Book Reviews

Integrity, Internal Control and Security in Information Systems: Connecting Governance and Technology. IFIP TC11/ WG11.5 Fourth Working Conference on Integrity and Internal Control in Information Systems November 15–16, 2001, Brussels, Belguim. Edited by Michael Gertz, Erik Guldentops, & Leon Strous. Boston, MA: Kluwer; 2002. 204 pp. $135.00. (ISBN: 1-4020-7005-5.)

This book is a comprehensive collection of essays and examples constructed on the importance of the concept of information technology governance. It employs a variety of resources from an international conference to show organizations and companies the necessity of integrating an information technology strategy into the overall strategic plan. As defined by the Information Systems Audit and Control Foundation (ISACA), IT governance is a structure of relationships and processes to direct and control the enterprise to achieve the enterprise's goals by balancing risk versus return over IT and its processes. In the book, IT governance is related to the extension of the mission and direction of a company to include information technology as an enabler to achieve established goals. Additionally, the role of information integrity in IT governance is explored for its significance in ensuring data integrity and accuracy. This book promotes the importance of IT to strategy decision makers and leaders in organizational management roles.

Integrity, Internal control and Security in Information Systems includes 12 papers from the Fourth International Working Conference on Integrity and Internal Control in Information Systems (IICIS), and is structured into five parts: Part 1 contains refereed papers that were selected by an international program committee. Part 2 is composed of invited papers that represent research conducted by experts in the field. Part 3 is a tutorial that explores the CobiT framework as an example of a system that can assist in implementing the fundamentals of IT governance techniques. Part 4 includes vendor white papers that present examples of software solutions that combat integrity problems. Part 5, the concluding panel session, explores problems in the areas of integrity and internal control that the conference would like to see addressed.

The contributors to the work range from different academic fields, newspapers, research institutions, and companies. The diversity of their backgrounds gives the collection breadth and depth. Overall, the chapters are organized in a logical manner and can be read with ease. The included papers address their subject matter from a mostly conceptual level, and any inclusion of research data or diagrams is generally nondisruptive to the narrative, making the text much easier to read as a whole.

Part 1, *The Refereed Papers*, includes a collection of literature regarding data integrity with a focus on confidentiality in systems composition. The first chapter examines the integrity capabilities of component-based systems and provides a rigorous definition of system integrity capacity. Also, a general discussion of the concept of integrity property includes multiple examples of contemporary security systems that exhibit the integrity capacity problem. A cautionary note regarding the data integrity capacity of certain secure systems was that systems could only trust data whose integrity is at or below that of the system interface components. One of the primary contributions of this paper is that it provides an initial approach to addressing the integrity capacity problem described in the literature and clarifies the limitation of a considerable body of previous work in the area.

The two following chapters are concerned with determining ways in which to overcome security-related problems. This topic is introduced by a survey and classification of the most commonly exploited software vulnerabilities. Categories of vulnerabilities are introduced, and extensive arrays of representative examples are provided. This categorization and examples can assist software engineers in identifying and decreasing the likelihood of the most commonly experienced security-related problems. These chapters provide a much needed focus on problem solution within this domain. There is often a great deal of problematizing of the situation that occurs, yet these articles seek to identify methods that decrease a variety of problems currently being encountered by software and security specialists.

The third chapter argues that the perils of evolving software are susceptible to destructive effects caused either by inadvertent errors or by deliberate attacks by malicious internal programmers. The author defines a concept of evolution-invariant software and shows that the dangers can be overcome by establishing overarching principles as invariants for a given financial software system. A remarkable finding here is that accounting principles for invariance have been formulated with a consistency of a small number of rules that specify and implement these principles. However, the approach in this paper may be oversimplified, particularly as it does not account for possible system failures.

The fourth chapter addresses issues of security in the specific domain of financial systems. The chapter addresses the nature of controls traditionally applied in the banking industry, the reasons why batch systems introduce risk into financial systems and why modern systems are moving towards real time to overcome this risk. The paper identifies the need for an entirely new set of controls and proposes an architecture of parallel, autonomous audit as a framework in which the new generation of controls might be developed. This paper could not be expected to have evaluated all possible risks involved. However, by clearly providing examples across different kinds of new payment systems and a reasonable summary of the new requirements, its contribution lies in a radically new paradigm for controls in the banking industry.

The last chapter of Part 1 reflects on general issues of information security by noting that the current computing environment is fairly homogeneous due to a relatively small number of operating systems and application functions. This situation allows attackers to focus their efforts on the few types of systems deployed. This paper examines several widespread computer attacks to understand the effect of technical diversity of the attacked systems, and to identify what role diversity played in the survival of these systems. Interestingly, the paper examines some of the most pervasive computer attacks and viruses and provides evi-

dence to support the conclusion that "diversity improves surviv-ability."

Part 2, *Invited Papers,* points out that having high-quality data cannot be over emphasized with the blooming of technologies such as data warehousing and data mining. In Chapter 6, various issues on data quality are examined, and an overview of current research in the area is given. Some of the emerging directions in data quality are discussed, including managing quality for the semantic web, and the relationship between data quality and data mining. It is a very brief summary on the development and direction of concurrent research to help the reader understand the concept and application of data quality.

The seventh chapter addresses the industry wide shift of utiliz-ing chip cards as a major step to address fraud. When the tech-nology infrastructure is already in place, it is especially important for users to understand the challenge of that innovative technology. An outline of the security features for Europay's chip card appli-cations is provided for Credit, Debit, and Electronic Purse Pay-ments. The legible overview of the EMV chip card specifications and the CEPS (Common Electronic Purse Specifications) enhances the understanding of various applications of chip card technology that are important to the overall theme of data integrity in systems being discussed here.

Part 3, Tutorial: "CobiT and IT Governance," is a valuable exploration of a systematic process that can be utilized to integrate information technology into the strategic vision of a company. *Control Objectives for Information Technology and related Tech-nology* (CobiT) is published by the Information Systems Audit and Control Foundation (ISACF) with a concentrated theme of busi-ness orientation. Currently, all portions of CobiT, except for the Audit Guidelines, may be downloaded on a complimentary basis from the ISACA Web site. In the tutorial, a series of questions that might be posed by an executive interested in IT governance are examined, with simple informative answers. The CobiT Frame-work is introduced with strategic pointers to show how CobiT can assist an organization in getting aligned with the importance of IT. The framework identifies the people who need to be involved to help implement CobiT principles in the organization. Also in-cluded in the tutorial are excerpts from the CobiT handbook published by the IT Governance Institute. This section of the book is extremely informative because it illustrates how IT governance can be implemented and where its value lies. In addition, many of the issues that could be confusing to someone unfamiliar with IT governance are simplified and placed into the context in which they are relevant. This part of the book could have been placed first to acclimate readers to the need for the integration of IT into strategic planning. In the last section of Part 3, there is a paper included from Royal Phillips Electronics regarding their experi-ence with using CobiT. Its inclusion adds validity to principles and strategies proposed by CobiT and the concept of IT governance.

In Part 4, *Vendor White Papers,* there are white papers that have been submitted by vendors who are involved with the areas of information and data integrity. They argue the case for why business process security and information integrity are integral to the success of a company and the achievement of its goals. Another framework example is proved with a system called X-TRA Secure. This system enforces business rules upon the users within a company, and targets them as the key to information security within a company. However, this system is highly intru-sive, requiring constant documentation and monitoring of internal systems users in a company. Understandably, rules must be en-forced in areas that deal with user access and security of informa-tion, but this system almost provides a "big brother" entity into the control and accessibility of information to users. Data integrity and information security are important. However, we must question the tradeoff between the need for security versus the resulting com-promise of the trust and autonomy of humans in our organizations.

Part 5, *Panel Discussion,* provides direction for further research in this area as well as questions that were left unanswered at the conclusion of the conference. Many of the questions examined deal with the state of complexity that exists in current technology and its impact on the ability to integrate its capabilities into the overall plan for the organization. This section provided an ade-quate conclusion to the book, because it organizes the unresolved questions that the reader might ask.

The value of this book lies within its ability to present the concepts, principles, and methods of IT governance so that a reader can understand the value in the process and the need for this type of strategy. The book also speaks to the overall importance of technology. For companies or organizations that are interested in leveraging IT for maximizing profit and the achievement of goals, IT governance is a method for doing so in a strategic manner. This book is interesting, informative, and comprehensive. One does not need to be an expert in the area of IT governance to see the value in this work; they just need to understand that IT can be utilized as a vehicle for the continued progress of innovation and the achieve-ment of strategic goals.

Allison Morgan
Cong Chen
School of Information Sciences and Technology
Pennsylvania State University
University Park, PA 16803
E-mail: amorgan@ist.psu.edu
E-mail: cchen@psu.edu

Leonardo's Laptop: Human Needs and the New Computing Technologies. Ben Shneiderman. Cambridge, MA: The MIT Press; 2002. 269 pp. $24.95. (ISBN: 0-262-19476-7.)

Years after Thomas Landauer warned us about the trouble with computers and Don Norman begged us to make them invisible, we find ourselves in a position where computing technologies are still visible and difficult to use. Unlike its predecessors, *Leonardo's Laptop* challenges technology developers *and users* to play their respective roles in shifting the focus from "what computers can do to what users can do." Shneiderman encourages users to be more responsible for the technology that they use; they need to expect and to demand more from computing technologies, and if neces-sary, to take political action. The author provides technology developers with a novel framework from which to design and develop solutions that support common relationships and activi-ties, especially in the areas of e-learning, e-business, e-healthcare, e-government, and creativity. Furthermore, the author encourages technology developers to follow Leonardo's left-brain–right-brain approach to improve design quality and universal usability. For readers who may think that all of this seems to be Pollyannaish, Shneiderman addresses potential oppositions in "The Skeptic's Corner" at the end of each chapter.

This book's warm and friendly tone makes it attractive to both technical and nontechnical readers. Its refreshing approach—to use Leonardo as an inspirational muse—empowers users and inspires developers. However, if readers are expecting a diagram or de-scription of a futuristic laptop a la Leonardo, then they will not find it among these pages; most of the technology discussion is about the four Web applications mentioned above. Readers will find an

interesting discussion and concrete guidance on how to better support human needs with technology. They also will find important nuggets of information on other issues like creativity and effective pedagogy.

The book begins with a discussion of the life, practices, and creative works of Leornardo. The chapter emphasizes Leonardo's use of various doodling and drawing media from portable notebooks to larger sketchbooks and plaster. It suggests the need for a line of computers from wearable devices to wall-sized models, which already exists. The real challenge is to leverage technology to integrate this line of computers. We have made some progress, in the area of context-aware computing in particular, but much more work remains to be done. For example, we still have to synchronize our various devices to our desktop or laptop computer. Why is it that the devices do not synchronize themselves?

The opening chapter emphasizes Leonardo's integration of many artistic and scientific disciplines in his work and suggests that designers of the new computing need to adopt this approach. Realistically, we cannot expect every designer to be able to seamlessly integrate art and science; typically, people are adept at one or the other. However, we can strive to create design teams that are capable of carrying out this charge. One thing that is not offered in the book is a discussion of specific areas of art and science, other than the ones we traditionally use, which could provide some value in technology design. Are there other skills that we need to develop or principles that we need to know? Are there other practices that we need to follow?

The chapter provides two goals to help focus future technology development efforts: enabling users to accomplish tasks, mainly supporting human relationships; and moving from machine-centered automation to user-centered services and tools. One could argue that the first goal is a long-standing goal of technology developers, but an emphasis on supporting human relationships is an interesting caveat. When I thought about some of the nonrelationship things that I do (e.g., organizing files and installing programs), I had a hard time accepting this notion that all tasks revolve around relationships. It was not until Chapter 5 that I learned that these mundane tasks could be considered as supporting a relationship with myself, which I do not agree with. I consider finding a movie to see or tracking the things that I need to do as supporting a relationship with myself; some things are just necessary evils.

One could also question whether we need to completely abandon machine-centered automation or whether we need to employ it in areas where it makes sense and works. For example, I would be happy if my devices automatically synchronized themselves, but I would not want my computer to automatically organize my files or install programs. Perhaps we need to strive for a balance between the two extremes.

Chapter 2 rehashes the countless fiascos that unusable technology has caused. It proposes that rather than focusing on technology breakthroughs, we need to focus on quality breakthroughs. It calls for a users' movement as a catalyst for the quality breakthrough, similarly to what happened in the automobile industry. Like Shneiderman, I wonder if technology users have become extremely dispassionate about this issue and have accepted that they need to put up with substandard technology, and furthermore have resigned themselves to thinking that there is nothing that they can do. Have we not been complaining about technology since its inception? Can we honestly expect to be heard now when we have not been listened to for so long? I actually like the alternative approach that Shneiderman proposes to raise the stakes even higher; it calls for software companies to pay users for wasted time and effort due to software failures. We could all be instant millionaires if this policy was enforced! But, we should not expect our checks in the mail any time soon. What is the best approach to hasten this quality breakthrough?

Chapter 3 discusses universal usability or users' ability to use technology, regardless of income, education, disabilities, etc. It points out that Leonardo was concerned about the underprivileged; thus, he sought to serve both the rich and the poor. The author states that the major challenge in this area is for technology developers to "bridge the gap between what users know and what they need to know" in accomplishing their tasks (e.g., how to begin using a computer, recover from a crash, or get rid of a virus). If we took this statement at face value, then we might think that we need more paperclips and other assistants, which we know do not work.

Chapter 4 discusses methods and tools, such as a user-centered design process and validated design guidelines, for addressing the challenge of bridging the gap between what users know and need to know. These methods and tools have been promoted for a long time, yet we are still in a position where this book is valuable. Thus, the chapter's contribution is unclear. Perhaps Shneiderman designed the chapter for technology developers, who need to be reminded to focus on supporting users. It would have been good to take this discussion a step further; for example, to discuss ways to help developers to adopt these approaches or to better support designers and developers in using them.

Chapter 5 presents a novel framework for designing technology to address human needs. It presents a 4×4 activities and relationships table (ART) to guide technology designs. The chapter identifies four relationship levels to consider: self, family and friends, colleagues and neighbors, and citizens and markets. It also identifies four common activities to consider: collecting (information), relating (communicating), creating (innovation), and donating (dissemination). As stated above, I find it difficult to accept the notion of doing tasks to support a relationship with myself.

The chapter suggests that technology solutions need to address each relationship and activity combination, which could be a daunting task. The remainder of Chapter 5 demonstrates how photo, PDA, and other applications address these combinations. Chapters 6–10 further demonstrate use of the ARTs to guide innovations in supporting e-learning, e-business, e-healthcare, e-government, and creativity. One thing becomes apparent from exploring the various ARTs in these chapters—addressing every combination can be quite challenging, even for its developer. All the ARTs in the book have empty cells, which one can consider as open research or development areas to think about. Perhaps these open areas might not be considered without going through the ART exercise.

Other than an overview of the ART and examples in several domains, there is not much guidance on how to use this tool. How do you incorporate it into work practices? How do you know when you have addressed enough of the combinations? Are certain combinations more important than others? All the examples appear to be retrospective analyses of existing technology. It will probably be more difficult to apply it to new technology.

The final chapter revisits some of the trends and predictions about the new computing, including Ray Kurzweil's prediction that "Before the next century is over, human beings will no longer be the most intelligent or capable type of entity on the planet- . . . Machines, derived from human thinking and surpassing humans in their capacity for experience, will claim to be conscious, and thus to be spiritual." Shneiderman rebukes this type of thinking and claims that it leads to an emphasis on replacing, rather than supporting humans with technology. He suggests that we turn our attention to even grander goals like using technology to improve the quality of life (e.g., reducing homelessness, illiteracy, automobile accidents, and the threat of war).

Shneiderman discusses several application areas that could improve the quality of life, specifically supporting e-learning, e-healthcare, e-business, e-government, and creativity. What about e-communication? How can we design better technology to mitigate email overload, for instance? It would have been interesting to

see an ART to provide more insight for dealing with this pressing problem.

Shneiderman concludes the book with a call for the next Leonardo. Answering this call means that one must commit herself to two of Leonardo's principles: (1) "technical excellence must be in harmony with user needs," and (2) "great works of art and science are for everyone." Hopefully, the tools and insight the book provides will help us to adhere to the first principle much sooner. The second principle may be harder to adhere to given the capitalist society in which we live.

Overall, this is a provocative and enjoyable book on how best to design the new computing technologies and what human needs or application areas are most important. Shneiderman shares an important and compelling vision for designers, entrepreneurs, and others concerned about shaping future technology.

Melody Y. Ivory
Information School
University of Washington
Seattle, WA 98195-2840
E-mail: myivory@u.washington.edu

Social Thinking—Software Practice. Edited by Yvonne Dittrich, Christiane Floyd, & Ralf Klischewski. Cambridge, MA: The MIT Press; 2002. 481 pp. $55.00. (ISBN: 0-262-04204-5.)

Social Thinking—Software Practice, is an odd book. It combines what is usually the domain of computer science, computer programming, and a social science approach to the analysis of programming as an activity—a social activity. Perhaps the one sentence that best characterizes the antithesis of the approach that this collection of essays takes comes from Chapter 2, *On Foundational Categories in Software Development*: "[T]echnology, not the social world whose product it is, frames both the questions and the answers designers and developers come up with" (p. 29). Thus, what the authors are arguing for is a rethinking of the education of software developers, and ultimately software practice. The approaches that are advocated are relatively well known to those who follow the European theories on software development, but not that well known to most computer science departments here in the United States. Among the approaches are ethnography, participatory design, activity theory, collaborative approaches (coming out of CSCW research), etc. Whatever the approach, however, the authors are arguing for putting the users at center stage and recognizing that a piece of software is a social product and that sociology (in the broadest sense) is as central to application development as programming.

As for the specifics of the book, it is difficult to go into any great detail in a work with 21 chapters by 34 authors. The 21 chapters are divided into five sections: Deconstructing, Informing, Grounding, Organizing, and Reorienting. For me, this organization was a bit of a stretch, but I suppose a book with so many chapters and so many different authors would have been an impossible task without it.

The *Deconstructing* section seeks to question "established paradigms" and point to new concepts in software development. It would seem that the established paradigms questioned are the traditional approaches taken toward software development in the United States. Certainly, ethnography and participatory design, the established paradigms in the UK and Scandinavia, are never questioned. Unfortunately, there is no name that I know of for this (USA-centric) approach other than, "the established paradigm." A friend of mine from industry once told me that the only methodology that industry recognized was, "I want it now, I want it fast, and I want it cheap" (this was certainly not lost on many of the authors, e.g., Dittrich, p. 181). But even that is not what the authors are arguing against. The "established paradigm" as they see it is the assumption that writing computer programs for public consumption is merely a technical problem. This paradigm further assumes that mathematical elegance is the determining factor in program design. Algorithms, not *agora*, are the primary concern of this paradigm.

The *Informing* section analyzes software development from the point of view of a social science. The introduction to this section insists that the articles are to be viewed principally as descriptive, not *prescriptive*. That is about as convincing as claiming that allegories are descriptive, not prescriptive. Perhaps most compelling for me from this section was Eevi Beck's chapter, "What Doesnt" Fit: The "'Residual Category'" as Analytic Resource. Beck cites Gadamer and the "hermeneutic" tradition as being a useful approach. What I think she was really calling for was a descriptive phenomenology along the lines of Schutz. (I would not say Husserl, for learned scholars assure us that Husserl was one of those pesky Cartesians. "And surely they are all, all honorable men" . . . and women.)

In the *Grounding* section, the authors use "empirical research as well as theoretical concepts" to advance the arguments made up until now. (Again, I'm not sure that any of these chapters could not have fit into one of the other sections.) Topics discussed include a couple more conceptual frameworks for understanding systems development, (more) examples of how to actually implement these ideas, and (another) case study. If there is one chapter that best accomplishes (in my mind) the attempt to bridge the two traditions, it is Dittrich's, which argues for the very fact of doing empirical research in software design.

In the *Organizing* section, changes in IT practice are related to larger issues of organizational change. Again, these examples seem to come from the Scandinavian countries. How that plays in Peoria isn't addressed. One intriguing chapter in this section is Korpela et al.'s chapter on how software is introduced and used, not developed, in organizations. "The essential message is that IS researchers need to engage in collaboration with IS practitioners . . ." (p. 305, italics are the authors).

Lastly, in the *Reorienting* section, I am not so sure we are reoriented as we are exhorted basically to apply the lessons advanced in the earlier parts of the book.

As for the book's intended audience, it includes "readers with personal experience in the field of computing, such as software researchers, teachers or practitioners on the student or professional level, and social scientists closely involved with software practice" (p. xi). The one group the editors leave out, ostensibly, is philosophers of science—natural and social. In fact, this may be the group most interested in the book. For what this book presents is truly a battle of worldviews—a battle between humanists and technologists, or a battle between socialists and capitalists (I'm not quite sure). Well, not so much a battle, because the other side never seemed to be invited. Horkehimer, Adorno, and Marx are all cited, but not Sellars or Carnap. Feyerabend is cited, but not Kuhn, Lakatos, or Popper. Moreover, large parts of the continental tradition—Schutz, Merleau-Ponty, or others of the Phenomenological school (or schools, depending on one's perspective)—appear nowhere. This is an interesting oversight, because, from the very first chapter, a contrast is drawn between the logical/empirical tradition and the hermeneutic/dialectic tradition. It would seem that if one wanted to clearly describe the differences between artifacts and people, one might start at a descriptive methodology (descriptive phenomenology).

Another key concept discussed in the book is that of "software practice." Floyd states, "software practice refers to the sum total of activities involved in developing and promoting the use of software" (p. 5). This being the case, it seems the book's audience would also include the marketing and sales departments, as well as the accounting departments of companies developing software in house. These organizations are central to determining what gets created, how much time is spent on developing (and the resultant "schedule crunch"), etc. One would think that until these departments—always much more powerful in any organization that programming—see some overriding need to change the way software is developed and sold, that little is going to change. I am not saying that this is right, only that it is the way of the (capitalist) world. In implicit recognition of this fact, Nissen, in Chapter 4, states in conclusion, "Software and hardware development has not just been technology driven in the past forty years. The industry has also profited, and still does, from a seller's market" (p. 87).

First, and I believe one of the more serious problems with this book, is that the intended purpose of the authors is unclear to the reader. I was never quite sure whether they meant for the book to be a treatise on the philosophy of science (i.e., computer science), an essay on the sociology of application development (software engineering?), or a prescriptive "call-to-arms" of a neo-Marxist sort (exhorting programmers to a more democratic, collaborative—if not collegial—approach to the design of software artifacts) against the forces of, what, Capitalism, soulless technology, or what? For protestations aside, this is a very prescriptive book. Also, it is hard to imagine a *non*philosophical work citing Horkehimer and Adorno. And the capitalist "just get it done" mentality would not receive a warm welcome in the pages of this book. That being the case, one wonders why the problems that other domains may have had integrating social theory into their practice were never discussed. Did not engineering, or accounting, or any other discipline have problems properly addressing the needs of their users/customers? What were their experiences?

Along those same lines, while the authors provide an interesting discussion of a more humanistic approach to the very notion of software development, they never make an overriding *need* for changing the way software is created in the United States and much of the rest of the world. Nissen, in Chapter 4, assures us that the nonhumanistic approach to application development won't go on forever, and it probably won't. But, as Kuhn (1969) pointed out, barring some crisis in software development, things will remain the same indefinitely, and that is precisely what I meant above about the authors making a convincing case for a social science approach to software development. Where is the crisis? Where is the business disruption? Where are the voices crying out for "salvation"?

The book also contains a number of problematic presuppositions that detract from the reading experience. For instance, social science for the most part is juxtaposed to systems development. Rarely, one runs across the term computer science instead of systems development. These latter terms are generally used interchangeably. The problem with this is that the social sciences are regarded in the book as an academic discipline, whereas systems development is of the domain of practice. But if systems development is equal to, or derivative of computer science, then does not systems development also have the right to lay claim to academic robes? It might have been argued (but never was) that as programming leaves the realm of computer science in academia (*theoria*) and becomes application development (*praxis*), it takes on this new characteristic. If simply leaving academia makes something nonacademic, then what about the application of social science methods to application development? This may seem to the reader an overly philosophical argument, but this is a very philosophical book. In fact, it indicated a possible direction for the justification of philosophy in the modern age—a discussion championed, if not initiated, by Mohanty (2000).

Lastly, there is a bit more hyperbole than I care to read in a book—*any* book. Nyce and Bader (Chapter 2), for instance, make the assumption that the national character of Swedes makes them somehow different from other individuals (other Europeans? Americans?). Okay, I'll bite. But when they assert that, "in Sweden, speech has the potential to triumph over differences in power, authority, and hierarchy" (p. 39), I get the feeling that they never head the English expression that the pen is mightier than the sword, or that they never bothered to read American history. Indeed, it is this type of hyperbole that might turn off otherwise sympathetic readers. In fact, the difference is probably due to the fact that Scandinavia's attempts in Socialism/Marxism have been fairly successful, and that computing on the Continent does not seem to be dominated by computer 'science" departments.

My final criticisms of the book concern the applicability of the proposed social science-based work practices. I remember at a conference once asking someone who was expounding the virtues of ethnography and participatory design how I could take this interesting, but time-consuming, approach to systems development back to my shop where I might have 2 months to analyze, design, debug, and rollout an application. My question was met with the sort of silence I meet with when accidentally using Philly street slang when talking with my Japanese stepsister. Something about this book leaves me with the same feeling.

The obvious preference, if not to say prejudice, in this book is for the Scandinavian approach to software development. But if there are different approaches based on national character or ethnic preferences (remember, this is the claim of Nyce and Bader, not me—is this the Balkanization of software development?), are there also not different ways to *properly* do software development based on national character or ethnicity? The fact (I assume that it is a fact, being the descendent of poor, shantytown Irish Americans) that the Scandinavian approach works, the book does not address whether this approach would work with Americans—a people who are pragmatic, capitalistic, and nonworker centric to the core. Given the above, what I really would like to have seen, for example, in the *Grounding* section, is a discussion of whether software practice is culturally determined. That is, is the "Scandinavian" approach particularly Scandinavian. Would it work in the United States? Would it work in China (the coming software powerhouse, from what I read)?

Another application question the authors fail to address is, "How important is it to do things 'right'"? After all, Wanda Orlikowski (1992) wrote a classic paper on the debacle of the Lotus Notes rollout at "Alpha" (exactly who "Alpha" was is doubtless one of the worst kept secrets in history). "Alpha" did almost everything wrong. Years later, however, Lotus was a staple of the organization. Interestingly, the one thing that was done "right" was the support—read, "mandate"—from upper management that everyone would use Lotus Notes, lest they feel upper management's wrath. I'd like to think that this is not the correct software methodology. Kaptelinin (Chapter 3) seems to assure us that this approach will not last forever. I want to believe him. My 15 years of industry experience, however, inclines me the other way.

In conclusion, although I would agree with the book's authors that application development needs a theoretical underpinning, I am not sure that this book will have great sway in the places where it is so desperately needed: in departments of computer science and computer engineering, particularly here in the United States of America where, according to Nyce and Bader, reason and democracy are not much cherished anyway. In the end, however, it might be that neither the hyper-rationalism of thinkers such as Carnap and Popper, nor the kinder, gentler social science approach suggested by these authors accurately reflects the process of software development. It may be, as

Feyerabend (1975) argues, that the method of software development, like that of science in general, is "anything goes."

Lewis Hassell
College of Information Science and Technology
Drexel University
Philadelphia, PA 19104
E-mail: lew.hassell@cis.drexel.edu

References

Feyerabend, P. (1975). Against method. London: NLB.
Kuhn, T. (1969). Structure of scientific revolutions (2nd ed.). Chicago, IL: University of Chicago Press.
Mohanty, J.N. (2000). The self and its other: Philosophical essays. Oxford, MA: Oxford University Press.
Orlikowski, W. (1992). Learning from Notes. In Proceedings of the conference on computer supported cooperative work '92. New York: ACM Press.

Dark Fiber: Tracking Critical Internet Culture. Geert Lovink. Cambridge, MA: The MIT Press; 2002. 382 pp. $27.95. (ISBN: 0-262-12249-9.)

This book focuses on the people and groups outside of businesses and governments, such as user groups, social movement, NGOs (nongovernment organizations), and artists, who use the Internet to propose ideas and theories outside the mainstream. Lovink argues that corporations and governments are trying to blockade the Internet from any voice of dissent, and that only by using "net criticism" and "tactical media" can the Internet be kept free of corporate and state control.

There are seven parts each divided between two and six sections of varying lengths. The first part, "Theory," is a discussion on speculative media theory and the virtual intellectual. The first essay is a fairly detailed look at German media theory and its background. The second essay is a discussion of the role of the intellectual in the virtual world. These two sections set the tone for the work and provide the readers with some foundation and definition for Lovink's ideas and theories.

The second section then shows the theories in action with two case studies, The Digital City and Nettime. Lovink shows how the Amsterdam Digital City evolved from freenet, with free access, e-mail, Web space, into an ISP. Lovink skillfully demonstrates how the Digital City's move from a freenet environment to a more corporate structure caused the "City" to deteriorate. The second section, "The Moderation Questions: Nettime and the Boundaries of Mailing List Culture," is a close examination of the history of nettime and the issues surrounding mail lists. Created in 1995, the nettime mailing list is a "cross disciplinary, international exchange for the 'cultural politics of the net'" (p. 68). These two essays are two of the most detailed discussions in the book.

"Crystals of Net Criticism" has six sections covering the issues of language, Push media, mass psychology, creating a standard global Internet time, storing collective memories, and the erosion of e-mail. Together these six short sections create a collage of important cultural issues. For instance, in "Fragments of Network Criticism," Lovink examines questions related to the storage of collective memories. Who will decide what gets stored and what gets deleted? Who will set the standards and rules for storage? As Lovink states "Media memory asks about the role of the machine in the social process of remembering and the contribution of technology in the everyday life dialogue with the past . . ." (p. 163).

The fourth set of essays is by far the best at demonstrating Lovink's complicated and intricate theories. "Travelogues" contains three essays that dramatically demonstrate the global impact of Internet culture. The first essay, "Culture after the Final Breakdown: Tirana, Albania, May 1998," is a view into how artists and nongovernment organizations in one of Europe's poorest and most isolated regions are using the Internet. Next, Lovink explains how the September 1999 earthquake in Taiwan was the impetus for people to use the Internet as a civic tool for discussing public issues. The final essay covers the story of a media center in Delhi that provides public access to the Internet. All three of these examples operate outside mainstream business and government organization to provide people with a voice and source of information where ideas can be shared freely.

One of the best sections of Part Five, "Dynamics of Net Culture," is the essay on tactical media, "An Insider's Guide to Tactical Media." Here, Lovink expounds on Internet activism and provides the readers with some history of tactical media. As Lovink explains, the "term tactical was introduced to disrupt and go beyond the rigid dichotomies that have restricted thinking in this area for so long," and that tactical media "are into questioning every single aspect of life, with 'the most radical gesture'" (pp. 255, 256).

Part Six, "Reality Check," covers a number of important topics. "Soros and the NGO Question, or The Art of Being Independent" is a good discussion of the how nongovernment organizations build on movements. One of the most thought provoking essays is "Information Warfare: From Propaganda Critique to Culture Jamming." One example of Lovink gives, as a positive example of infowar, is the proliferation of Linux as an option over using Microsoft.

The final collection of essays, "Towards a Political Economy," is a discussion of the rise and fall of dotcoms, bandwidth issues, and the question of "free" in what Lovink calls the "dark side of the electronic gold rush" (p. 357). Among the examples Lovink uses are Napster and the losses suffered by Dell and Cisco due to overproduction.

Dark Fiber is a timely and important collection of essays, but the book is not easy to read. The flow seems disjointed and each essay lacks context. Part of the difficulty is in the transition from one media to another. Although the author states that by putting his postings in book form "offers an opportunity for the author to reassess and filter thoughts into a comprehensive body of files" Lovink forgets that his book will be many readers' first introduction to nettime and his theories. So, readers will not have the context of the discussion before and after individual postings. For instance, "Fragments of Network Criticism" was posted on nettime on August 23, 1999, under a different title. Going to nettime and finding and reading the original posting and as well as the 10 or so responses in the thread make for much clearer understanding of the topic and gives the reader a much better sense of the topic's context. Also, although Lovink does use endnotes to document sources, he does not provide the original title of essays from nettime postings, which makes it harder for the reader to go into the archive to investigate the essays and topics that interest them. Also missing from the work is a conclusion tying the various essays together.

Despite the above weakness the book is an important addition to growing body of work on Internet culture. Lovink does create a sense of what is to come now that cyberspace utopia and dot.com mania has past. The essays also give readers a new way to think critically about the Internet. Readers are challenged to consider how various groups use the Internet to promote their own agendas, for good or bad. A self-proclaimed radical media pragmatist, Lovink's portrait of an Internet controlled by governments and corporations and offering no voices of dissent is a frightening one.

The work also fills an important gap in the literature. Most works concentrate on how people are steering the Internet instead of how the Internet is steering people. Another strength of this work is that Lovink has been a part of or knows the people involved with the projects discussed in the book. Overall, *Dark Fiber* is recommended especially for academic libraries.

Lisa A. Ennis
Georgia College & State University
Milledgeville, GA 31061
E-mail: liennis@gcsu.edu

Research Methods for Students, Academics and Professionals: Information Management and Systems, 2nd edition. Edited by Kirsty Williamson. Wagga Wagga, NSW: Centre for Information Studies, Charles Sturt University; 2002. 352 pp. (softcover). $AUD60. (ISBN: 1-876938-42-0).

This book is the most recent of only about half a dozen research methods textbooks published for information science since 1980. Like the others, it is directed toward students and information professionals at an introductory level. Unlike the others, it describes an unusually wide variety of research methods, especially qualitative methods.

This book is Australian, with a concern for human behavior in keeping with that country's reputation for research in the social sciences and development of qualitative data analysis software. The principal author is Kirsty Williamson, who wrote or co-wrote half the chapters. Eleven other authors contributed: Amanda Bow, Frada Burstein, Peta Darke, Ross Harvey, Graeme Johanson, Sue McKemmish, Majola Oosthuizen, Solveiga Saule, Don Schauder, Graeme Shanks, and Kerry Tanner. These writers, most of whom are affiliated with Monash University or Charles Sturt University, represent multidisciplinary and international backgrounds. The field they call information management and systems merges interests of information management or information studies (including librarianship, archives, and records management), and information systems, a subdiscipline of computing that focuses on information and communication technologies.

The stated purpose of the book is to help information professionals become informed and critical consumers of research, not necessarily skilled researchers. It is geared toward explaining not only methodology, but also the philosophy, relevance, and process of research as a whole.

The Introduction and Section 1 establish these themes. Chapter 1, on research and professional practice, explains the value of research for solving practical problems, maintaining effective services, demonstrating accountability, and generally contributing to useful knowledge in the field. Chapter 2 on major research traditions presents a broad picture of positivist and interpretivist paradigms, along with a middle ground of post-positivism, in such a way as to help the new researcher grasp the assumptions underlying research. Woven into this chapter is an explanation of how quantitative and qualitative methods complement each other, and how methodological triangulation provides confirmatory benefits. Chapter 3 offers instructions for beginning a research project, from development of the research problem, questions, and hypotheses to understanding the role of theory and synthesizing the literature review. Chapter 4 on research ethics covers unethical use of power

positions by researchers, falsifying data, and plagiarism, along with general information on human subjects protections and roles of ethics committees. It includes intriguing examples of ethics cases to stimulate discussion.

Sections 2 and 3 make a key distinction between research methods, which encompass the theories and purposes underlying research design, and research techniques, which are specific means for collecting data. The rationale is that one research technique, such as interviewing, may be appropriate for more than one research method, such as survey or case study.

In Section 2, eight chapters describe survey, case study, experimental, system development, action, ethnography, historical, and Delphi research methods. The methods progress roughly from most to least used in information science, and for the least used, the authors take pains to elucidate the means to achieving methodological rigor. Chapter 8 presents a noteworthy argument for legitimizing system development as a valid methodological approach within the larger context of information systems research. System development is seen as belonging to the cycle of theory to practice required to create effective information systems, a cycle that emphasizes human and social aspects as a necessary counterpoint to the obvious technological aspects.

The four chapters in Section 3 discuss specific techniques that may be used with different methods. Chapter 13 on sampling summarizes probability and nonprobability sampling techniques and when they are appropriate. Chapter 14 describes the two most common data-collection techniques, questionnaires and interviews, and looks at their respective uses. Chapter 15 covers focus groups and Chapter 16 ethnographic techniques, including participant observation. Throughout Sections 2 and 3, attention is paid to the subtleties of collecting data from people, such as ways to obtain access and avoid major types of biases.

In Section 4 on data analysis, only Chapter 17 deals directly with analyzing quantitative and qualitative data. It does so in limited space by describing the general process for handling each type of data. This is followed by evaluating research publications in Chapter 18, which offers valuable advice for critically assessing studies that employ different methods. The last part of the book is a postscript with seven questions that invite readers to reflect on issues of focus and ethics, to become aware of their responsibility for approaching research conscientiously. Although these three parts together do not constitute a unified conclusion, each does provide thematic closure for preceding chapters.

Writing a book of this sort presents certain challenges that the authors have conspired to tackle through organization as well as content. One of these challenges is presenting vital and pervasive research issues. These are nicely bounded by the structure of the book, with philosophical, social, and ethical considerations introduced in Section 1, revisited in middle chapters, and reinforced in the postscript. A second challenge is untangling the complexities of interrelated research methods. Here the strategy of distinguishing between research methods and research techniques is carefully explained, but admittedly strained. In separate chapters, for instance, survey is presented as method, and questionnaire (commonly called survey) as technique; ethnography as method and as multiple techniques; Delphi as method when it is also technique; and focus group as technique when it is also method. A third challenge is deciding where to stop in a book of medium length. The introduction states that bibliometrics and content analysis are omitted, although Chapter 17 on data analysis does cover some content analytic techniques under the heading of qualitative analysis. And while software packages for analyzing quantitative and qualitative data are mentioned, computer-based techniques for data collection, such as transaction logs, are not. Generally, the authors favor discussion of more obtrusive approaches to data collection (excepting historical) and their concomitant issues of human interaction.

Despite its many authors, the book is remarkably consistent in tone, comprehensible, and easy—even enjoyable—to read. The chapters function as self-contained units, each beginning with learning objectives and ending with discussion questions, further reading, and references. The book contains numerous examples and graphics, a glossary, and an index. The examples, many drawn from the authors' own research, represent an impressive range of research problems. In this second edition, published just two years after the first, the chapter on research ethics and a set of author bios have been added. A few other improvements might have been made. First, the references could have been updated from the first edition. Second, the downside of having multiple authors is that the methods chapters vary in structure and emphasis, making it difficult to compare methods. It would be helpful to novice researchers to have some general criteria for choosing methods before they encounter these chapters. This could be done in Chapter 3 on beginning the research design process, which now stops at the literature review. It would also help to explicitly list advantages and disadvantages in each methods chapter, as only half the chapters have such lists now.

Although this book is not explicitly or exclusively devoted to qualitative methods, it fulfills a need for more coverage in that area in information science. It is sufficiently clear and readable for any level of university student, although for graduate students I would supplement it with selected skills readings. It is an excellent introductory resource that delivers lots of bang for the buck in relatively few pages, compared to tomes in other social sciences that are twice as expensive and contain twice as many pages of detail that may never be utilized.

Linda Schamber
University of North Texas
Denton, TX 76203-1068
E-mail: schamber@unt.edu