

EE4213 Human-Computer Interaction

Week 13 Evaluation (Part B)

Review Exercise: For each of the following activities, identify if it corresponds to formative evaluation or summative evaluation. Briefly explain your answers.

(a) The project team identifies and fixes severe usability problems before the website goes live.

➔ *This is an example of **formative** evaluation because the evaluation occurs when the website is still in the development.*

(b) Before starting the process of creating a new version of CityU's AIMS, the project team evaluates the current version of the website, particularly looking for major usability issues.

➔ This is an example of **summative** evaluation because the item being tested is the current version of CityU's AIMS, which already exists.

(c) An animation team tries several motion capture systems and finally decides to use hand-drawn animation instead.

➔ This is also an example of **summative** evaluation because the motion capture systems being tried are products that are completed and available for sale. That is, the testing occurs after the completion of the products.

Evaluation Methods - Analytics

- Method for evaluating **user traffic** through a system or part of a system, e.g., systems involve selling a product or service can find out what customers do and want, which is important for improving the product or service
- Particularly useful for evaluating the usability of Website via logging user activity, counting and analysing data in order to understand what parts of the Website are being use and when
- Companies may develop their own analytics tool or use the services of companies which specialize in providing analytics and the analysis necessary to understand large volumes of data

Techniques for user testing

1. Usability engineering

- Levels of usability are specified **quantitatively**
- Criteria are specified for judging a product's usability
- Usability specification:
 - **Usability attribute/principle** – principle to test
 - **Measuring concept** – More concrete by describing the attribute in terms of the actual product
 - **Measuring method** – state how the attribute will be measured
 - **Now level** (value in existing system) / **worst case** (lowest acceptance value) / **planned level** (target for the design) / **best case** (best possible measurement)

Usability Engineering Example:

Test the usability of an electronic diary device

- **Attribute:** Guessability (defined by usability engineers)
- **Measuring concept:** Ease of first use of system without training
- **Measuring method:** Time to create first entry in diary
- **Now level:** 30 sec. on paper-based system
- **Worst level:** 1 min. (determined before test)
- **Planned level:** 45 sec. (determined before test)
- **Best case:** 30 sec. (determined before test)

If the averaged time (say, from 100 users) is 55 sec., it is usable, although the planned level is not met

Measurement methods can be determined from:

1. Time to complete a task
2. Percent of task completed
3. Percent of task completed per unit time
4. Ratio of successes to failures
5. Time spent in errors
6. Percent of number of errors
7. Percent or number of competitors better than it
8. Number of commands used
9. Frequency of help and documentation use
10. Percent of favorable/ unfavorable comments

Measurement methods can be determined from:

11. Number of repetitions of failed commands
12. Number of runs of successes and of failures
13. Number of times interface misleads the user
14. Number of good and bad features recalled by users
15. Number of available commands not invoked
16. Number of regressive behaviors
17. Number of users preferring your system
18. Number of times users need to work around a problem
19. Number of times the user is disrupted from a work task
20. Number of times user loses control of the system
21. Number of times user expresses frustration or satisfaction

2. Experiment

- Aim: Answer a question or to test a hypothesis chosen by evaluator
- A number of experimental conditions are considered which differ only in the value of some controlled variables
- Predict the relationship between two or more variables
- Quantitative measurements are collected
- Analysed and validated statistically & replicable.

Factors that are important to reliable experiment:

- **Participants**

- Representative
- Sufficient sample

- **Variables**

- Independent variable (IV) – manipulated (**controlled**) by the evaluator to produce different conditions, e.g., interface type, number of menu items
- Dependent variable (DV) – depends on IV and are **measured** in the experiment, e.g., time taken, number of errors

- **Hypothesis**

- Prediction of outcome in terms of IV and DV
- Alternative hypothesis: there is difference between conditions
- Null hypothesis: states no difference between conditions - aim is to disprove this

Allocation of participants

- **Within** groups design
 - All participants perform in all conditions
 - Transfer of learning is possible but less costly & less likely to suffer from user variation
- **Between** groups design
 - Each participant performs in one condition only
 - No transfer of learning but more users required & variation can bias results

e.g., assume 2 different conditions and 10 measurements are needed, within groups design requires 10 participants while between groups design requires 20

Example

- Alternative hypothesis: Users will remember the natural icons more easily than the abstract icon
- More easily: the speed at which a user can correctly select an icon
- Independent variables: 2 sets of icons
- Dependent variables: time & number of mistakes

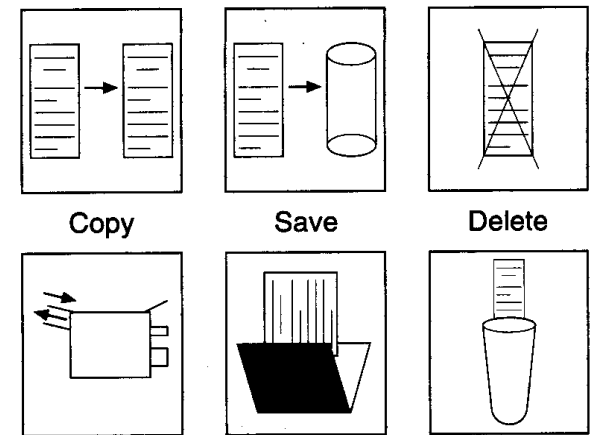


Figure 11.3 Abstract and concrete icons for file operations

Table 11.2 Example experimental results – completion times

		(1)	(2)	(3)	(4)	(5)
Subject number	Presentation order	Natural (s)	Abstract (s)	Subject mean	Natural (1)–(3)	Abstract (2)–(3)
1	AN	656	702	679	-23	23
2	AN	259	339	299	-40	40
3	AN	612	658	635	-23	23
4	AN	609	645	627	-18	18
5	AN	1049	1129	1089	-40	40
6	NA	1135	1179	1157	-22	22
7	NA	542	604	573	-31	31
8	NA	495	551	523	-28	28
9	NA	905	893	899	6	-6
10	NA	715	803	759	-44	44
mean (μ)		698	750	724	-26	26
s.d. (σ)		265	259	262	14	14
		s.e.d. 117		s.e. 4.55		
Student's <i>t</i>		0.32 (n.s.)		5.78 ($p < 1\%$, two tailed)		

Rough conclusion: natural icons require less time in average which means that alternative hypothesis is chosen

Techniques for user modelling

- Provide a way of evaluating products or designs without directly involving users via prediction
- “Less expensive” than user testing
- Usefulness limited to systems with predictable tasks, while they are difficult to apply in large-scale dialogs
- Based on expert error-free behavior, i.e., errors are not allowed in the execution
- Unpredictable factors such as individual differences among users, fatigue, mental workload, etc.

1. GOMS

- Proposed by Card, Moran and Newell in 1983
- Most well-known predictive modelling method in HCI
 - Stand for **goals, operators, methods, selection**
 - Goal: what the user wants to achieve (e.g., find a website about HCI design)
 - Operators: basic actions user performs to attain the goal (e.g., press keyboard key, click mouse)
 - Methods: learned procedures for accomplishing the goals (e.g., type "human computer interaction", press "search" button)
 - Selection: means of choosing between methods

GOMS Example:

Delete text in a paragraph using WORD

Goal: delete text in a paragraph

Menu-Option Method:

Step 1. Highlight text

Step 2. Execute "Cut" command in "Edit" menu

Delete-Key Method:

Step 1. Press "Delete" key to delete character one by one

Operators to use in above methods:

Click mouse

Drag cursor over text

Select menu

Move cursor

Press keyboard key

Selection of methods:

Rule 1: Use Menu-Option Method if large amount of text is to be deleted

Rule 2: Use Delete-Key Method if small amount of text is to be deleted

- Uses of GOMS:

- Provide measures of performance

(e.g., increase steps in methods → increase short term memory requirement)

- Provide suggestions for improving the design (e.g., old-version ATMs returned cards in the last step)

2. Keystroke level model (KLM)

- A very low-level GOMS model which can provide actual **numerical** predictions of user performance
- 7 execution phase operators:
 - Physical motor -
 - K** **k**eystroking, actually striking keys
 - B** **b** pressing a mouse **b**utton
 - P** **p**ointing a target
 - H** **h**oming, switching hand between mouse & keyboard
 - D** **d**rawing lines using the mouse
 - Mental
 - M** **m**entally preparing
 - System
 - R** **r**esponse (can be ignored)

Times are empirically determined

$$T_{\text{execute}} = T_K + T_B + T_P + T_H + T_D + T_M + T_R$$

Table 6.1 Times for various operators in the KLM (adapted from Card, Moran and Newell [37])

Operator	Remarks	Time (s)
K	Press key	
	good typist (90 wpm)	0.12
	poor typist (40 wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitts' law	$0.1 \log_2 (D/S + 0.5)$
	average movement	1.10
H	Home hands to and from keyboard	0.40
D	Drawing – domain dependent	–
M	Mentally prepare	1.35
R	Response from system – measure	–

KLM Example:

Delete the word “not” from the sentence
I do not like using keystroke level model

Assumptions:

- User’s hands at keyboard at the beginning
- User is a good typist

Which of the following methods is faster? Menu-Option Method or Delete-Key Method?

Menu-Option Method:

Mentally prepare	M	1.35
Switch to mouse	H	0.40
Move cursor to just before "not"	P	1.10
Hold mouse button down	B	0.10
Drag the mouse across "not" and one space	P	1.10
Release mouse button	B	0.10
Move cursor to "Edit"	P	1.10
Click mouse	2B	0.20
Move cursor to "Cut" option	P	1.10
Click mouse	2B	0.20

Hence the total execution time is 6.75 sec.

Delete-Key Method:

Mentally prepare	M	1.35
Switch to mouse	H	0.40
Move cursor to just before "not"	P	1.10
Click mouse	2B	0.20
Switch to keyboard	H	0.40
Press "Delete" (for "n")	K	0.12
Press "Delete" (for "o")	K	0.12
Press "Delete" (for "t")	K	0.12
Press "Delete" (for "space")	K	0.12

Hence the total execution time is 3.93 sec.

➔ Delete-Key Method is faster

3. Fitts' Law

- The law predicts that the **time to point at an object** using a device is a function of the distance from the target object & the object's size
- The further away and the smaller the object, the longer the time to locate it and point to it
- Useful for evaluating systems for which the time to locate an object is important such as handheld devices
- Other relevant applications include:
 - Predict expert text entry rates for several input methods on a 12-key cell phone keypad
 - Evaluate tilt as an input method for mobile devices with built-in accelerators