Evaluating and Comparing Interaction Styles

Georgios Christou and Robert J.K. Jacob

Department of Computer Science, Tufts University
161 College Ave, Medford, MA 02155
{christou | jacob} @cs.tufts.edu

Abstract. Research has created many new generation (post-WIMP) Interaction styles (IS) in the past years. From Ubiquitous to Affective Computing, researchers have not kept a uniform standard, so that Interaction Styles (IS) could be compared to one another. In this paper, we offer an approach that tries to put all IS on an equal footing. In this way one can compare different IS together, or even evaluate an IS by itself and find its weak and strong areas.

1 Introduction

In their classic article, Direct Manipulation Interfaces [1], Hutchins, Hollan and Norman discussed the cognitive basis of the then newly emerging WIMP interfaces. A lot has happened since then, though, and research has taken many different paths. A lot of different interaction styles have emerged, and are still emerging. Interaction styles (IS) [4] like Virtual Reality (VR) [7], Ubiquitous Computing (Ubicomp) [8], and Tangible User Interfaces (TUIs) [6] are a few of the examples at the forefront of research and development. These interaction styles try to liberate the user from the classical view of the desktop as the basic metaphor, and the keyboard and mouse as the basic means of interacting with the computer.

The problem is that these new interaction styles are very disparate. Research has not tried to keep common threads between them, so researchers don't have any common ground on which to view the whole landscape of IS. Not only there hasn't been any attempt to unify these different paths in Post-WIMP IS research, what amplifies this problem is that each IS has created its own ways in allowing the user to communicate with the computer, whereas in the WIMP world, there are only a few well known ways of interacting, like using the keyboard and mouse.

Because of the diversity and the apparent disparity between all these IS there is no way for a researcher to compare and evaluate them on an equal footing. This leads to many hours of user testing. Other times researchers may need to evaluate certain ways of doing things in an IS vs. another IS, but again there is no clear way of comparing two different IS. There is actually no clear and objective way to evaluate research decisions even.

So the problem becomes how to put all the IS on an equal footing and allow researchers and designers to make broad evaluations. What we need is a way to evaluate objectively an IS and see whether it fulfills the goals of why an IS should be built in the first place, namely allowing ease of use and ease of learning. This theory or framework should also allow for comparisons between IS. It should allow researchers to find good and bad points between IS and not only in evaluation of one IS. Finally, it should allow researchers to make more informed research decisions as to the directions their research should take.

In this paper we propose an approach that may give a solution to this problem.

2 "Ease" of Use

When looking at all these different IS there seems to be no real common threads on which to base a comparison or evaluation. The biggest commonality that all of the IS have is the user. Most of the IS are made for one purpose. Help the user first learn a system easily, and then become very efficient at accomplishing tasks, or Actions, with a system based on an IS. Even games and other entertainment software still try to make it easy for the user to learn how to use the interface, so that the user can start playing and get satisfaction from the product as soon as possible. Doing that, they try to reduce Hutchins, Hollan and Norman's "Gulf of Execution" [1], the gap between the user's intentions and the actions necessary to input them into the computer. So here is something that is common between them after all. The question then becomes how we can use this similarity in order to evaluate and compare IS.

Every system built that is based on any of the IS in existence, has a subset of allowable Actions that the user can perform. An Action is what the user does in order to accomplish a goal in the system, whereas a task is the goal that the user is trying to achieve by executing an Action, or a series of Actions. The user's first task then, is to learn how, and then perform those Actions that are allowable. In their paper, Kaptelinin et al. [3], present a checklist for evaluation of systems. In their checklist, under the heading of "Learning/Cognition/Articulation" they advise the designer to evaluate "Components of target actions that are to be internalized."[3]. They do not explain however, how the one should go about testing these components.

Keeping in mind that we do not want to compare user interfaces, but rather IS, an example of how we could use this is the following. I know how to turn my head in real life in order to see behind me (a common navigation action), so once I realize that I am able to do that in VR, I can just transfer my knowledge from the real world to VR, and not need to learn a new way of viewing the world in VR. We can thus test people to see how well the average population does in the "turning the head" Action, and use the results of the test as a score for it. Or we could score users on the use of the mouse, and get an average. In the same way, we can see how many things the user needs to know in order to perform an Action. For example, the user needs to know how to use a mouse, a keyboard and know the editing commands in order to use Microsoft Word efficiently. These are objective measures of a) how well a user would perform an Action and b) how many things the user draws from in order to learn how to perform the new Action. Table 1 shows a comparison example of Actions that users perform under different IS. We can evaluate all the Actions, giving them scores

	WIMP		VR		UbiComp	
	Action	**Score**	**Action**	**Score**	**Action**	**Score**
Navigational	1.Moving the mouse 2.Scrolling	X1 X2	1.Turn head 2.Walk	Y1 Y2	1.Turn Head 2. Walk	Z1 Z2
Editing	3.Use of keyboard	X3	3.Use of keyboard	Y3	3.Writing	Z3
Selecting	4.Clicking	X4	4.Grabbing	Y4	4.Grabbing	Z4
Total		Xtot		Ytot		Ztot

Table 1. Comparison of three different IS in three different types of Actions

X1, X2, etc. then take the totals Xtot, Ytot, and Ztot, divide them with the number of Actions that are included (for example in WIMP we would divide Xtot by 4) and compare them. We use the average rather than the sum to include the effect of the number of Actions that the user has to learn in our scoring strategy. The system that has the highest average wins, because we are looking for things that the average user does most expertly. Thus the high scores would show the highest expertise of users and lower scores show that users need to learn either more Actions or learn some of the Actions better in order to become proficient in the IS.

Each Action can be scored according to "how well one knows" to perform that Action. Another way of looking at this is "what must I know before I learn to perform this new thing?" The first way of evaluation can be used with experienced users, or even test knowledge transference. For example, if I know how to use the mouse in one application, I pretty much know how to use it in all applications. The second part of evaluation deals with new knowledge. If an Action is based on something familiar, then we may have knowledge transference which makes the Action easier to learn. One can create a test for every Action that is used in a system, or that is available by a particular interaction style. Thus, a user can take a test on an Action that is allowable by an IS. We can compare that to the average and we can tell how well the particular user knows how to perform the specific Action. Or we could see how far the user is from the average and see whether they are experts or beginners.

Another example might be to tally the score for every Action that the user can perform in a system, and see how proficient that user is in working with the system. In this way, Actions can be classified according to their "knowledge" factor. Thus we can see which Actions are already known by our users, which are just familiar, and which will need to be learned.

In the same way, we can compare different IS. We can compare an Action that is performed in one IS, with the same Action performed in a different IS. We can then compare the averages of users using the first versus the second, and see which IS allows for the more natural interpretation of the Action.

3 Conclusion

In this paper we have proposed a new approach to evaluating and comparing different interaction styles. We believe that this is crucial in order to be able to analyze IS and say objectively which are the strong and weak points of it. We have included some examples in order to show what we mean by the term Action, and how the Actions would fit in a comprehensive taxonomy. We have not included all categories of Actions in Table 1, for the sake of brevity. We also talked about Microsoft Word, but only in order to examine some Actions that fall under the Editing category, and not to evaluate the system or its interface. We are not interested in comparing systems, but rather Interaction Styles.

Another benefit of this analysis is that when analyzing a IS, one can see how easy or how hard to learn or to perform the allowable Actions are, and research can focus on making better devices or simplifying the “weaker” Actions. But most of all, it puts all the different IS on an equal footing, and allows comparisons that previously were just not possible to make. Thus it allows researchers to make objective decisions about new Actions that they create in an established IS, compare different IS together, or even how changing existing Actions impacts an IS.

There is a lot of work to be done though. We need to figure out at what level one should look at the Actions in an IS. Also, there are the issues of subjective satisfaction of users and of creating a comprehensive taxonomy of Actions that allows researchers to not only classify existing Actions, but also Actions that may be invented in the future. We are in the process of working on these issues, and we hope to present a more complete framework that helps the researchers make evaluations and research decisions, and that gives realistic comprehensive results.

References

1. Hutchins, E., Hollan, J., & Norman, D. Direct Manipulation Interfaces. In D. A. Norman & S. W. Draper (Eds.) User Centered System Design: New Perspectives in Human-Computer Interaction. Lawrence Erlbaum Associates: Hillsdale, NJ (1986).
2. Belotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., & Lopes, C. Making Sense of Sensing Systems: Five Questions for Designers and Researchers. Proceedings of CHI’02, ACM Press 415-422.
3. Kaptelinin, V., Nardi, B., and Macaulay, C., The Activity Checklist: A Tool for Representing the “Space” of Context. Interactions, July + August 1999, 27-39.
4. Shneiderman, B. Designing the User Interface: Strategies for Effective Human Computer Interaction, Addison Wesley Longman, Inc. 1998.
5. J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, Computer Graphics: Principles and Practice, Addison-Wesley, Reading, Mass.(1990).
6. Ishii, H. and Ullmer, B., "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms," Proceedings of Conference on Human Factors in Computing Systems (**CHI '97**), ACM, Atlanta, March 1997, pp. 234-241.
7. J.D. Foley, "Interfaces for Advanced Computing," Scientific American, Vol. 257(4) pp. 127-135 (October 1987).
8. M. Weiser, "The Computer for the Twenty-first Century," Scientific American, Vol. 265(3) pp. 94-104 (September 1991).