Reprint from Proceedings of the 1st International Conference on Universal Access in Human-Computer Interaction (5-10 Aug 2001, New Orleans, Lawrence Erlbaum)

Identifying Critical Interaction Scenarios for Innovative User Modeling

V. Katie Emery[1], Julie A. Jacko[1], Thitima Kongnakorn[1], Vipat Kuruchittham[1], Steven Landry[1], George McLeland Nickles III[1], Andrew Sears[2], Justin Whittle[1]

[1]School of Industrial & Systems Engineering
Georgia Institute of Technology
765 Ferst Drive
Atlanta, GA 30332-0205

[2]School of Information Systems
UMBC
1000 Hilltop Circle
Baltimore, MD

ABSTRACT

Usability testing typically focuses on methodology and metrics, while the specific interactions being tested are chosen in an ad hoc way. This paper demonstrates a framework for organizing interaction scenarios for graphical user interfaces (GUI). The framework is an adaptation of the two-dimensional abstraction hierarchy introduced by Rasmussen [1] in which an interaction consists of a purpose, functionality, and form. Interactions for a GUI are organized into four main categories, with numerous subtasks. The four main categories determined are 1) object manipulation, 2) content manipulation, 3) view manipulation, and 4) information presentation. The general framework can guide evaluators in choosing key interaction scenarios for GUI applications across a diverse array of user capabilities.

1.0 INTRODUCTION

Many tools exist to facilitate the evaluation of computer interfaces (e.g., cognitive walkthroughs, heuristic evaluations, user testing, etc.). Ideally these methods would investigate all possible interaction scenarios that might occur in a particular application between the user and the computer. However, this is not realistically feasible. As a result, evaluators often employ only small subsets of interactions or selected representative interactions. To our knowledge no research to date has provided a systematic method to determine what interactions should be investigated, or how interactions could be compared to other interactions within the same interface, or with interactions in other interfaces. A hierarchical framework that organizes potential interaction scenarios would provide support for choosing interactions to be tested, and allow comparison of interactions within and between applications.

One difficulty in creating such a framework is that an interface abstracts the user from the task they are trying to accomplish. The interface can be viewed as a "cognitive agent" on which the user must act to accomplish the desired goals [2]. The user's ultimate goal (e.g., producing a document, creating a chart, coordinating data, etc.) is not directly performed, but instead is composed of a sequence of low-level interactions with the interface. In this regard the "usability" of an interface is a combination of low-level actions (e.g. moving a mouse, pressing a key combination, etc.), the sequence of low-level interactions required to accomplish a goal, and the ability of the interface to satisfy the user's goal. These different aspects of the task reflect both physical and cognitive behavior from the user. Additionally, interactions can be initiated by the interface (e.g. providing feedback, alerting the user to system status, etc.), further complicating efforts to create a general framework of interactions.

For a high level goal, such as inserting a chart into a document, there are a number of lower level objectives that must be accomplished, typically by performing actions on objects through the functionality of the interface. This structure, characterized by purpose at the higher levels, functional aspects in the middle levels, and form at the lower levels, can be represented as an abstraction hierarchy. This type of hierarchy, applied to interactions in a graphical user interface (GUI) can elicit key interaction scenarios. In the succeeding sections, a description of the general framework is followed by an example of how it may be applied.

2.0 ABSTRACTION HIERARCHY

Rasmussen's abstraction hierarchy [3] serves as the foundation for the framework and possesses two dimensions, as shown in Figure 1. On the horizontal axis is decomposition, where moving to the right can

be considered "zooming in" on components of the system. On the vertical axis is the level of abstraction, which breaks down the task, from goals or purposes at the highest level down through the physical form of the interface at the other. Goals propagate down through the hierarchy, eventually affecting the physical functioning of the system, while the physical form can produce effects that propagate up the hierarchy, and can affect the purposes, either by confounding them, or creating new purposes.

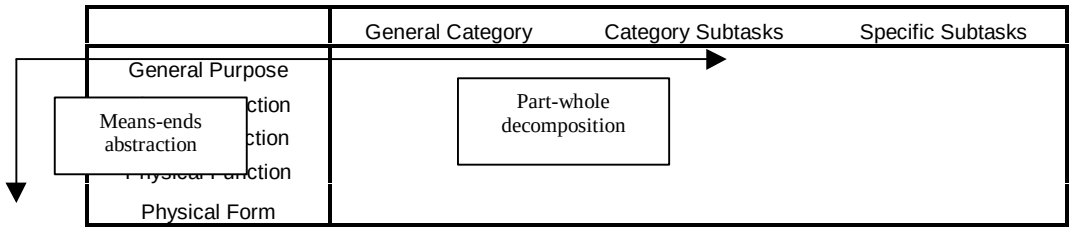

Figure 1. Abstraction hierarchy

It should be noted that this use of the abstraction hierarchy differs from its use by Rasmussen in ecological interface design (EID). In EID, the hierarchy is used to describe the process that is being controlled through the display, so that the display reflects the underlying functionality of the process. For its use in eliciting interaction scenarios, the process (e.g. creating a document, manipulating data) is not being shown. Instead, the abstraction diagrams the functionality of the interface.

The value of this structure is that, in segmenting an interaction into several levels of abstraction, the interaction can be compared with other interactions at each level. For example, within a completed framework, one could compare different physical actions that can accomplish the same purpose, or compare the ability of someone to accomplish the same purpose across different interfaces. The hierarchy also clearly indicates the tangible connections between purpose, function, and form.

Applying the abstraction hierarchy to interaction scenarios, the high level General Purpose is given by the particular goal of the user, and is not dependent upon the application. Each succeeding level answers the "how" that accomplishes the preceding level, in increasing detail. The Abstract Function is the high-level description of the method by which the General Purpose will be accomplished. The next level, General Function, describes how the Abstract Function is accomplished in terms of the functioning of the general class of applications in question. The Physical Function level describes how, in regards to the functionality of the particular interface, the General Function is accomplished. Finally, the Physical Form describes the items necessary to accomplish the Physical Function.

2.1 The general framework for graphical user interfaces. For a particular class of applications, such as graphical user interfaces (GUI), a portion of the framework can be completed, easing implementation of the framework for particular interfaces. In this regard a literature search for interaction scenarios was conducted for GUI research. The interactions described in the literature were catalogued and categorized into General Functions. These General Functions were then classified into the type of manipulation that was being accomplished, resulting in the level of Abstract Function. The results of the literature search and the resulting categorizations are shown in Table 1. The four Abstract Functions are 1) object manipulation, 2) content manipulation, 3) view manipulation, and 4) information presentation. Note that the level of General Purpose, and Physical Function and Form are removed, as these relate to specific objectives in the case of the former, and in specific interfaces for the latter. Short descriptions of the categories, including examples of objects to which they apply, are also given.

3.0 APPLICATION TO SPECIFIC INTERFACES

The first few hierarchies for specific interactions involved in the opening of a document in Microsoft Word 2000 is shown in Table 2. For clarity and brevity, the Physical Form appears in parentheses on the same level as Physical Function.

In this example the advantages of the hierarchical form can be seen. Lower level activities can be identified and measured (e.g., mouse movements, keyboard movements, etc.). The structure of the interaction shown by the various hierarchies and how the hierarchies are combined in many cases reflect the navigation required, both within the general function, and for the general purpose overall.

Each interaction at each level must be executed for the General Purpose to be accomplished successfully. The complexity of the interaction, and the eventual success of the interaction, is dependent not only on the underlying low-level interactions (e.g., ctrl+o, select "file" menu, etc), but also on the structure of the high level interactions and the feedback provided by the interface.

Table 1: Hierarchy Framework and References

	General category	**General subtasks**	**Specific subtasks**	**References**
Abstract function	Object manipulation			
General function		Open/Close (application, file, window …)		(Miah & Alty, 2000; Wixon, Williges, & Coleman, 1985)
		Add/Delete (file, application, graphic, …)		(Bordegoni, 1994)
		Change attributes (color, text properties, …)		(Wixon et al., 1985)
		Change position		(Bordegoni, 1994; Bowman, Johnson, & Hodges, 1999; Foley, Wallace, & Chan, 1984; Miah & Alty, 2000)
		Change orientation		(Bordegoni, 1994; Bowman et al., 1999; Foley et al., 1984; Gallimore & Brown, 1993)
		Change size		(Miah & Alty, 2000)
		Change membership (objects, files, …)		(Bowman et al., 1999)
		Activate/Deactivate (window, object, …)		(Bordegoni, 1994; Bowman et al., 1999; Foley et al., 1984; North & Shneiderman, 1997; Perlman, Green, & Wogalter, 1992)
		Export (print, send to audio channel, etc.)		(Shneiderman, 1998)

	General category	**General subtasks**	**Specific subtasks**	**References**
Abstract function	Content manipulation			
General function		Text entry		(Foley et al., 1984; Kishino & Hayashi, 1995; Perlman et al., 1992; Thimbleby, 1983; Wixon et al., 1985)
		Object creation		(Bordegoni, 1994; Foley et al., 1984)
		Selection from a set of continuous values		(Ackerman & Cianciolo, 1999)
		Selection from a discrete array		(Ackerman & Cianciolo, 1999; Buxton, 1988; Instance & Howarth, 1993; Perlman et al., 1992)
		Import (scan, tactile interface, sound, …)		(Brewster, Raty, & Kortekangas, 1996)

	General category	**General subtasks**	**Specific subtasks**	**References**
Abstract function	View manipulation			
General function		Scrolling in 2,3 dimensions		(Bederson & Meyer, 1998; Darken & Sibert, 1996; Kaptelinin, 1995; North & Shneiderman, 1997; Swierenga, 1990)
		Paging		(Paap, Noel, McDonald, & Roske-Hofstrand, 1987)
		Zooming		(Beard & John Q. Walker, 1990; Bederson & Meyer, 1998; Kommers, 1991; North & Shneiderman, 1997)
		Changing perspective		(Arsenault & Ware, 2000; Darken & Sibert, 1996)
		Channel control (i.e. direct input to particular instance, channel, etc.)		(Arnold, A. G., & Roe, R. A. 1989; Goldstein, J., & Roth, S. F. 1994)

	General category	**General subtasks**	**Specific subtasks**	**References**
Abstract function	Information presentation			
General function		Alerting (any of the senses)		(Ware, Bonner, Knight, & Cater, 1992; Wiener & Curry, 1980)
		Feedback (any of the senses)		(Arsenault & Ware, 2000; Bowman et al., 1999; Dennerlein, Martin, & Hasser, 2000)
		Attention directing (any of the senses)		(Ware et al., 1992)
		Unrequested information (balloon help, …)		(Freeman, 1994)
		Requested information (document, image...)		(Jacobson, Fusani, & Yan, 1993; Wixon et al., 1985)

3.1 Applications to User Needs. The failure of any key interaction in the task sequence, either due to the human or machine side of the system can potentially result in a failure to achieve the system goal(s). It is therefore important for the evaluator to consider the user in the development of the hierarchy. The evaluator needs to discover what assumptions each interaction assumes about the user's skills and abilities, and about the context of use. Those assumptions that are incorrect will result in an inability for the user to achieve her or his desired goal using that interface.

Table 2: Completed hierarchy for two tasks

	General category	**General subtasks**	**Specific subtasks**
General purpose	Open a file		
Abstract function	Object manipulation		
General function		Open a new document	
Physical function (Physical form)			Ctrl+o (keyboard) Select "file" menu, select "open" (mouse) "Open" button on toolbar (mouse)

	General category	**General subtasks**	**Specific subtasks**
General purpose	Open a file		
Abstract function	Information presentation		
General function		Visual feedback	
Physical function (Physical form)			Blank file opens (monitor)

	General category	**General subtasks**	**Specific subtasks**
General purpose	Open a file		
Abstract function	Content manipulation		
General function		Selection from a discrete array	
Physical function (Physical form)			Select file to open (mouse) Move to file to open (keyboard)

4.0 CONCLUSIONS

The hierarchy presented in this paper provides practitioners and researchers with a way in which to organize interaction scenarios to achieve more systematic interface evaluations. In addition, this organization scheme, which is grounded in Rasmussen's hierarchy, enables researchers to more quickly identify and attend to interaction scenarios that may pose particular challenges for people with limited capabilities due to perceptual, physical and/or cognitive impairments. By organizing and categorizing interaction scenarios in this manner, researchers and practitioners are better equipped to engage in accurate modeling of human performance for computer-based tasks. Accurate user modeling is particularly challenging for users with reduced capabilities due to perceptual, physical, and/or cognitive impairments because, to-date, researchers have not comprehensively characterized the influence of these types of impairments on computer-based task performance. The organization scheme introduced in this paper will facilitate modeling of this type by providing an organization scheme of interaction scenarios that should be investigated across diverse user groups. This paper serves as a launching point as the development of this scheme is ongoing.

References

1. Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). Cognitive Systems Engineering. New York: John Wiley & Sons, Inc.
2. Norman, D. A. (1991). "Cognitive artifacts". In J. M. Carroll (Ed.),Designing Interaction: Psychology at the Human-Computer Interface. NewYork: Cambridge University Press.
3. Rasmussen et al., 1994.
4. Miah, T., & Alty, J. L. (2000). Vanishing Windows Technique for Adaptive Window Management. *Interacting with Computers, 12*(4), 337-355.
5. Wixon, D. R., Williges, R. C., & Coleman, W. D. (1985). *Collecting Detailed User Evaluations of Software Interfaces.* Paper presented at the Proceedings of the Human Factors Soceity 29th Annual Meeting.
6. Bordegoni, M. (1994). Parallel Use of Hand Gestures and Force-Input Device for Interacting with 3D and Virtual Reality Environments. *International Journal of Human-Computer Interaction, 6*(4), 391-413.
7. Bowman, D. A., Johnson, D. B., & Hodges, L. F. (1999). *Testbed evaluation of virtual environment interaction techniques.* Paper presented at the Proceedings of the ACM symposium on Virtual Reality Software and Technology.
8. Foley, J. D., Wallace, V. L., & Chan, P. (1984). The Human Factors of Computer Graphics Interaction Techniques. *IEEE Computer Graphics and Applications, 4*(11), 13-48.
9. Gallimore, J. J., & Brown, M. E. (1993). Visualization of 3-D Computer-Aided Design Objects Articles. *International Journal of Human-Computer Interaction, 5*(4), 361-382.
10. North, C., & Shneiderman, B. (1997). *A Taxonomy of Multiple Window Coordinations*, [web page]. Available: ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/3854HTML/3854.html.
11. Perlman, G., Green, G. K., & Wogalter, M. S. (1992). *A Comparison of Direct-Manipulation, Selection, and Data-Entry Techniques for Reordering Fields in a Table.* Paper presented at the Proceedings of the Human Factors Society.
12. Shneiderman, B. (1998). Interaction Devices, *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (3rd ed., pp. 305-349). Reading, MA: Addison-Wesley.
13. Kishino, S., & Hayashi, Y. (1995). A Method for Estimating Code Key-In Times of College Students. *International Journal of Human-Computer Interaction, 7*(2), 123-134.
14. Thimbleby, H. (1983). Guidelines for 'Manipulative' Text Editing. *Behaviour and Information Technology, 2*(2), 127-161.
15. Ackerman, P., & Cianciolo, A. T. (1999). Psychomotor abilities via touch-panel testing: Measurement innovations, construct, and criterion validity. *Human Performance, 12*(3-4), 231-273.
16. Buxton, W. (1988). The Natural Language of Interaction: A Perspective on Non-Verbal Dialogues. *INFOR: Canadian Journal of Operations Research and Information Processing, 26*(4), 428-438.
17. Instance, H., & Howarth, P. (1993). *Performance Modeling and Investigation of User Performance using an Eyetracker as an input device.* Paper presented at the the Polish-English Meeting on Information Systems, Bialystok, Poland.
18. Brewster, S. A., Raty, V.-P., & Kortekangas, A. (1996). *Enhancing Scanning Input with Non-Speech Sounds.* Paper presented at the Second Annual ACM Conference on Assistive Technologies.
19. Bederson, B., & Meyer, J. (1998). Implementing a zooming user interface: Experience building Pad++. *Software - Practice and Experience, 28*(10), 1101-1135.
20. Darken, R. P., & Sibert, J. L. (1996). Navigating Large Virtual Spaces. *International Journal of Human-Computer Interaction, 8*(1), 49-71.
21. Kaptelinin, V. (1995). *A Comparison of Four Navigation Techniques in a 2D Browsing Task.* Paper presented at the Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems.
22. Swierenga, S. J. (1990). *Menuing and Scrolling as Alternative Information Access Techniques.* Paper presented at the Proceedings of the Human Factors Society 34th Annual Meeting.
23. Paap, K. R., Noel, R. W., McDonald, J. E., & Roske-Hofstrand, R. J. (1987). *Optimal Organizations Guided by Cognitive Networks and Verified by Eyemovement Analyses.* Paper presented at the Proceedings of IFIP INTERACT'87: Human-Computer Interaction.
24. Beard, D. V., & John Q. Walker, I. (1990). Navigational Techniques to Improve the Display of Large Two-Dimensional Spaces: The Visual Presentation of Information. *Behaviour and Information Technology, 9*(6), 451-466.
25. Kommers, P. A. M. (1991). *Virtual Structures in Hypermedia Resources Congress II: Design and Implementation of Interactive Systems.* Paper presented at the Proceedings of the Fourth International Conference on Human-Computer Interaction.
26. Arsenault, R., & Ware, C. (2000). *Eye-hand co-ordination with force feedback.* Paper presented at the Proceedings of the ACM Conference on Human Factors and Computing Systems, New York.
27. Arnold, A. G., & Roe, R. A. (1989). Action Facilitation; A Theoretical Concept and its Use in User Interface Design. Paper presented at the Proceedings of the Third International Conference on Human-Computer Interaction.
28. Goldstein, J., & Roth, S. F. (1994). Using Aggregation and Dynamic Queries for Exploring Large Data Sets. Paper presented at the Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems.
29. Ware, C., Bonner, J., Knight, W., & Cater, R. (1992). Moving Icons as a Human Interrupt. *International Journal of Human-Computer Interaction, 4*(4), 341-348.
30. Wiener, E. L., & Curry, R. E. (1980). Flight-Deck Automation: Promises and Problems. *Ergonomics, 23*(10), 995-1011.
31. Dennerlein, J. T., Martin, D. B., & Hasser, C. (2000). *Force-feedback improves performance for steering and combined steering-targeting tasks.* Paper presented at the Proceedings of the ACM Conference on Human Factors and Computing Systems, New York.
32. Freeman, D. (1994). *Object Help for GUIs.* Paper presented at the ACM Twelfth International Conference on Systems Documentation.
33. Jacobson, T. L., Fusani, D. S., & Yan, W. (1993). Q-Analysis of User-Database Interaction. *International Journal of Man-Machine Studies, 38*(5), 787-803.