View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

SIGHCI 2004 Proceedings

Special Interest Group on Human-Computer Interaction

2004

Learning, Performance, and Analysis Support for Complex Software Applications

Steven R. Haynes
Pennsylvania State University, shaynes@ist.psu.edu

Thomas George Kannampallil
Pennsylvania State University, thomasg@ist.psu.edu

Follow this and additional works at: http://aisel.aisnet.org/sighci2004

Recommended Citation

Haynes, Steven R. and Kannampallil, Thomas George, "Learning, Performance, and Analysis Support for Complex Software Applications" (2004). *SIGHCI 2004 Proceedings*. 15.
http://aisel.aisnet.org/sighci2004/15

This material is brought to you by the Special Interest Group on Human-Computer Interaction at AIS Electronic Library (AISeL). It has been accepted for inclusion in SIGHCI 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Learning, Performance, and Analysis Support for Complex Software Applications

Steven R. Haynes
School of Information Sciences and Technology
Pennsylvania State University,
shaynes@ist.psu.edu

Thomas George Kannampallil
School of Information Sciences and Technology
Pennsylvania State University,
thomasg@ist.psu.edu

ABSTRACT

We propose a three-part framework describing support tools for users of complex software applications such as enterprise resource planning and decision support systems. The model is motivated by the objectives of learning, performance, and analysis and is grounded in the theories of constructivism, pragmatism, and reflection respectively. This mapping is supported both by results of prior research and by a case study formative evaluation of a complex, cognitive support system developed for anti-terrorism resource allocation. The work contributes to the field of system usability by providing an integrative framework linking established theoretical positions with empirical research on human-computer interaction.

Keywords

Complex systems, user support, help, learning, reflection.

INTRODUCTION

This paper proposes an integrated framework characterizing user support for complex applications as consisting of learning to use, efficiently using, and effectively applying results from use. By complex applications we refer to systems such as enterprise resource planning (ERP) and decision support systems, which are used to help structure and solve ill-structured problems. By user support we refer to tools, content, and other materials such as training courses, help systems, manuals, tutorials, and so on, created to enhance a system's usability and usefulness.

The model we propose is derived from a set of practical objectives each of which is informed by integration of theory and empirical results from prior research. We provide preliminary results from a formative evaluation to show how the model can inform design activities. The model is descriptive in that it draws on theory and empirical support for how people approach and interact with complex software applications and is prescriptive in that it attempts to provide general design guidelines.

BACKGROUND

Mirel (1998) defines the tasks that give rise to complex systems development as characterized by the following core attributes:

- Indeterminacy of both task goals and criteria for task completion.
- Requiring higher order cognitive skill and integrating knowledge from different areas.
- Requiring advanced learning and instruction for effective performance.

Approaches to understanding human-computer interaction for systems supporting these tasks are correspondingly complex. Becoming proficient with complex software applications presents a learning challenge for their users. Performance support systems (PSS) is a largely practice-driven approach designed to enhance user performance with user interface and support environment features that account for user information needs (Bezanson, 1995). While PSS are targeted directly at complex productivity software systems such as those central to the work reported here, they do not address the objectives of learning and analysis that bracket task performance, nor has their development so far been grounded in related theoretical positions and empirical findings.

STUDY, DOMAIN, SETTING, AND METHOD

Since spring of 2002 we have been working with the U.S. Marine Corps on a methodology, decision model and cognitive support system to be used by installation commanders and their staffs when making resource allocation decisions for anti-terrorism/force protection (ATFP) mitigation projects. The domain presents a number of critical challenges to existing resource allocation approaches including the need to account for a range of social, psychological, and technical factors. The model consists of three major components: facility prioritization, mitigation project utility, and optimization of resource allocation. These three along with supporting services (e.g., security, management of standing data, etc.) have been implemented as a distributed, web service-based cognitive support system.

The ATFP model and cognitive support system were evaluated using informal design reviews, focus group design reviews, and one-on-one guided walkthroughs. Design reviews were used to clarify and refine understanding of the domain and to develop strategies for providing cognitive support. Once a working prototype of the system was complete, we began usability studies through guided walkthroughs with users in the field.

The guided walkthrough method is an evaluation technique designed to investigate the usability and comprehensibility of a system early in the development process. It combines the theoretical basis of the cognitive walkthrough (CW) method (Polson, Lewis, Rieman, & Wharton, 1992) with a focus on actual users as informants. The CW method is based on the theory of exploratory learning that describes how users form goals, explore the actions available to them to make progress towards their goals, and continually assess whether the actions they take lead towards achieving identified goals.

Walkthroughs were conducted with 29 prospective system users at six different sites. Walkthrough participants included public works officers, provost marshal officers (military police), installation anti-terrorism officers, and civilian facility planners. Walkthroughs lasted from 45 minutes to two hours and involved working through a use scenario with the ATFP system. Walkthrough participants were asked to comment on what they were seeing and on their reaction to the information and user interface controls on the page. Particular attention was given to elements of the system that were unfamiliar, confusing, or otherwise at odds with their expectations.

RESPONSE TO FORMATIVE EVALUATIONS

We identified three key support requirements characterizing users' interactions with the ATFP cognitive support system: *learning*, *performance*, and *analysis*. By learning we refer to the cognitive activities of users as they approach a complex system and attempt to make sense of it. Performance refers to task performance, working with the system to complete some sub-task related to completion of a higher-level objective. By analysis we refer to the activities involved in understanding the information derived from system use, and how this information applies to the task objective. Like all models and frameworks, ours is an abstraction and necessarily obscures the true complexity and inter-relatedness of these three core support requirements.

Three key theories contribute to characterizing and understanding the cognitive priorities of users in each of the three phases described by the model. These theories and their correspondence with system use objectives appear in figure 1 below and in the following sections we justify this mapping.

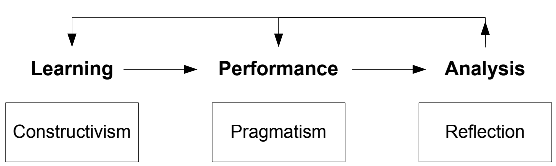

Figure 1 - The Learning-Performance-Analysis Framework

Learning

People actively construct their understanding of the environment, other people and social systems, and the tools they use to do work. One approach to development of systems with high learning requirements is learner centered design (LCD) (Soloway, Guzdial, & Hay, 1994). Approaches to learner-centric design are inherently constructivist in that they explicitly acknowledge the process of active cognitive organization and reorganization of new knowledge.

Mental models are considered the basic mechanism with which people actively construct their understanding. Mental models are among the most basic constructs supporting cognition (Johnson-Laird, 1983) and the manipulation of mental constructs has been described as an effective means for creating a bridge between task goals and the tools they use to achieve them (Gentner & Stevens, 1983). Baecker (1995) argues that users employ mental models in their attempts to understand and predict information system behavior, and their success in using a system depends on how well their mental model corresponds to the model represented in the system design. Where mental models are incomplete or incorrect users experience frustration and that may inhibit them from exploring and utilizing a system's full potential.

Research suggests that tool users with better conceptual models perform better and are better equipped to be innovative in their system use (Halasz & Moran, 1983; Borgman, 1999; Fein, 1993; Lewis, 1986; Rosson, Carroll, & Bellamy, 1990; Norman, 1983). Borgman (1999) provides evidence that users provided with a conceptual model perform tasks related to documents retrieval better than those who are provided with only operational, task-oriented methods. Halasz and Moran's (1983) experiments with students learning to use a stack calculator also provides support for the role of conceptual models in tools performance and innovative, strategic thinking.

Scaffolding has been proposed as the process of assembling instructional content to evolve understanding of a system. Norman (1983) argues that by understanding users' existing mental models, and by developing techniques to help them scaffold more accurate representations of system structure and behavior, designers can assist users as they construct understanding of the new systems they encounter. There is widespread agreement that prior knowledge influences learning, and that learners construct concepts using prior knowledge (Resnick, 1983). System training tools and materials should strive to impart more accurate mental models as these lead to increased learning performance, positive attitudes toward using the system, and the ability to apply learning to new systems in new domains (Sein, Bostrom, & Olfman, l993).

Learning ATFP

The ATFP consists of three major decision component modules that interact to perform resource allocation: facility prioritization, determination of mitigation project utility, and optimal resource allocation. We found that most study and design review participants were able to comprehend the functionality of the system when provided with a simple diagram of the three modules, the essential data required for each, and the interactions between the different modules. These representations are provided on a set of very high level, largely visual and conceptual training pages designed to impart only the most essential information about the system, a conceptual scaffold, including its component structure, the key data required for each component, and the interactions between the different components. The objective of this conceptual training is to provide a scaffold for construction of an accurate representation of the system's structure, functionality, and applicability to their domain.

Performance

Complex software applications that realize support for complex tasks are demanding of users' time and cognitive effort. Implicit in these complex design efforts is that these systems provide a positive cost-benefit advantage to the implementing organization. Ensuring adoption as users learn a new and complex software tool requires decreasing the cognitive demands of supported tasks.

First proposed by Charles Saunders Pierce (Pierce, 1954), pragmatism holds that the truth and meaning of a concept is determined by its utility and veridicality in practical contexts. In other words, when faced with a problem and objective, only the most relevant features of the problem space are considered. Central to the notion of pragmatism is effect; the ontology of causes, mediating factors, and results from actions that bear on solving the problem. The derivative pragmatic theory of explanation frames information use as "an interest-relative notion" in which information is useful or not only from a particular aspect or within a particular context (van Fraassen, 1991).

Pragmatism has been applied to performance with complex technologies in what Carroll and Rosson (1987) describe as the "production paradox," a principle of cognition according to which people are driven to produce direct, meaningful results from their work. The paradox is that users of complex tools are too busy managing their workload to spend time learning more about the tools they use, even though taking this time out would make them more productive in the long term. Most users of complex tools never become expert, they level off as intermediate users without learning advanced features and shortcuts designed to make use more efficient and effective. To overcome the paradox involves integrating learning with task execution directed at some meaningful objective.

Examples of the production paradox in action abound. Rieman (1996) describes a field study of computer users in everyday working situations where he found their exploratory learning limited by constraints of time and task goals. Neerincx and de Greef (1993) evaluated the help system of the statistical software package SPSS, and showed that integrated help did not improve the performance of the users because both their volume and content did not match user goal requirements.

Minimalism is a design response to pragmatism and the production paradox based on the idea that users see documentation as an impediment to getting work done (Carroll, 1990). The goal of the documentation developer is to minimize these barriers while providing the essential information required to use the system. The four principles of minimalism documentation design are its action-oriented nature, optimal (generally minimal) use of text, support of error recognition and recovery, and modularity (Carroll, 1990; Van der Meij, 1992). According to minimalism, users actively construct their mental model of a system by continually generating, testing and evaluating hypotheses related to its structure and behavior (Van der Meij, 1992).

Performance with ATFP

Several tenets from pragmatism inform the design of ATFP performance support tools. These are centered on the different types of content provided in the help facility, which delivers a page of content for each of the different functional pages in the system. Content is organized into sections corresponding to different types of explanation as appropriate to their information requirements. An example explanation page is provided in the figure below.

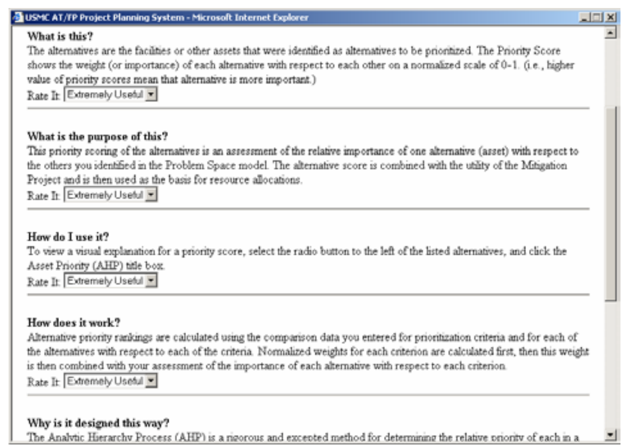

Figure 2 - Example ATFP System Minimalist Help Content

Help content is minimalist in that it attempts to impart only the most essential, goal-oriented explanations to system users. By separating content into different explanation types users can quickly focus in on the explanation that meets their immediate requirements.

Analysis

Complex software systems typically include results analysis as one of their supported user activities. Results analysis is a distinct activity with its own requirements for user support, such as visualization or complex results and their justification. In consequential domains, usability is closely related to the comprehension and trust that derive from the transparency of system results. Schön (1983) argues that many difficulties in professional work are a result of the disjunction between highly specialized but narrow training, and the complex, unique, and socially embedded "messiness" of the problems professionals encounter in practice. Two phenomena are central to our ability to deal with these difficulties, knowing-in-action and reflection-in-action. According to Schön, an expert professional's normal operating mode is based on a pragmatic *knowing-in-action* where:

"Our knowing is ordinarily tacit, implicit in our patterns of action and in our feel for the stuff with which we are dealing. It seems right to say that our knowing is in our action." (Schön, 1983, p. 49)

However, situations such as encountering a particularly difficult problem, or experiencing surprise at a particular outcome, result in a change of mode from knowing-in-action to reflection-in-action. In these situations, support for analysis become the dominant user requirement:

"As he tries to make sense of it, he also reflects on the understandings which have been implicit in his action, understandings which he surfaces, criticizes, restructures, and embodies in further action." (Schön, 1983, p. 50)

The priorities that characterize reflection make it a counter-point to the production mode that characterizes task performance with complex systems. In reflection, understanding is the objective rather than the more immediate goals motivating completion of an operation.

Schon (1983) describes an action-breakdown-reflection cycle that underlies professional practice. Breakdowns in otherwise unconscious, situated activity present opportunities for learning and for the active construction of new knowledge (Winograd & Flores, 1986). Breakdowns occur either in cognition or in task execution and can be either destructive, when they result in frustration, or useful when leading to better understanding and user performance. Recovery from breakdowns involves reflecting on problem fragments, making connections between fragments as the user constructs scaffolded understanding, and creating new abstractions, as mental models, to better describe what is happening in a given situation (Cox & Greenberg, 2000).

Explanation and justification are reflective responses to breakdowns in the use of complex systems. Theories of explanation provide the micro-structure for support content by relating system structure, behavior, and results to the range of factors that both guide and constrain how a system is designed and built to operate in a domain (Haynes, 2000). In decision and other cognitive support systems, sensitivity analysis is an important facilitator of reflection, since it helps identify the contributions of individual inputs to results and outcomes (Saltelli, 2000). Graphical sensitivity analysis methods involve information visualizations through representation in graphs, charts, or other data displays. Visual cues to the working of complex systems is acknowledged as an important aid to reflection, comprehension, problem solving, and decision making (Tufte, 1997).

Analysis Support in ATFP

An issue that emerged in both group design reviews and individual walkthroughs was the importance of explaining and justifying how and why the system arrived at a particular set of recommendations. A breakdown occurs when a user is confronted with a significant system result with consequences requiring reflection-on-action. Justification plays a crucial role in the interaction between a user and a decision system as it supports user attempts to determine if a result is correct and reliable.

In the ATFP system, sensitivity analysis is used to help users understand how their inputs on model setup pages affected final results and recommendations. Users interrogate the results obtained with the input they have given, evaluate the correctness of prior input with respect to the results obtained, and consider how changes to the input would affect the final results. The figure below shows one of the ATFP system's visualization dashboards showing how prioritization criteria weight scores were derived from pair-wise comparisons of criteria dyads.

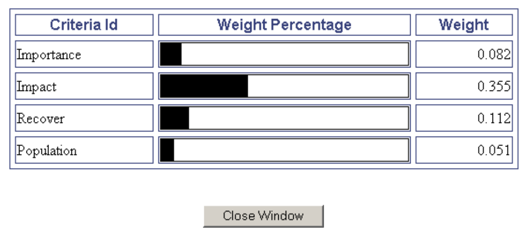

Figure 3 - ATFP Sensitivity Dashboard

In addition to sensitivity analysis and other dashboards, the ATFP help facility also provides explanation content on the topics *How does it Work?* and *Why Is It Designed This Way?* (see Figure 2) supporting users who want to delve into the details of system operations and design.

CONCLUSION

We have described a conceptual model for learning, performance, and analysis user support in complex systems. The integration realized in the model incorporates theories of constructivist learning, pragmatic performance support, and reflective analysis as a lens through which to view complex systems use and usability.

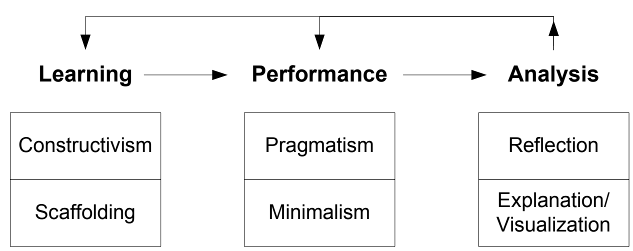

Figure 3. The Learning, Performance, Analysis Model

Our assessment of the appropriateness of these theories is grounded in research on human-computer interaction including mental models, scaffolding, minimalism and the production paradox, and explanation and visualization impact users' experiences with software tools. This work contributes to cognitive and design research in information systems usability by providing a theoretically and empirically grounded model for user support.

ACKNOWLEDGMENTS

This research was supported by a grant from the United States Marine Corps through the Marine Corps Research University.

REFERENCES

1. Baecker, R. (1995) Design & Evaluation Readings in Human Computer Interaction: Toward the Year 2000, Morgan-Kauffman, San Francisco.
2. Bezanson, W.R. (1995) Performance support: online, integrated documentation and training, *Proceedings of ACM Special Interest Group on Design of Communication (SIGDOC) ACM SIGDOC-95*, Savannah, Ga.
3. Borgman,C.L. (1999) The user's mental model of an information retrieval system: an experiment on a prototype online catalog, *International Journal of Human Computer Studies,* 51, 2, 435-452.
4. Carroll, J. M. and Rosson, M. B. (1987) Paradox of the Active User. In J. M. Carroll (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge, MA: MIT Press, 80-111.
5. Carroll, J.M. (1990) The Nurnberg Funnel, MIT Press, Cambridge, MA.
6. Cox, D and Greenberg, S. (2000) Supporting collaborative interpretation in distributed Groupware. *CSCW* , 289-298
7. Fein,R. M., Olson, G. M., and Olson, J.S. (1993) A mental model can help with learning to operate a complex device, *Proceedings of the INTERCHI '93 conference on Human factors in computing systems*, Amsterdam, The Netherlands, 157-158.
8. Gentner, D. and Stevens, A. (1983) Mental Models, Erlbaum, Hillsdale, NJ.
9. Halasz, F. G. and Moran, T. P. (1983) Mental Models and Problems Solving in Using a Calculator, *Proceedings of CHI'83 Human Factors in Computing Systems*, Boston, Ma, 212-216.
10. Haynes, S. R. (2000) Explanation in Information Systems: Can Philosophy Help? *Proceedings of the 8th European Conference on Information Systems,* Vienna, Austria, 31-38.
11. Johnson-Laird, P.N. (1983) Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness, Harvard University Press, Cambridge, MA.
12. Lewis, C. (1986) A model of mental model construction, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Boston, Ma, 306-313.
13. Mirel, B. (1998) Minimalism for complex tasks, In J. M. Carroll (Ed.), Minimalism beyond the Nurnberg funnel, MIT Press, Cambridge, MA.
14. Neerincx, M, and de Greef, P. (1993) How to aid non-experts, *Proceedings of the INTERCHI '93 conference on Human factors in computing systems*, Amsterdam, The Netherlands, 165-171.
15. Norman, D. A. (1983) Some observations on mental models In D. Gentner & A. Stevens (Eds.), Mental models, Erlbaum, Hillsdale, NJ.
16. Pierce, C. S. (1954) The Essentials of Pragmatism, J. Buchler, (Ed.), Philosophical Writings of Pierce ,New York: Dover, 251-268.
17. Polson, P.G., Lewis, C., Rieman, J., and Wharton, C. (1992) Cognitive walkthroughs: A method for theory- based evaluation of user interfaces. *International Journal of Man-Machine Studies 36,* 741-773.
18. Resnick, L.B. (1983) Mathematics and science learning: A new conception, *Science*, 220, 477- 478.
19. Rosson, M. B. Carroll, J. M. and Bellamy, R. K. E.,(1990) SmallTalk Scaffolding: A Case Study of Minimalist Instruction. *Proceedings of the SIGCHI conference on human factors in computing systems,* Seattle, Wa.
20. Saltelli, A., Chan, K., and Scott, E. M. (2000) Sensitivity Analysis, John Wiley & Sons.
21. Schön, D. A. (1983) The reflective practitioner: How professionals think in action, Basic Book, New York.
22. Sein, M.K., Bostrom, R., and Olfman, L. (1987) Training end users to compute: cognitive, motivational and social issues. *INFORM*, *3*, 236-255.
23. Soloway, E., Guzdial, M., and Hay, K. E. (1994) Learner-Centered Design: The Challenge for HCI in the 21st Century, *Interactions*, 1, 2, 36-48.
24. Tufte, E. R. (1997) Visual explanations: images and quantities, evidence and narrative, Graphics Press, Cheshire, CT.
25. Van der Meij,H. (1992) A critical assessment of the minimalist approach to documentation, *Proceedings of the 10th annual international conference on Systems documentation,* Ottawa, Canada 7-17.
26. van Fraassen, B. (1991) The Pragmatics of Explanation. In R. Boyd, P. Gasper, & J. D. Trout (Eds.), The Philosophy of Science, MIT Press, Cambridge, MA, 317-327.
27. Winograd, T. and Flores, F. (1986) *Understanding Computers and Cognition: A New Foundation for Design,* Addison-Wesley, Menlo Park, CA.