conducted in usability laboratory environment.

2.1 Preliminary Research Methodology

User Observation

Pilot study of behavior coding system consists of two handheld products for usability evaluations with two users per each product. Objective of pilot study is to define a generic activity which user performs during operating the product. Then the data was compared with the general motion described in Time-motions theory of hand motion. Surface scan about general evaluators was observed as well. The first was inspection on simple and clearly seen handheld “Game boy” testing.

Two video cameras were set up as a recording device. The first camera was set as a perspective view to record visual of hand motion. Closed up view also took place with the second camera which purpose of recording exhaustive hand motion along with facial expression. Method of mind exploring “Thinking Aloud (Ericsson & Simon, 1993)” is used in this study. Researcher asked every participant to say what they think along the testing period and audio was recorded together with synchronized visual. Recording of regular activity coding was recorded with computer tool “NOLDUS”. The second study was usability testing on “Sony™ Video Camcorder”. It is the same way as the recent testing. Simple testing tasks were defined before testing began. As well as the previous study, two video cameras were set up and coding data by NOLDUS software. Two experience users of electronic device were recruited as participants on each product testing (total of 4 participants). Testing area located in fully usability testing lab in Bangkhuntien campus of KMUTT

As the result, most of the hand’s motions were investigated based on fundamental hand motion (Frank B. Gilberth, 1927) as grasp, inspect, hold, rest, search and press. Those motions were able to separate in two different purposes. The first one was product’s operation motion sorted out by hold, grasp and press. The others such as inspect, rest and search were categorized as activity. Therefore, how to code that motion would be different depending on the objective. The activity would be coded by human.

Literature Reviews

Three problems are questioned how to improve evaluators in three scopes:

Response – How could evaluator know what is going on?

Activity - How evaluator can detect exact activity?

Interaction – How can evaluator understand that particular action?

Studying in-depth about those three stated questions were investigated. Firstly, evaluator encounters with many obstacles such as event happen quickly, continuously and suddenly. Limitation of human performance affects not only the ability to capture motion, but also the visual sensing to observe that circumstance. If evaluators need to observe for a long period of time, fatigue is in consideration. Fatigue would reduce human efficiently and attention. Problem mentioned above presents the effect of time related to coding process. However problem about visual capability and fatigue probably requires an individual experience and skill to keep it in sight.

Secondly, after evaluators detect end-user response and carry on coding, next problems of accuracy are getting in charge. Molish (1998) presented this problem in his study. Less than 1% of data overlaps in finding the same test by different evaluators. Different data they received leads to different analysis and problem identification. For example, evaluators always hesitate to decide what user's do is and sometime they cannot identify user's activity due to unclear image enough. Reference is taken charge in this session. This problem in practical sometime was originated by evaluators themselves. Missing data coding and unidentified activity related to time are spotted frequently.

Lastly, usability coding is not just only record what have been seen. Interpretative and user emotional recognition would be kept in mind. Clearer image of the context importantly contribute to discreet interpretation. Connection between observed activity and user emotion is more subjective to make that decision. Not only evaluator lacks of enough data, but also their bias significantly affect on reliability of the test. In conclusion, six keywords were analyzed as the criteria to improve evaluators' coding capability such as time, skill, reference, evaluator error, and context and bias as can be seen in table 1. The solution represents the design criteria for the data collection.

Table 1. Design criteria based on seven stage analysis

		Design criteria
Respons	Time	Less skill or skill less when performing the test.
	Skill	Reduce evaluators' coding workload.
Activity	Reference	Coding system should be provided in human language for easy interpretation.
	Evaluator error	Time and user activity should be synchronized in the same way.
Interaction	Context	Clear image of activity and emotion (facial expression) should be represented.
	Bias	Precise system and accurate data output.

Usability coding framework

In process of recording data, there are many variables and methods to code that particular data. From more subjective methods such as ethnographic observation through precise method as computer event log are based on dissimilar framework. Many frameworks relate to physical interface such as time-motion study. Next paragraph represents a related framework used to collect and analyze data. Advantage and disadvantage of each framework are in consideration in order to classify and extract the guideline for designing the tool. The framework that can be used for collecting data are Time-motion study (Gilberth,1927), Task model (Woodson, 1981), Scenario approach (Gaffney,2000),and AEIOU (Robinson,1994).

Table 2.Variable coded in observation framework

	Coded data						
Time-motion	Time	Therblig	Object				Goal
Task model	Sequence	Task	Object	User			Goal
Scenario approach	Event	Activity	Object	User	Interaction		Goal
AEIOU framework		Action	Object	User	Interaction	Environment	

From the table 2, every framework requires some similar variables to be coded. [Movement – Task – Activity – Action] refers to a particular movement of target. Those variables are always coded in every framework. Moreover, [Object] is often coded in every framework as well. In descending order, [User] and [Goal] are frequently coded by three frameworks. This analysis of framework shows that variables that frequency uses are Movement, Object, User and Goal were defined as the variable which researching tool need to be coded.

Table 3. Framework vs. seven stage variable

	Skill	Time	Reference	Evaluator error	Context	Bias
Time-motion	■		■			■
Task model			■		■	
Scenario approach					■	
AEIOU framework	■		■			■

Although time-motion study and AEIOU framework was specify obviously observation scope, bias has still existed if evaluators do not have enough experience and skill in observation. This result reiterates research objective of inefficient time usage and error obviously. Since the objective of this research is to reduce time in coding, labor coding and accuracy. Time-motion framework was selected to be an initial framework for data collection. Other frameworks such as AEIOU require less skill. However in time consumption, AEIOU trends to use time over than time-motion framework. Hence, AEIOU should not be a better way to go if objective of research is to reduce task of coding. As the result from the table 3, there are only two frameworks which have potential regarding to context.

From the tooling design stand point, completed coding by scenario approach should be troublesome due to substantial interpretation. However tool may be not able to code scenario. It can help evaluators by preparing some task data for better scenario coding. This is why task model was raised as the secondary approach. Collaboration between time-motion and task modeling results in time recoding and low-level activity. The best way to reduce task of coding is generally thinking as a computer task. Tool need to be designed in order to take charge of recording and coding primary data by a computer in real time. It should be ease of use, clear visualization and reliability.

Table 4.design criteria conclusion

Primary criteria	Introduce real time coding. Reduce the task of coding Code data regarding to time and primary activity Perform an accurate data output
Secondary criteria	Ease to understand Clear visualization Data can be rearranged for scenario coding

Analysis on Norman' seven stages of action leads to more understand about evaluator's perception, interpretation covering to what observer needs to observe an end-user. In the conclusion, Collaboration of time-motion and task model was selected to be a principle of the design.

2.2 Validation Methodology

2.2.1 Demonstration

Introduction on background and operation of research tool was demonstrated to two usability professionals and six experienced evaluators. Software interface, input and output of tool were quickly previewed containing a detailed description of the tool concept.

2.2.2 Test drive

After demonstration finished, every participant was asked to do data coding with this new tool. End-users as volunteer who inexperience with dummy product were invited to join the test. Tested end-user was not be duplicated for the next test. Base on the usability performance evaluation, participants was assigned to do around four to five tasks without telling them the instruction. This is similar to Neilson's evaluations of traditional heuristic evaluation. Each participant decided task sequence and how they would perform the evaluation.

They are self-determined in the length of time and amount of effort they would put into the evaluation. While participant performed testing, observation on both of their behavior reaction to this tool and the way they operate tool were recorded in digital files by another camcorder.

2.2.3 Questionnaire

Questionnaire is extremely useful to investigate the less observable aspects of interface design that cumulatively contribute to a user's subjective feelings of satisfaction or frustration. Furthermore, two important facts were also taken into account – it is not a design of a software interface but it is a concept of how to code data and how it helps evaluators to perform testing under evaluation.

The usability questionnaire supports testing and determination of user subjective on satisfaction with the tool algorithm, as well as their satisfaction with its ease of use, efficiency, effectiveness in their opinion, as well as with the attitude of the system during its usage. Questionnaire platform was based on comparison between researching system and manual coding either by computer assist tool or traditional pen-paper. Participants indicate level of their agreement with a questionnaire statement on a five-point likert scale. An average value taken from all participants indicates an objective value of system as pleasantness *(A.Granic, 2005).*

2.2.4 Interviews

Interviews and questionnaire is conducted immediately after the test drive. Structure of interviews is somewhat different depending on participants. Two groups of participants were divided into professional and experienced usability testing. A briefly interview script was developed for two different groups. Interviewing on professional group generally performed in a quite open-end question. Both groups are asked about the efficiency of tool inspection and satisfaction.

Some special questions were selected according to their questionnaire answers. In additional, every participant was crossed-checking by asking what were they doing during testing and described in detail about that particular activity. The answer from interview was compared to the questionnaire answer to ensure reliability. During period of interviewing, video camera was used to record.

2.2.5 Population

Usability testing goal is mainly used to explore problem and interaction between interface and users. Recruiting of participants was elaborately recognized. People involved in this research were separated into two groups of usability evaluator as participants and end-users. Reaching goal of tool inspection requires evaluators who have experience in software and physical interface. Thereupon direct users mean usability evaluator were recruited from professional usability community such as software organization, design firm and graduated students. According to Jakob Neilson's statement, a few professional is able to inspect more than 60% of interface problem *(Neilson 1993)* so two professional were persuaded to join. As local usability and design business in Thailand still in growing up period, amount of usability well-informed person are limited in narrow. Plus recommendation from Neilson in literature as 85% of an interface's problems can be found with as few as 5 participants *(Neilson 1993).* This is why 5 participants not included

professional were recruited. Among of these participants were recruited from graduated level of KMUTT's human-centered design students who completed a usability testing class.

Another group was end-users who invited to be observed by evaluators. End-users people were asked to join testing as volunteers. Total of 7 people were filtered before invitation. First, a person must not have experienced to the MP3 player. For this study, the MP3 watch player is used to evaluate since it shows a complication of use in buttons and navigation (see Fig 5). Second, their knowledge about usability evaluation should be none or superficially. Moreover, end-users behavior during test should not interfere the observation and to reduce bias of end-users about this system.

2.3 Tool Design

2.3.1 Equipment

Tool was designed base on the conclusion of the analyzed criteria and framework. Combination of physical input module and computer software were integrated into automatic coding system. Physical input data from user interaction were detected and send through the computer. The tactile sensor is QT113 (see Fig. 2) from a Quantum research group. It was designed for human touch (see Fig 1). To expand application of this tool to any physical interface in the future, sensor installation attached on a product should be designed. After investigate on many installations, sensors need to easily attach to interface while fixing component must be still have ability of electrical conduction.

New modern technology in textile industry, conductive yarn was introduced to apply as the sensor installation component. With outstanding advantage of conductive yarn when weaved to fabric, that fabric is flexible and still has fine electrical conductivity. Five millimeter square fabrics was cut in pieces and gently fuse together with electric wire and interface buttons of MP3 watch by very simple slim double-side tape (see Fig 5). Four buttons of dummy product were attached with 4 kits of wire connected to the sensor module. Sensor module was developed from microcontroller logic board which already programmed to promptly receive data from interface. The sensitivity of sensor needs to be calibrated according to the force that used to press on that application. To extend the variety of application, sensitivity of this module has ability to fine tune for the other specific applications by simply changing a capacitor in the logic board (Quantum Inc, 2007).

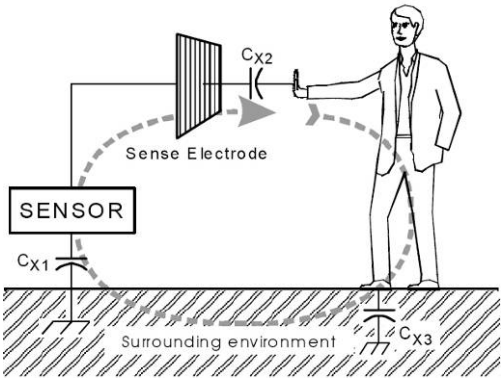


Figure 1. Touching sensor's principle

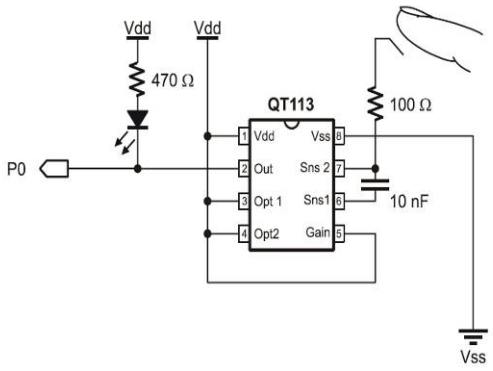


Figure 2.Touching sensor's circuit

Apart from description of data sensing side, software interface and interpretation elaborately realized. Software was approached by legendary spreadsheet software Microsoft Excel with collaborating with VBA (Microsoft visual basic for application). The left side of the primary screen was the space for coded data in form of table. Visual of user facial expression (camera #A) was closely projected into window on the top of the screen. The additional camera view (camera #B) was customizable. In this research, this camera was set up to capture overall product's activity shown on hand movement over product. This visual component was located at the center bottom of the screen. The last camera (camera #C) was used to record evaluator's behavior during performing the test. The right side of the screen was the place for problem check box, task definition box and other tool controls. In the secondary screen, it is the screen contained the tool utility windows and camera control box which is unnecessary to attend during user testing performed.

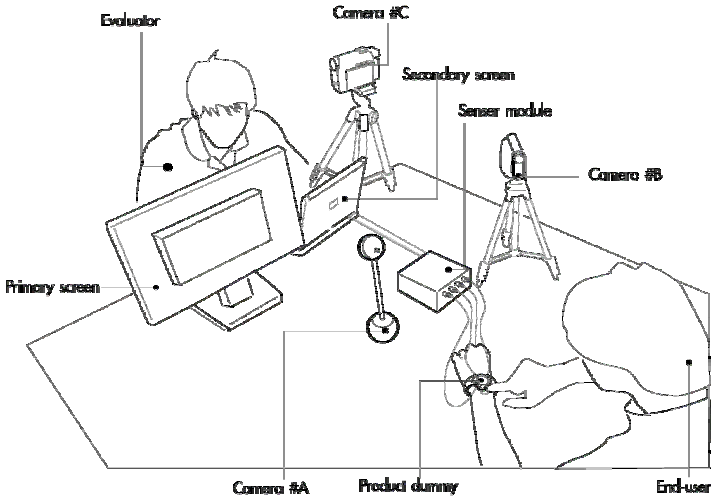


Figure 3. Testing equipment setting

2.3.2 Software interface

In general, accessibility of designed interface should be recognized ahead of usability testing plan identification. As a spontaneous manner, function map and interface's procedure were packed into coding module.

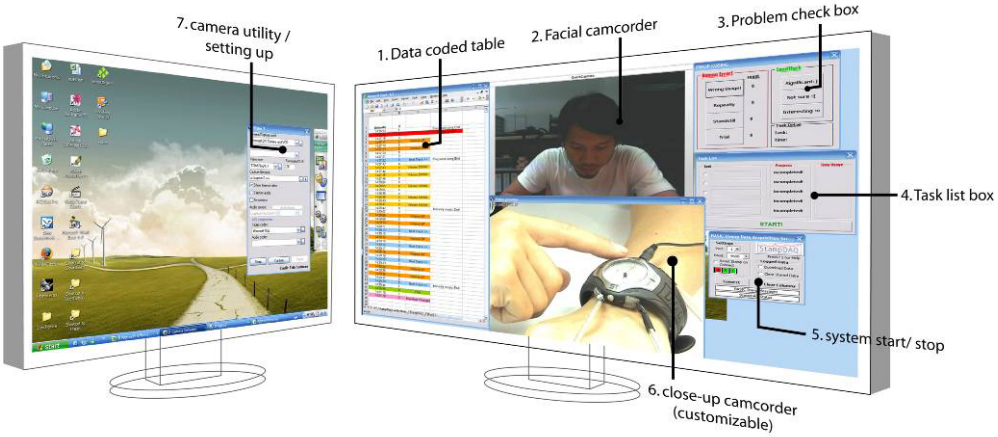
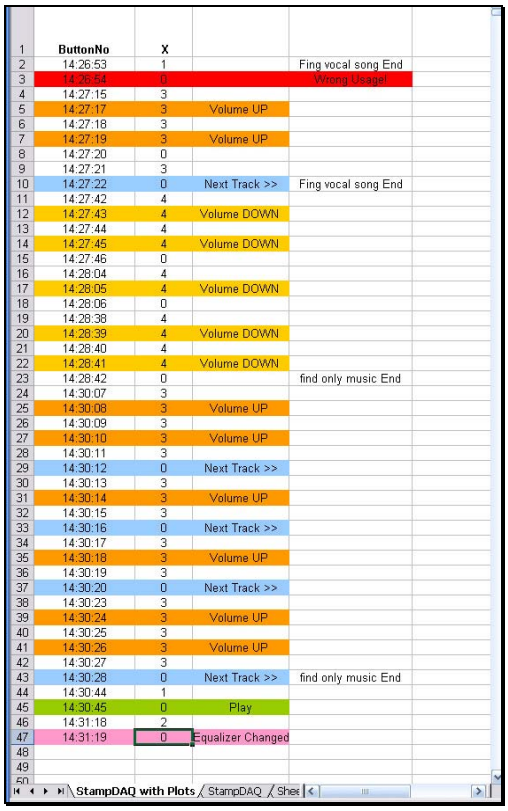


Figure 5. Display monitors for intelligent coding and a MP3 watch for testing



	ButtonNo	X		
2	14:26:53	1		Fing vocal song End
3	14:27:14	0		Wrong Usage!
4	14:27:15	3		
5	14:27:17	3	Volume UP	
6	14:27:18	3		
7	14:27:19	3	Volume UP	
8	14:27:20	0		
9	14:27:21	3		
10	14:27:22	0	Next Track >>	Fing vocal song End
11	14:27:42	4		
12	14:27:43	4	Volume DOWN	
13	14:27:44	4		
14	14:27:45	4	Volume DOWN	
15	14:27:46	0		
16	14:28:04	4		
17	14:28:05	4	Volume DOWN	
18	14:28:06	0		
19	14:28:38	4		
20	14:28:39	4	Volume DOWN	
21	14:28:40	4		
22	14:28:41	4	Volume DOWN	
23	14:28:42	0		find only music End
24	14:30:07	3		
25	14:30:08	3	Volume UP	
26	14:30:09	3		
27	14:30:10	3	Volume UP	
28	14:30:11	3		
29	14:30:12	0	Next Track >>	
30	14:30:13	3		
31	14:30:14	3	Volume UP	
32	14:30:15	3		
33	14:30:16	0	Next Track >>	
34	14:30:17	3		
35	14:30:18	3	Volume UP	
36	14:30:19	3		
37	14:30:20	0	Next Track >>	
38	14:30:23	3		
39	14:30:24	3	Volume UP	
40	14:30:25	3		
41	14:30:26	3	Volume UP	
42	14:30:27	3		
43	14:30:28	0	Next Track >>	find only music End
44	14:30:44	1		
45	14:30:45	0	Play	
46	14:31:18	2		
47	14:31:19	0	Equalizer Changed	

StampDAQ with Plots / StampDAQ / Sheet

Figure 6. “Coded data table” window

Software would interpret data and data sequence refers to the structure stipulated. Interpreted data was automatically coded in the 3rd column (see Figure. 6) while two previous columns showed time that activity happened and raw coded data respectively. Data in the table updates at any times when system detects the interaction between user and product interface. In order to explicit the visualization during system abruptly code data, table’s row color was represented differently on what data received.

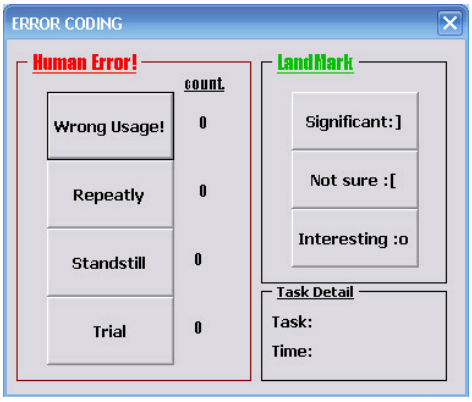


Figure 7. “Error Coding” window

Figure 8. “Task list” window

Task list box was designed to help evaluator assign the task name which appeared in the coding table. While assigned task was activating, task status shown in the next column would detect from “incomplete” to “collecting data”. Eventually word changed to “finished” in the end. After each task was finished, time usage on any task was calculated and display in the “Time Usage” column on the right side. To assist evaluator review coded data and observation records throughout real-time problem detection easier than ever, interface of problem marking was introduced as also. According to preliminary research and interviewing with professional, generic incident which leads to be human error which always found and

noticeable were classified into 4 incidents such as “Wrong usage”, “Repeat”, “Standstill” and “Trial”. Evaluator just clicks on each button to code into the table. At that moment, error wording would be inputted into the coding table at the 4th column and emphasized with red color. Occasionally, incident which evaluator was wondering were not in the human error list. Evaluator still remains ability to mark that incident by clicking on the “Landmark” frame (weight). Wording of “Significant :]”, “Not sure :[” and “Interesting :o” reflects the level of incident observing.

3. RESULTS

3.1 Effectiveness of Intelligent coding tool

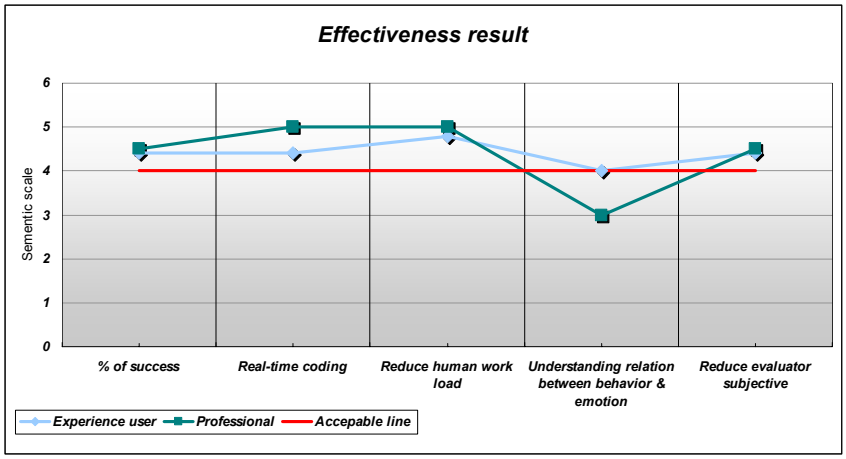


Figure 9. Effectiveness result chart

In general aspect, effectiveness means successfulness of usage. If goal is achieved, it implies the success. In contrast, the result is ineffective unless goal was achieved. Questionnaire about effectiveness was raised as the exploring issue.

3.2 Efficiency of Intelligent coding tool

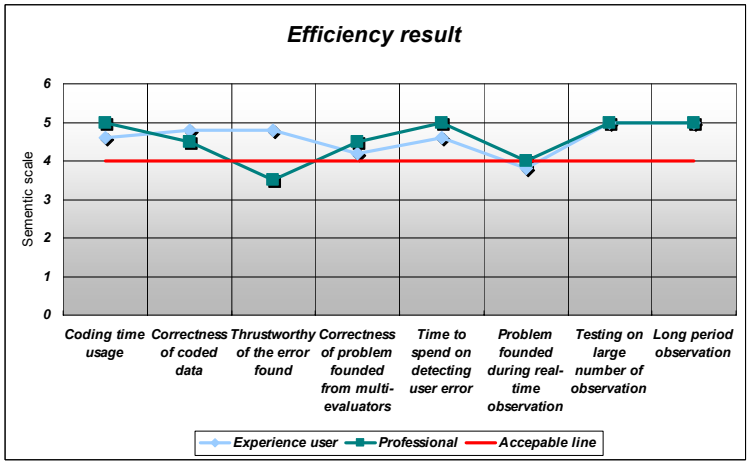


Figure 10. Efficiency result chart

Question in term of time usage, correctness, trustworthy covering to real-time problem detection ability was investigated through mainly questionnaire. Interview and observation were applied for the purpose of retrospective checking and user's insight exploration. The mean score less than 4 would be considered to be a topic of discussion.

3.3 Ease of use of intelligent coding tool

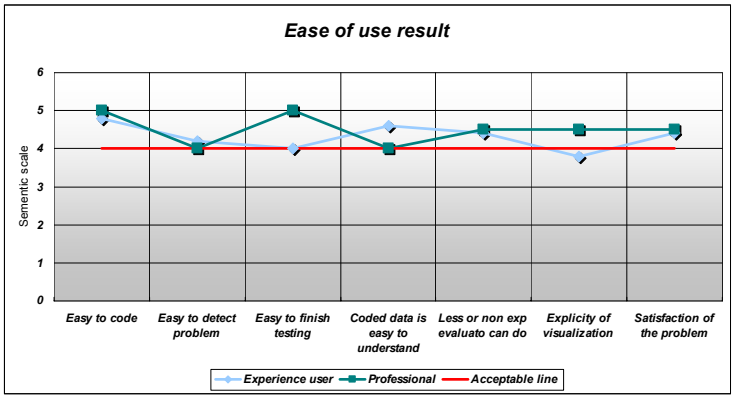


Figure 11. Ease of use result chart

Measurement of participants feeling in easiness of the tool was considered as a milestone in tool evaluation. Because ease of use affects on the inefficient of the tool and human performance along with satisfactory, evaluation were based on the same method as the effectiveness and efficiency by questionnaire. Not only interface was evaluated in this section, but overall system usage was also in consideration.

Table 5. Intelligent coding tool vs. Manual style (Noldus) feature comparison

Features	Intelligent coding tools	Manual style (refer to Noldus)
Modules		
Live coding	Tool assist usability testing to record data	None
Off-line coding	Multi-view Video with time-stamp replay	Multi-view Video with time-stamp replay
Video Module	record and off-line video replay synchronized by time-stamped	record and off-line video replay synchronized by time-stamped
Multiple Video Extension	Limited to 2 camera now.	Up to 4 cameras
External Data Module	Not offer now.	synchronized recording of data on an external data acquisition system
Screen Capture Module	By software	Through add-on product
Mobile Lab capability	Every module completely connected to a laptop	Every module completely connected to a laptop
Setting up experiments		
Record observations data	Behavior recorded by tool, facial expression by human in real-time or off-line	Both of behavior and facial expression recorded off-line by human
Coding interpretation	Represent in product function or language with user be able to program	Result in subject, behavior and modifier
Setting Variable	Input variable in program source code	Set by user in the software interface

Coding scheme		
Coding syntax	Function, task sequence, etc depend on user programming	Subject, behavior and modifier which need to define before
Anytime coding scheme editor	Now, need to edit in program source code	Able to edit scheme anytime, even while observing
Coding result	Word, sentence and time I table format for easy to find and analyze	Subject, behavior, modifier and time which able to analyze by statistic included in software
Describe set of behavior in human-language	Yes (depend on user programming)	Describe in 3 word of subject, behavior and modifier
Visualizing data		
See the recorded behavioral data, video in one plot	Yes	Yes
Visualize multiple observations at once to compare flows of events.	Yes, video represent incorporate with data table	Yes, video represent incorporate with data table
Problem identifier assist	Problem marker and color highlight	No.

3.4 Feature comparison

Intelligent coding system emphasizes how system communicates with the user. The representation of the recorded data is obviously different from the reference system such as Noldus (see table 5.). Reducing coding task feature such as live coding and problem identify assistant were purposed in the system and received good feedback from the tester both of by interviewing and questionnaire. However capability of the system is still limited by the programming flexibility. All result needs to be programmed in computer language before system starts. It may be a huge impediment for the tester who is not familiar with programming. The basic observation features such as video recording, time-video synchronization and statistic analysis capability of data exists both of our system and Noldus. Lastly, the intelligent coding tools perform a relevant feature in usability coding process experimented by many testers. Nevertheless, capability and expansion ability of the tool still needs to be developed more especially in task-sequence programming which needs to be more user-friendly process than the source code programming. An expert suggests that system should memorize the first trial of user to use as a coding scheme later.

3.5 Conclusion

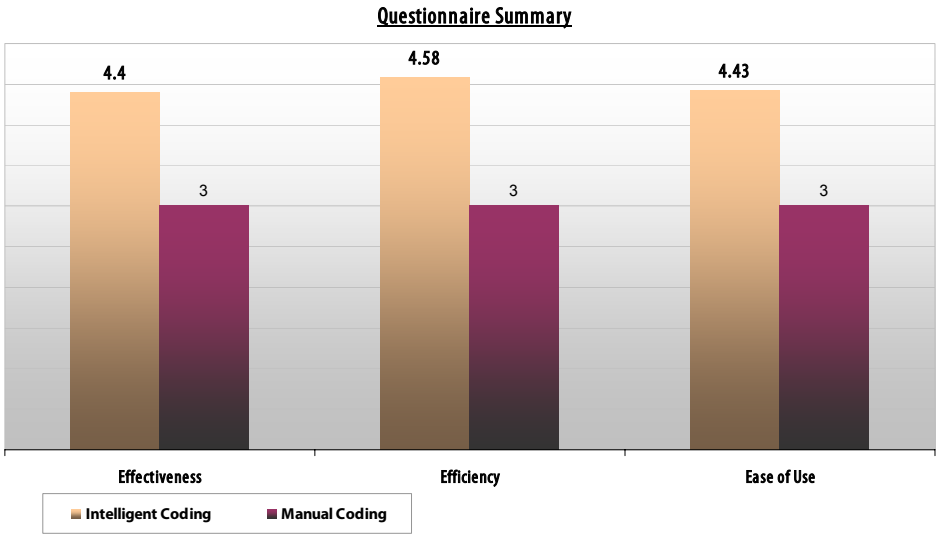


Figure 11. Questionnaire summary

The review from experts shows that there is a potential in ability to code data. Both of extenuating of coded data subjection and human work productivity were tended to be improved. This tool reported potential in assisting evaluator to do coding with high success rate than traditional manual coding. Automatic coding approaching on physical interface can reduce task of coding and time to perform the testing compare to the manual method. Supplement of simple, exact and correct data was provided by the tool; however error coding potential did not take advantage on automatic coding. Error coding still need evaluators to work on and require their skill to accomplish. In the ease of use perspective, participants can understand how to use this tool at the first time after introduction. Participants have positive feedback about the way to represent data and additional "error button". It looks simple to operate and difficult to misuse. A minor change of interface color should be developed. Moreover, several of error buttons need to be customized by evaluators to match the specific testing. In conclusion, automatic and real-time coding was operated well and stands in good position to apply to the usability testing especially for usability amateur.

4. REFERENCE

Arroyo, E. (2006). *Usability tool for analysis of web designs using mouse tracks*. MIT Media Lab, Cambridge

Granic, A. (2004). *Evaluation of Interaction Designing Web-Based Intelligent Tutoring Systems*. University of Split Nikole Tesle

Gaffney, G. (2000). *Usability technique – Information & Design*. http://www.infodesign.com.au

Keppel G. (1982). *Design and analysis: a researcher's handbook* 2nd edition. Prentice-Hall, Englewood Cliffs, New Jersy, United state.

Lillian M. & Frank B. Gilberth, (1927). *Time and Motion Study Developments.* American Machinist 66

Mayhew, Deborah J. (1999). *The Usability Engineering Lifecycle: A Practitioner Guide to User Interface Design.* Morgan Kaufman Publishers

Molich, R. (1998). Politics and Usability: Test your skills against the experts, Panel: *Proceedings of ACM CHI 2003 Conference of Human Factors in Computing Systems*.

Molich, R., Thomsen, (1999). *Comparative evaluation of usability tests.* Conference on Human Factors in Computing Systems (Pittsburgh, PA, May), pp. 83–86. New York, NY: ACM Press.

Nielsen, J. (1993). *Usability Engineering*. Boston, MA: Academic Press.

Nielsen, J., and Molich, R. (1990). *Heuristic evaluation of user interfaces.* *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.

Quantum research Inc. (2006). "*Qprox – QT113 datasheet*" http://www.qprox.com

Rademaker P. (1991). *Task analysis of an equipment assignment problem*. AI Memo, AI Lab Vrije Universiteit Brussel, Brussels Belgium

Rick. E. Robinson,(1994) "*Working with new media's cultural intermediaries*" Eshraghian Laboratories

Souza, Randy,(2002). *Best Practices for Usability Testing*. Forrester Report,

Wesley E. Woonson. (1981). *Human engineering standards and design handbooks*. McGraw-Hill Professional Press