Concept Maps: Theory, Methodology, Technology
Proc. of the Second Int. Conference on Concept Mapping
San José, Costa Rica, 2006

USING CONCEPT MAPS TO INTEGRATE RESULTS FROM A COGNITIVE TASK ANALYSIS OF SYSTEM DEVELOPMENT

Robert R. Hoffman, Florida Institute for Human & Machine Cognition, Pensacola, FL, USA
Kelly Neville & Jennifer Fowlkes, CHI Systems, Inc., Orlando FL, USA
Email: rhoffman@ihmc.us, www.ihmc.us

Abstract A case can be made that the current US system for procuring large-scale information technologies is partly responsible for resulting in systems that are user-hostile, when they should be human-centered (Hoffman & Elm, 2006). Systems for both training and decision aiding cause automation surprises and create a need for kluges and work-arounds. A cognitive task analysis of systems development was conducted in which a number of highly experienced program developers and managers were interviewed, to identify factors contributing to current high rates of systems development failure and to determine ways in which the principles and methods of cognitive systems engineering might be integrated into systems development. Results of the analysis were depicted in Concept Maps that provide a broad overview of procurement. Concept Maps are used to identify and characterize system development activities, challenges, and strategies. The results affirm the claim that current systems development frameworks and methods are incompatible with known principles of human-centered computing and cognitive systems engineering. Looking ahead, the Concept Maps will be used as a springboard for creating new approaches to procurement, including the representation of system requirements, and the development of new systems to support development teams in achieving and maintaining common ground and addressing challenges of creating human-centered systems for complex cognitive domains.

1 Introduction

The failure rate among software development projects is notoriously high. The Wall Street Journal has reported that 50% of software projects fail to meet expectations and 42% of corporate information technology projects are discontinued before completion (cited by Coyle, 1999). A 1995 US Department of Defense study estimated that 46% of the system development efforts it funds result in products that are not successfully used and 29% never even produce a product (cited by Leishman & Cook, 2002). These statistics translate into practitioners who lose out because they don’t have the systems they need to perform their work effectively, not to mention billions of dollars squandered. For example, the US Internal Revenue Service spent $4 billion dollars on computer systems that, in the words of an IRS official, "do not work in the real world” (Marketplace, January 31, 1997). The U.S. Federal Bureau of Investigation spent $170 million dollars on a problem-riddled software development effort before abandoning it (Eggen, June 6, 2005). In general, large-scale information processing (or “intelligent”) systems, for both training and decision aiding, cause automation surprises and create a need for kluges and work-arounds (Koopman & Hoffman, 2003). This means that even those system development efforts that are successfully implemented tend not to support their users. Research suggests a myriad of complicated methodological, cultural, organizational, managerial, and technological factors are to blame (e.g., Davenport, 1998; Freeman, 2001; Nah, Lau, & Kuang, 2001; Whittaker, 1999).

Our preliminary discussions with systems developers suggested that Spiral and Waterfall modeling (and their variants) are often used to frame system development efforts, but subsequently end up as mere cover stories to describe the system development process. What people say they do often is not what they actually do. In addition, we came to suspect that “user involvement” in the design process is usually too little and too late. Worse still, the kinds of evaluations and verifications that are typically involved are based on a "satisficing" criterion—that is, users work with the new system for a while and then are queried concerning their opinions (completing what some system developers call "smile sheets"), resulting in evidence that some people like it, more or less, at least some of the time. Thus, we perceived an opportunity to pursue the implications of Human-Centered Computing for the procurement process, in particular the fact that requirements specification is not regarded as a process. "Requirements creep" is seen as a negative thing, a thing to be avoided, when an empirical looks suggests that it is an inevitability (Hoffman & Elm, 2006). An example of the sometimes stark disconnect between the procurement process and human-centering shows clearly in one statement in a US Department of Defense procedural guide (DoD, 1996): "Design efforts shall minimize or eliminate system characteristics that require excessive cognitive, physical, or sensory skills (Para. C5.2.3.5.9.1)." In effect, this translates to a built-in push to create systems that can actually prevent the achievement and exercise of expertise.

To follow-up on the disparity, we found an opportunity to take an empirical approach and have a detailed look at what really happens in the procurement of large-scale information processing and decision-aiding systems. We pursued our initial findings by conducting a cognitive task analysis to seek a more detailed insight into the reasons for the high rate of systems development failure. In particular, the analysis was conducted to identify factors contributing to the failure rate and ways in which cognitive systems engineering principles, methods, and tools might contribute to reducing the rate. The cognitive task analysis involved an assessment of the systems development literature and the elicitation and assessment of the knowledge and beliefs of systems engineers. The results were used to create a number of Concept Maps using CmapTools (Cañas et al., 2004) that will be used to guide an effort at creating new approaches and methods for procurement.

2 Cognitive Task Analysis

We conducted a documentation analysis, of about 50 sources. The sources included widely cited books such as *The Mythical Man-Month* (Brooks, 1995) and *Rapid Development* (McConnell, 1996), government acquisition and systems engineering publications such as the *DoD Architecture Framework* (DoD, 2003), website content describing systems engineering frameworks and methods (e.g., the Systems Modeling Language [SySML] and the Capability Maturity Model® Integration [CMMI®]) process, and journal articles and technical reports addressing systems development and acquisition methods and tools, challenges, principles, and practices. Examples are titles such as *Common Errors in Large Software Development Projects* (Gaitros, 2004) and *Weaving Together Requirements and Architecture* (Nuseibah, 2001). Our documentation analysis highlighted some limitations in systems development, that is, key contributors to the failure rates described above. Each category of limitation can be expressed as a major systems development challenge that currently is not being adequately addressed:

- Identifying and meeting user work support needs,
- Accommodating new information and requirements across the system development effort and system lifespan,
- Maintaining team coordination, and
- Coping with the complexity involved in large-scale procurements.

We conducted in-depth interviews with six individuals having at least 20 years of experience in software engineering, including experience either as managers of large-scale procurement projects, leaders of teams of software engineers, or systems engineers in large-scale procurement projects.

Five participants were interviewed (taking about 2-hr per session). The sixth participant participated in two 2-hr sessions. The interview protocol was based on the Critical Decision Method (CDM; e.g., Hoffman, Crandall, & Shadbolt, 1998) and the Standard Operating Procedure knowledge elicitation method (Hoffman, Ford, & Coffey, 2000). Both procedures support the participant in the recall of a past challenging or difficult systems development project, and included probe questions for eliciting more detail about specific systems development practices—for example, the conditions under which a practice occurs, the expertise required, and the value and true purpose. The cases that were discussed included, for example, the procurement of a decision support system for aircrews, the upgrade of a legacy training system, and the development of interfaces for an electronic library.

Each interview transcript was broken down into a series of ‘transcript chunks,’ where each chunk explained a single concept or made a particular argument. Each transcript chunk was assessed to determine the system development activities to which it related, and was coded in terms of types of system development activity (e.g., “risk management,” “coordinate with customer”), and challenges to system development (e.g., ”shifting requirements,” “team coordination and conflict management”). These data were represented in two ways. One was in the form of timelines having the purpose of showing when in the procurement and system development process activities are performed and when some new form of support tool might be useful to program stakeholders. For example, interview data suggested that significant effort was required to foster interactions among development sub-teams, among different contractors working on the same system, and between the team and the sponsor. However, these types of coordination activities are largely neglected by current frameworks and by the support tools that are used in systems development.

We also represented the results in the form of Concept Maps. The purposes of these were to paint a broad picture of the procurement/system development process, to capture some of the strategies that effective managers used to cope with the challenges, and to lay out the relations of all of the various stakeholders who need to maintain common ground during system development activity. In addition to the challenges, the themes identified across interviews included strategies for coping with the challenges of evolving requirements and the challenges of team coordination. With few exceptions (the use of whiteboard meetings and conversation), strategies were applied relatively inconsistently across development efforts, and seem to have varied considerably with respect to their effectiveness.

3 Example Concept Maps

Concept Maps we have developed capture findings of the cognitive task analysis and our ideas about how to use human-centering principles to address systems development and procurement deficiencies identified in that analysis. The first example Concept Map (Figure 1) presents a broad view of the shortfalls in current system development processes. This Concept Map integrates information from the literature review and the interviews. Figure 2 exemplifies a set of Cmaps that describe solutions to current systems development challenges. We are using this set of Concept Maps to shape new approaches and methods to systems development. This particular Concept Map describes the challenge of attaining *common ground* within the system development team and the means by which common ground may be achieved to help minimize problems associated with team coordination. Figure 3 shows one of a number of specific topics that were brainstormed as we reviewed and analyzed the interview data. Concept Maps such as this might be used by development teams to provide a rationale for requirements, in this case testable hypotheses about a new technology (haptic interfaces) to be integrated into a training system. Note that this Concept Map utilizes the “nested node” capability, in this case to highlight hypotheses concerning the use of haptic interfaces on training. Nested nodes are nodes that group a number of nodes (and their links) together within a single node. A nested node can be expanded (or collapsed) by clicking on the arrow box at the right-hand side of the nested node. Expanded nested nodes appear as large grey boxes in Figure 3.

4 Discussion

The cognitive task analysis results captured in the Concept Maps show that existing systems engineering and acquisition methods are inadequate in critical ways. They fail to support domain assessment. They fail completely to cope with the problem of emerging and changing requirements. They do not address team coordination sufficiently. These challenges are associated with: (1). An inadequacy of current artifacts (i.e., information and documentation management tools and documentation formats and categories), (2). Conflicts among engineers who use different approaches, and (3). Politics and agendas among individuals and organizations with different priorities that are often at odds with the goal of making systems that are usable, useful and understandable to the people who will be the eventual users of the systems.

The findings suggest that all stepwise descriptions of system development (i.e., spiral and waterfall modeling) are misleading. The empirical reality of systems development is that it is a macrocognitive activity in which key functions are parallel, simultaneous and highly interdependent (Klein, et al., 2003). The macrocognitive nature of systems development can be predicted on the basis of the laws of cognitive work and notions of Human-Centered Computing (Hoffman, et al., 2004; Hoffman & Elm, 2006). These laws and notions should be used to develop frameworks and methods that facilitate the true work of systems development. Accordingly, our goal is to use the results of the work reported here, especially the Concept Maps, to guide the development of a new systems engineering and acquisition framework. The framework, called the *Joint Systems Engineering Method* (JSEM), will hopefully address systems development challenges, support systems development in an empirically honest manner, and draw principles and methods of cognitive systems engineering and human-centered computing into systems development.

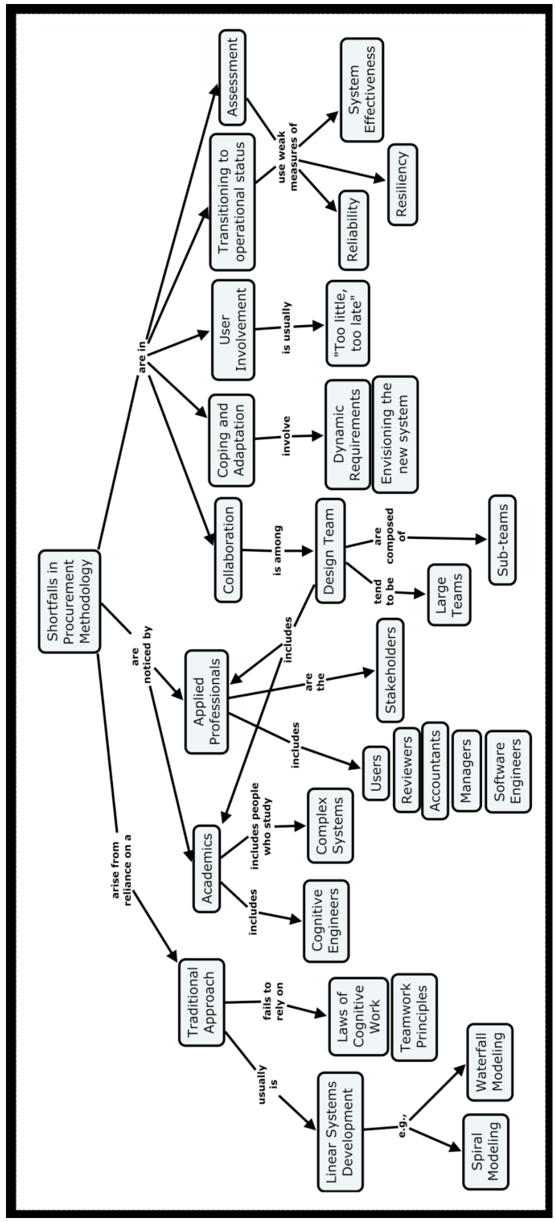

Figure 1. A Concept Map about shortfalls in the system development/procurement process.

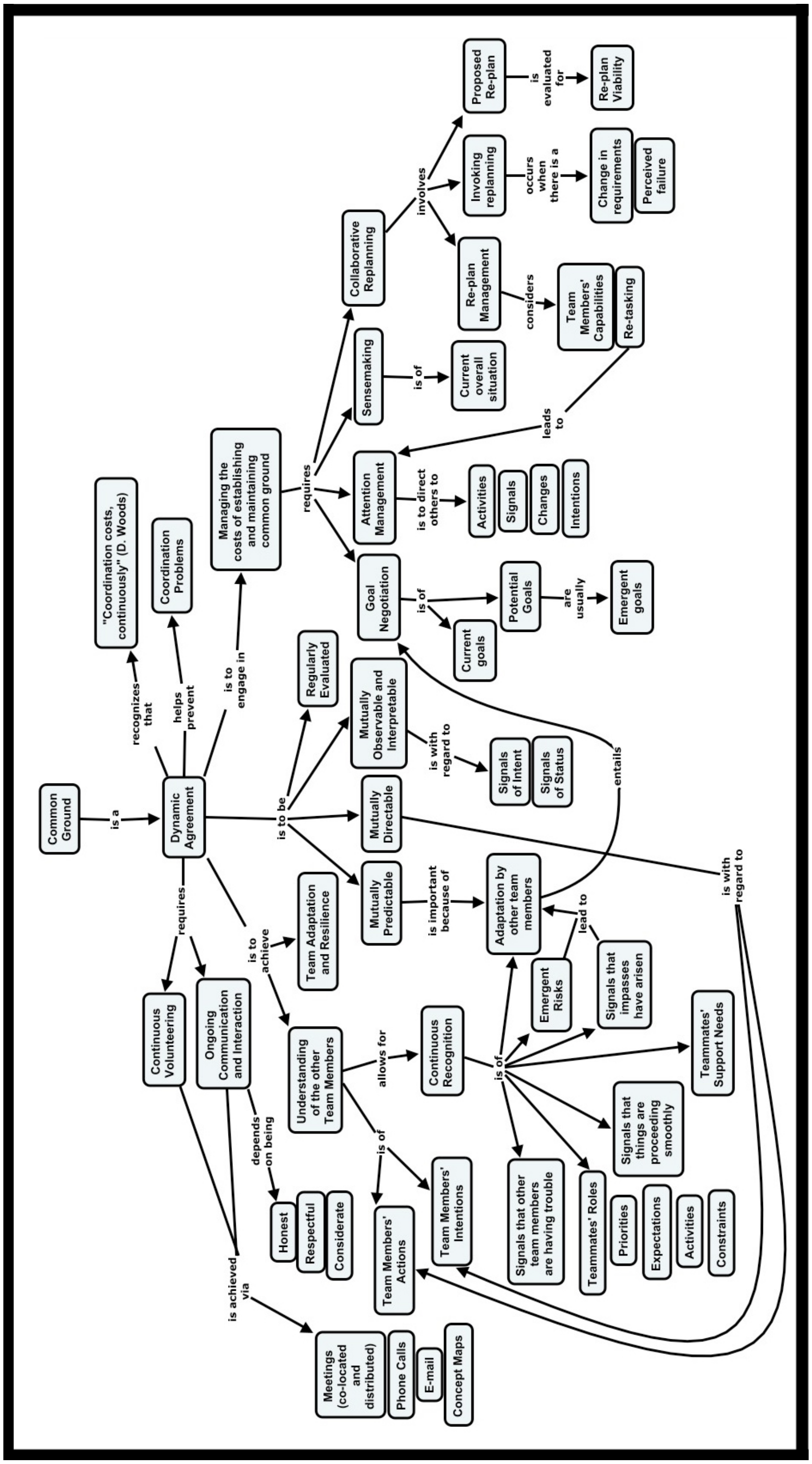

Figure 2. A Concept Map about the challenge of maintaining common ground.

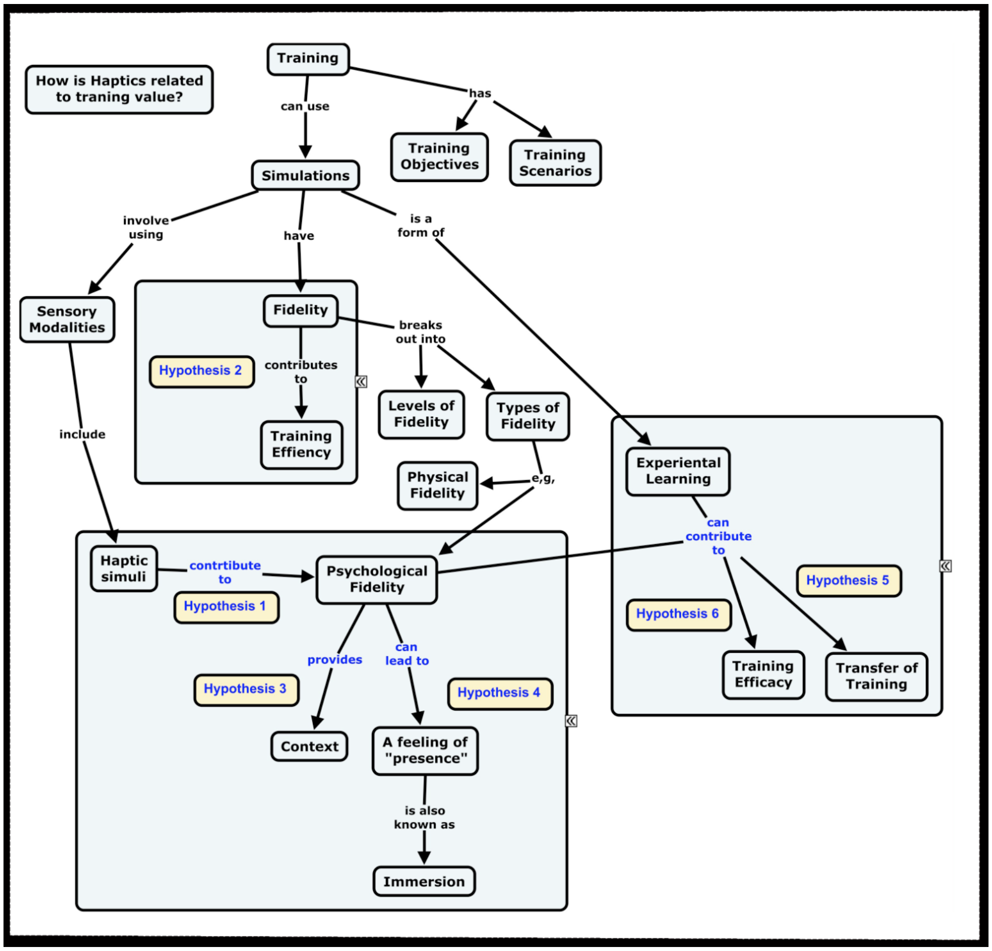

Figure 3. A Concept Map about hypotheses concerning training design.

Concept Maps additionally may contribute to the development of methods and tools that improve the roles of principles and methods of cognitive systems engineering and human-centered computing in systems development. Figure 3 demonstrates to us how a Concept Map can help a team capture the rationale for requirements and thereby contribute to informed risk management and decision making about requirements with respect to human-centered as well as more common technology-centered requirements. Concept Maps also have potential as communication/coordination tools and may help different disciplines such as cognitive systems and systems engineering work together. They may help users, customers, and developers share knowledge (although see Freeman, 2004 for contrary findings). They may contribute to the general avoidance of the team coordination problems that often plague systems development (e.g., Bush, 1997; Richey, n.d.). Another possible role for Concept Maps lies in the management of the significant amounts of information and knowledge that system development teams generate and use. These and other uses of Concept Maps as tools that help systems development teams will be explored as we shape and evolve the Joint Systems Engineering Method.

5 Summary

Large-scale software projects often fail to meet expectations because practitioners are not provided with the technologies that they need to perform their work effectively. Spiral and Waterfall modeling (and their variants) are often used to frame system development efforts, but subsequently end up as mere cover stories. In addition, "user involvement" in the design process is usually too little and too late. We conducted a cognitive task analysis to seek a more detailed understanding of the reasons for the high rate of systems development failure. We conducted a documentation analysis of about 50 sources (journal articles, books, book chapters, government documents, systems engineering standards, and website content) describing systems and software engineering methods and tools, and system development and acquisition processes. This analysis highlighted some limitations in systems development. Using a variant of the Critical Decision Method, we conducted in-depth interviews with six individuals having at least 20 years of experience in software engineering, including experience as managers of large-scale procurement projects. The data were represented in two ways. One was in the form of timelines showing when in the procurement and system development process activities are performed and where some new form of support tool might be useful to program stakeholders. We also represented the results in the form of Concept Maps. These were created to summarize results concerning problems and issues in the government procurement process. They make clear the challenges that confront all of stakeholders in the procurement process, and therefore are suggestive of how the system development process might be better managed. At the same time, Concept Maps might facilitate the integration of human-centering considerations as primary to insuring that systems are usable and useful.

6 Acknowledgement

The work reported here was made possible by sponsorship from the US Air Force Systems Engineering Center under the Program, "Embedding Cognitive Systems into Systems Engineering Practice" (SBIR AF05-071), through a contract, "Joint Systems Engineering Methodology," to Chi Systems, Inc.

7 References

Boehm, B. (1988). A spiral model of software development and enhancement. *Computer*, *21*, 61-62.

Brooks, F.P. (1995). *The mythical man-month*. Boston, MA: Addison-Wesley.

Bush, M. (1997). NIMROD case study. Retrieved 14 June 2006 from http://www.scism.sbu.ac.uk/~bushm/units/NIMROD_Case_Study.html.

Cannon-Bowers, J. A., & Salas, E. (1998). *Making decisions under stress: Implications for individual and team training*. Washington DC: American Psychological Association.

Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., et al. (2004). CmapTools: A Knowledge Modeling and Sharing Environment. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping* (Vol. I, pp. 125-133). Pamplona, Spain: Universidad Pública de Navarra.

Coyle, P.E. (1999, May). *Simulation based acquisition for information technology*. Paper presented at the 1999 Academia, Industry, Government Crosstalk Conference, Washington, DC.

Davenport, T.H. (1998). Putting the enterprise into the enterprise system. *Harvard Business Review*, July-August, pp. 1-10.

Department of Defense (2004). "Department of Defense Architecture Framework Version 1.0, Vol. 1: Definitions and Guidelines." Washington, DC: Department of Defense.

Department of Defense (2003). "Department of Defense Instruction 5000.2: Operation of the Defense Acquisition System." Washington, DC: Department of Defense.

Department of Defense (1996). "Mandatory Procedures for Major Defense Acquisition Programs and Major Automated Information Systems Acquisition Programs." Instructions 5000.2-R. Washington, DC: Department of Defense.

Eggen, D. (June 6, 2005). FBI pushed ahead with troubled software. Retrieved 14 June 2006 from *Washingtonpost.com*.

Freeman, L.A. (2004). The effects of concept maps on requirements elicitation and system models during information systems development. In A. Cañas, J. D. Novak, & F. M. González (Eds.), *Proceedings of the First International Conference on Concept Mapping: Concept Maps: Theory, Methodology, Technology* (pp. 257-264). Navarra, Spain: Pública de Navarra.

Gaitros, D.A. (March 2004). Common errors in large software development projects. Crosstalk. Retrieved 14 June 2006 from http://www.stsc.hill.af.mil/crosstalk/2004/03/0403Gaitros.html.

Hoffman, R. R., Ford, K. M., & Coffey, J. W. (2000). *The Handbook of Human-Centered Computing* (Deliverable on the Contract, *Human-Centered System Prototype*). National Technology Alliance. Retrieved 14 June 2006 from http://ihmc.us/research/projects/EssaysOnHCC/

Hoffman, R.R., Coffey, J.W., Carnot, M.J., & Novak, J.D. (2002). An empirical comparison of methods for eliciting and modeling expert knowledge. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*. Santa Monica, CA: Human Factors and Ergonomics Society.

Hoffman, R. R., Crandall, B., & Shadbolt, N. (1998). Use of the critical decision method to elicit expert knowledge: A case study in the method of cognitive task analysis. *Human Factors*, *40*, 254-276.

Hoffman, R. R., & Elm, W. C. (2006, January/February) HCC implications for the procurement process. *IEEE: Intelligent Systems*, pp. 74-81.

Hoffman, R. R., Roesler, A., & Moon, B. M. (July/August 2004). What is design in the context of Human-Centered Computing? *IEEE: Intelligent Systems*, pp. 89-95.

Klein, G., Ross, K. G., Moon, B. M., Klein, D. E., Hoffman, R. R., & Hollnagel, E. (May/June 2003). Macrocognition. *IEEE Intelligent Systems*, pp. 81-85.

Koopman, P., & Hoffman, R. R., (November/December 2003). Work-arounds, make-work, and kluges. *IEEE: Intelligent Systems*, pp. 70-75.

Leishman, T. R., & Cook, D. A. (2002, Apr). Requirements Risks Can Drown Software Projects. *Crosstalk*, *15*, 4-8.

Marketplace. (1997, January). *Marketplace for January 31st, 1997*. National Public Radio.

McConnell, S. (1996). *Rapid development: Taming wild software schedules*. Redmond Washington: Microsoft Press.

Nah, F.F., Lau, J.L., & Kuang, J. (2001). Critical factors for successful implementation of enterprise systems. *Business Process Management Journal*, *7*, 285-296.

Nuseibeh, B. (2001). Weaving together requirements and architecture, *IEEE Computer*, *34*, 115-117.

Richey, G.K. (n.d.). "F-111 systems engineering case study." Air Force Center for Systems Engineering. Retrieved 14 June 2006 from http://cse.afit.edu/page.cfm?page=49

Whittaker, B. (1999). What went wrong? Unsuccessful information technology projects. *Information Management and Computer Security*, *7*, 23-29.