

Chapter 7.3

Software Use Through Monadic and Dyadic Procedure: User-Friendly or Not-So-Friendly?

Gregory E. Truman
Babson College, USA

ABSTRACT

Our research objectives are to provide a theoretical discussion on how software may impact user performance in ways contrary to designers' intentions and users' desires, and to empirically evaluate user performance impacts that derive from ostensibly performance-enhancing software features. We propose that dyadic procedure is associated with higher levels of user performance when compared to monadic procedure. Using word-processing software utilization as the research context, we test the proposition on data from 46 participants. Contrary to expectations, the results suggest that dyadic procedure may decrease the accuracy of users' work. We conclude that software design features that are intended to improve user performance may have opposite effects, which raise questions about these features' utility and desirability.

INTRODUCTION

Users' access to and utilization of computers have become widespread due in part to developments surrounding graphical user interfaces, multifaceted packaged software, the Internet, and electronic commerce. However, there are indications that suboptimal utilization persists (Brynjolfsson, 1996; Marcolin, Compeau, Munro, & Huff, 2000), which challenges a frequent assumption that unqualified utilization is positively related to performance (Thompson, Higgins, & Howell, 1994). Thus, it may be that users do not know how to carry out effective and efficient computer use, which may potentially have adverse consequences for individual and firm-level performance.

Partly in response to this situation and partly due to competitive necessity, software vendors have continually improved their products to increase ease of use and to enhance user performance outcomes. Despite these advances, we contend that software innovations do not uniformly produce favorable performance impacts. Moreover, we argue that a critical examination of software innovations' impact on user performance is needed

for two related reasons. First, the various ways that users apply an innovative feature do not always coincide with developers' intentions, therefore the effects of any feature cannot be fully predicted. Second, while a software designer may *intend* that an innovative feature enhance user performance, the feature may *actually* bring about reduced performance.

Research Objective

Our research objectives are to provide a theoretical discussion on how software may impact user performance in ways contrary to designers' intentions and users' desires, and to empirically evaluate user performance impacts that derive from ostensibly performance-enhancing software features.

Research Scope

Our characterization of computer use relates to software that end users typically utilize in the workplace. End users include workers whose formal role designation lies outside the IS area (McLean, Kappelman, & Thompson, 1993) and who commonly use software referred to as productivity software. Productivity software includes Microsoft Office, Corel WordPerfect Office, Lotus SmartSuite, and the like.

A focus on this type of software is particularly relevant because it is pervasive in the workplace. Among U.S. workers who use computers, about 63 million employees or about one half of the U.S. workforce as of October 1997, 57% use word- or document-processing programs, 41% use spreadsheet or analysis programs, and 26% use desktop publishing or graphic programs (U.S. Census Bureau, 1998-2001)¹. Moreover, it is estimated that there are approximately 300 million Microsoft Office users worldwide ("WordPerfect," 2002). Within the productivity-software market segment, Microsoft's increasingly dominant share is exemplified by cessation of market-share tracking efforts since 2000 ("Microsoft," 2002).

THEORETICAL PERSPECTIVE

In his seminal work on decision-support systems (DSSs), Silver (1990) identified two conceptually distinct ways that computer-based DSSs may change decision-making processes—nondirected and directed. The nondirected view specifies that any direction of change in decision-making processes is determined solely by the decision maker and is, therefore, relatively independent of the computer-based DSS. In contrast, the directed view specifies that a DSS will force a direction of change in the decision-making process that may or may not be consistent with the decision maker's preferences. Under this view, while not entirely subjugated to the computer-based DSS, the human decision maker's discretion over decision making is limited or reduced by the computer-based DSS.

Where DSSs manifest directed change in decision-making processes, Silver (1990) identified *system restrictiveness* and *decisional guidance* as two system-level attributes that may possibly influence decision-making behavior. System restrictiveness is defined as "the degree to which and the manner in which a DSS limits its users' decision-making processes to a subset of all possible processes" (p. 52). Decisional guidance is defined as "the degree to which and the manner in which a DSS guides its users in constructing and executing decision-making processes, by assisting them in choosing and using its operators" (p. 57). In summary, DSSs may effect directed or nondirected change in decision-making behavior. Where the DSS is directive, it may either restrict or guide.

While these attributes are not entirely independent, Vessey, Jarvenpaa, and Tractinsky (1992) adopt them as two categories among three that together form a CASE tool classification framework. The third category conceptually overlaps with Silver's (1990) concept of nondirected change—"A flexible CASE tool is designed to allow the user complete freedom in using it" (p. 92). They

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/software-use-through-monadic-dyadic/163877

Related Content

The Role of Trainer Behavior in End User Software Training

Deborah Compeau (2002). *Journal of Organizational and End User Computing* (pp. 23-32).
www.irma-international.org/article/role-trainer-behavior-end-user/3746

A Framework for the Quality Evaluation of B2C M-Commerce Services

John Garofalakis, Antonia Stefaniand Vassilios Stefanis (2013). *Mobile and Handheld Computing Solutions for Organizations and End-Users* (pp. 130-147).
www.irma-international.org/chapter/framework-quality-evaluation-b2c-commerce/73210

A Methodology for Evaluating Tutorials for Computer Application Software

Richard Yellen (1989). *Journal of Microcomputer Systems Management* (pp. 12-24).
www.irma-international.org/article/methodology-evaluating-tutorials-computer-application/55654

Information Technology Supported Communication - Group Cohesion, Agreeability, and Performance: The Role of Media Richness

Michael B. Knightand D. Scott Hunsinger (2010). *Computational Advancements in End-User Technologies: Emerging Models and Frameworks* (pp. 242-259).
www.irma-international.org/chapter/information-technology-supported-communication-group/38096

Are Remote and Non-Remote Workers Different? Exploring the Impact of Trust, Work Experience and Connectivity on Performance Outcomes

D. Sandy Staples (2002). *Advanced Topics in End User Computing, Volume 1* (pp. 302-324).
www.irma-international.org/chapter/remote-non-remote-workers-different/4438