Research Report: The Evolving Relationship Between General and Specific Computer Self-Efficacy—An Empirical Assessment

Ritu Agarwal • V. Sambamurthy • Ralph M. Stair
Decision and Information Technologies, The Robert H. Smith School of Business, The University of Maryland, College Park, Maryland 20742-1815
Decision and Information Technologies, The Robert H. Smith School of Business, The University of Maryland, College Park, Maryland 20742-1815
College of Business, The Florida State University, Tallahassee, Florida 32306-1110
ragarwal@rhsmith.umd.edu • smurthy@rhsmith.umd.edu • rstair@garnet.acns.fsu.edu

The concept of computer self-efficacy (CSE) recently has been proposed as important to the study of individual behavior toward information technology. This paper extends current understanding about the concept of self-efficacy in the context of computer software. We describe how two broad types of computer self-efficacy beliefs, general self-efficacy and task-specific self-efficacy, are constructed across different computing tasks by suggesting that initial general CSE beliefs will strongly predict subsequent *specific* CSE beliefs. The theorized causal relationships illustrate the malleability and development of CSE beliefs over time, within a training environment where individuals are progressively provided with greater opportunity for hands-on experience and practice with different software. Consistent with the findings of prior research, judgments of self-efficacy then serve as key antecedents of the perceived cognitive effort (ease of use) associated with technology usage. Further, we theorize that self-efficacy judgments in the task domain of computing are strongly influenced by the extent to which individuals believe that they are personally innovative with respect to information technology. Panel data were collected using a longitudinal research design within a training context where 186 subjects were taught two software packages in a sequential manner over a 14-week period. The emergent patterns of the hypothesized relationships are examined using structural equation modeling techniques. Results largely support the relationships posited.
(*Computer Self-Efficacy; Technology Acceptance; Software Training; Longitudinal Study; Causal Model*)

1. Introduction

As information technology becomes more pervasive in organizational and personal lives, there is an increasing need for stimulating individuals with a wide variety of different predilections, experiences, aspirations, and goals toward utilizing the technology as an integral part of their existence. Recognizing the importance of this issue, existing academic and practitioner literature has drawn upon theories from diverse research streams, including social psychology, cognitive psychology, and the diffusion of innovations (Rogers 1995) to propose models of technology acceptance and use: the technology acceptance model (TAM)

INFORMATION SYSTEMS RESEARCH, © 2000 INFORMS
Vol. 11, No. 4, December 2000, pp. 418–430

1047-7047/00/1104/0418$05.00
1526-5536 electronic ISSN

(Davis et al. 1989), the theory of reasoned action (TRA) (Ajzen and Fishbein 1980), and the theory of planned behavior (TPB) (Ajzen and Madden 1986, Mathieson 1991). Although these models are different with respect to the specific relationships included, key similarities exist in their core constructs. In particular, all these models posit that individual beliefs or perceptions about, and attitudes toward, a new information technology (IT) are highly salient determinants of usage behavior.

Recent work in technology acceptance has called for the inclusion of an additional construct in the study of individual behavior toward new information technology—the concept of *self-efficacy* (Compeau and Higgins 1995a, b, Marakas et al. 1998). Emerging from a rich theoretical background in social learning and social cognition, self-efficacy refers to individuals' beliefs about their ability and motivation to perform specific tasks (Bandura 1977, 1986). In the domain of information technology in particular, studies of the effects of self-efficacy collectively point to its crucial role in determining individual behavior toward and performance using information technologies (Compeau and Higgins 1995b, Gist et al. 1989).

Surveying conceptual and empirical work related to the concept of self-efficacy, Marakas et al. (1998) draw a distinction between general computer self-efficacy and task-specific self-efficacy. General computer self-efficacy is defined as "an individual's judgment of efficacy across multiple computer application domains," whereas task-specific computer self-efficacy is defined as perceptions of ability to perform *specific* computer-related tasks in the domain of general computing. Although Marakas et al. (1998) made a compelling argument for the conceptual and operational distinction between these two constructs, empirical work examining the development of relationships between general computer self-efficacy and computer-specific self-efficacy is limited.

This paper extends current understanding about the concept of self-efficacy in the context of computer software. We draw upon the conceptual ideas of Marakas et al. (1998) to describe how the two types of computer self-efficacy (viz., general CSE and task-specific CSE) beliefs are constructed across different computing tasks. The theorized causal relationships illustrate the malleability and development of CSE beliefs over time within a training environment. Consistent with the findings of prior research, judgments of self-efficacy then serve as key antecedents of perceived cognitive effort (ease of use) associated with technology usage. Further, we extend the extant literature on CSE by showing that self-efficacy judgments in the task domain of computing are strongly influenced by the extent to which individuals believe that they are personally innovative with respect to information technology (Agarwal and Prasad 1998). Panel data collected via a longitudinal research design within a training context where 186 subjects were taught two software packages in a sequential manner are used to test the research model. The emergent patterns of the hypothesized relationships are examined using structural equation modeling techniques.

2. Conceptual Background

Prior Research in Computer Self-Efficacy

The concept of self-efficacy owes much of its conceptual development and empirical refinement to more than two decades of research by Bandura and his colleagues. Bandura postulates that self-efficacy beliefs are developed through four primary sources of information: "enactive mastery experiences that serve as indicators of capability; vicarious experiences that alter efficacy beliefs through transmission of competencies and comparisons with the attainments of others; verbal persuasion and allied types of social influence that one possesses certain capabilities; and physiological and affective states from which people judge their capableness, strength, and vulnerability to dysfunction" (Bandura 1997, p. 79). Of these, enactive mastery experiences, i.e., experiences gained through progressive trials (either success or failure) in a task domain, are considered to be the most potent and salient source of efficacy information. Central to Bandura's notion of self-efficacy is the idea that this personal belief is a major basis of action. The posited relationship between self-efficacy and behavior has been empirically validated in diverse domains such as education, health, and organizational life (see Bandura 1997).

Compeau and Higgins (1995a) define computer self-efficacy as the judgment of one's capability to use an information technology. They remark on the relative paucity of prior research examining the influence of self-efficacy in the context of computer training.[1] Table 1 provides a select review of published *empirical* research that explicitly incorporates self-efficacy in its conceptual and/or research models.[2] Although several articles incorporate more complex models and constructs than the ones listed, for conceptual simplicity, only the direct antecedents and consequences of CSE have been shown in this table.

As shown in Table 1, a wide variety of antecedents and consequences of self-efficacy have been studied using diverse research methodologies. Antecedents include *social influence* types of constructs, such as encouragement by others (Compeau and Higgins 1995b) and management support (Igbaria and Iivari 1995); *demographic variables*, such as computer experience (Igbaria and Iivari 1995, Henry and Stone 1994) and prior performance (Compeau and Higgins 1995a); and various individual *beliefs*, including self-conceptions of ability (Martocchio 1994). Similar variety exists in the consequences studied, which include *outcomes* such as actual performance (Compeau and Higgins 1995a, Gist et al. 1989), satisfaction (Henry and Stone 1994), and learning (Martocchio 1992); *beliefs*, such as affect, anxiety and outcome expectations (Compeau and Higgins 1995b), ease of use (Venkatesh and Davis, 1996), and perceived behavioral control (Taylor and Todd 1995a); and *behaviors*, including system use (Igbaria and Iivari 1995, Compeau and Higgins 1995b) and early adoption (Burkhardt and Brass 1990). Although a detailed discussion of the work listed in Table 1 is beyond the scope of this paper (for a more complete review of the literature, see Marakas et al. 1998), a few key findings are worth noting.

First, there is significant support for a relationship between CSE and individual beliefs about IT, particularly perceived ease of use (Venkatesh and Davis 1996) and outcome expectations (Compeau and Higgins 1995a, b). Second, the effects of different manipulations of *training* on computer self-efficacy have received significant attention. This focus is not surprising given the basic tenet of social learning theory: Beliefs about self-efficacy are modified and shaped through information collected and synthesized from four major sources. Arguably, a training context would contain one, if not all, of these sources. Third, individual beliefs about *themselves* can have an effect on self-efficacy beliefs. For example, in a software training context, Martocchio (1994) found moderating effects of individual conceptions of their own ability on the relationship between pre- and post-training self-efficacy beliefs. Finally, Table 1 suggests that prior research has examined the construct of CSE from two major perspectives: (1) as a malleable set of beliefs that can be manipulated through training and other interventions; and (2) as a dispositional individual difference quality that guides and circumscribes behavior with regard to new IT.

The Research Model

Figure 1 illustrates our research model. Each one of the constructs and relationships is discussed below.

We examine the development of self-efficacy beliefs over time within a software training environment. As noted above, researchers have often been interested in examining self-efficacy not as a static dispositional variable, but rather as a pliant set of beliefs that develop over time within a training context (Mathieu et al. 1993). A training context provides for the simultaneous existence of all four major sources of self-efficacy: opportunities for enactive mastery through hands-on trial and error; vicarious experience by watching others perform; verbal persuasion through performance feedback received from the instructor; and psychological and affective states such as stress and anxiety, which could be induced by required interaction with the computer. Although several studies have examined the effects of training manipulations on pre- and post-training judgments of computer self-efficacy, the elapsed time between assessments of CSE has been fairly limited. For instance, Martocchio and Webster's (1992) study assessed changes in self-efficacy after a four-hour training session. The same elapsed time for measurements appears in Martocchio

[1]The study of self-efficacy related to other task domains has a long standing tradition dating back to the 1970s (see Bandura 1997 for a comprehensive review). However, in this article we focus only on computer self-efficacy because, as underscored by Bandura, self-efficacy refers to a task- and context-specific set of beliefs; i.e., self-efficacy is a "particularized judgment of capability that may vary across realms of activity" (Bandura 1997, p. 42).

[2]Only articles published in journals are included.

Table 1 Select Empirical Research on Self-Efficacy in Computing and High Technology Domains

Authors	Antecedents of CSE	Consequences of CSE	Method and Sample	Findings
Venkatesh and Davis (1996)	None	Perceived ease of use	3 experiments with 108 student subjects; 6 different software packages	CSE → Perceived ease of use
Compeau and Higgins (1995a)	Prior performance Behavior modeling	Outcome expectations Performance	Experiment with 88 professional subjects; 2 training methods examined; Wordperfect and Lotus software	Prior performance → CSE (Day 2 only) Behavior modeling → CSE (Lotus only) CSE → Performance outcome expectations CSE → Personal outcome expectations CSE → Performance
Taylor and Todd (1995a)	None	Perceived behavioral control	Survey of students using a computing resource center	CSE → Perceived behavioral control
Igbaria and Iivari (1995)	Computer experience Organizational support	Computer anxiety Perceived ease of use Perceived usefulness Computer use	Survey of 450 users in Finnish businesses	Computer experience → CSE Organizational support → CSE CSE → Computer anxiety CSE → Perceived ease of use
Compeau and Higgins (1995b)	Encouragement by others Others' use Support	Outcome expectations Affect Anxiety Usage	Survey of 1,020 subscribers of a business periodical	Encouragement by others → CSE Support → CSE (negative) CSE → Outcome expectations—performance CSE → Affect CSE → Anxiety CSE → Usage
Henry and Stone (1994)	Management support Ease of system use Previous computer experience	End-user system satisfaction	Survey of 384 hospital staff members	Management support → CSE Ease of use → CSE Previous computer experience → CSE CSE → End-user system satisfaction
Martocchio (1994)	Pre-training CSE × Conceptions of ability Conceptions of ability	Acquisition of declarative knowledge	Experiment with 76 service and administrative employees in a university	Pre-training CSE × Conceptions of ability → Post-training CSE
Martocchio and Webster (1992)	Pre-training CSE Type of feedback	None	Field experiment with 68 university employees	Pre-training CSE → Post-training CSE Type of feedback → Post-training CSE
Webster and Martocchio (1992)	None	Computer playfulness	Survey of 77 employees participating in a university sponsored training program	CSE ↔ Computer playfulness (correlation)
Martocchio (1992)	Post-training computer anxiety (effects on Post-training CSE) Training context Pre-training expectations	Learning	Quasi experiment with 79 university employees	Training context → CSE CSE → Learning (negative) Pre-training expectations → CSE

Table 1 (continued) **Select Empirical Research on Self-Efficacy in Computing and High Technology Domains**

Authors	Antecedents of CSE	Consequences of CSE	Method and Sample	Findings
Ellen et al. (1991)	None	Resistance to change	Study 1: Experiment with 246 college students Study 2: Experiment with 86 undergraduate business students	CSE → Resistance to change (negative)
Burkhardt and Brass (1990)	None	Early adoption of a computer system	Longitudinal surveys of 81 employees	CSE → Early adoption
Gist et al. (1989)	Behavior modeling Pre-training CSE × Training method	Performance	Experiment with 108 managers and administrators in a university; 2 training methods examined; financial software program	Behavior modeling → CSE Pre-training CSE × Training method → CSE CSE → Performance
Hill et al. (1987)	None in study 1 Previous experience in study 2	Behavioral intentions to enroll in a computer course (study 1) Behavioral intentions to purchase a computer and/or enroll in a computer course (study 2)	Study 1: survey of 157 females and 147 male undergraduate students Study 2: survey of 133 female undergraduate students	CSE → Behavioral intentions (study 1 and study 2)
Hill et al. (1986)	Perceptions of product complexity in study 1 None in study 2	Liking of product (study 1) Intentions to try a new piece of software	Study 1: experiment with 83 undergraduate students Study 2: Experiment with 44 undergraduate students	Perceptions of product complexity → CSE CSE → Liking of product CSE × Source expertise (negative) → Intentions to try a new piece of software

Notes. The term CSE is used throughout for simplicity. Although the conceptual definition of computer self-efficacy is largely the same for all studies, various researchers have operationalized CSE in different ways, and at different levels of specificity.

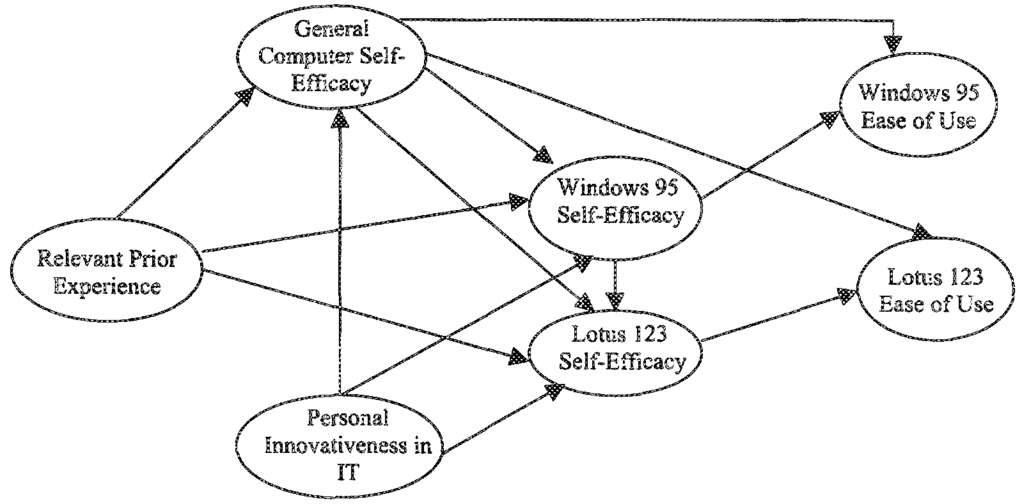
(1992), while Gist et al. (1989) examined the effects of a three-hour training session on self-efficacy. Because a fundamental notion underlying Bandura's conceptualization of self-efficacy is that this set of beliefs develops and crystallizes over time; it is clearly worthwhile to examine the longitudinal development of self-efficacy beliefs.

Consistent with recent literature (Marakas et al. 1998), our research model makes a distinction between the concepts of general CSE and software-specific self-efficacy (SSE). While CSE is a generalized individual trait, SSE refers to individual's feeling of self-efficacy relative to a specific software package. Thus, SSE is a "particularized judgment" as opposed to an omnibus feeling of efficacy related to the broad task of computing. Gist et al. (1989) found empirical evidence of a relationship between pre-training general computer self-efficacy and the specific software self-efficacy developed through training. Thus, within a training context, we would expect pre-training general CSE to be a predictor of SSE[3] developed through the training imparted.

However, individuals enter a training situation with varying degrees of prior experience in the particular activity domain. Indeed, prior experience has been an important individual difference variable utilized in past studies as a predictor of self-efficacy (Marakas et

[3]Although CSE refers to general beliefs about the capability to use new software, we use the term SSE to represent self-efficacy beliefs with regard to a *specific* materialization of software, such as the database product Access.

Figure 1 **The Research Model**



al. 1998). The fact that prior experience is relevant to a variety of beliefs about, and behaviors toward, information technologies is echoed in Taylor and Todd's (1995b) recent study. Because our primary focus is on isolating relationships between general and specific self-efficacy, the research model includes the construct of prior experience with computing as a control variable. Therefore,

Hypothesis 1. *After controlling for prior experience, pre-training general CSE beliefs will significantly influence posttraining SSE beliefs with a specific software.*

Bandura (1997) cautions against the use of general purpose or contextless conceptualizations and measures of self-efficacy. He also points out that although self-efficacy judgments are idiosyncratic to particular domains, perceived self-efficacy will tend to transfer across two tasks within the same domain based on the extent of similarity between the qualitative features of the two tasks and the skills they require (Bandura et al. 1977). For example, perceived efficacy in performing high-risk physical activity generalizes to other types of physical stressors (Brody et al. 1988). Within the domain of computing, it is reasonable to assume that familiarity with one piece of software should increase an individual's belief in their capability to use another software, i.e., specific software self-efficacy is likely to exhibit carryover effects. Thus, when training on multiple software packages is imparted sequentially, it is expected that prior self-efficacy with regard to general software, as well as specific software learned earlier, would have a cumulative effect on subsequent SSE assessments. The effects of CSE on subsequent assessments of SSE are captured in Hypothesis 1, whereas the carryover effects of SSE with the first software are tested through Hypothesis 2.

Hypothesis 2. *After controlling for prior experience and pre-training general CSE, post-training SSE beliefs developed after training on the first software package will significantly influence post-training SSE beliefs developed with the next software package in the training sequence.*

As argued earlier, individuals do not approach a new task devoid of any preconceptions about their ability to successfully perform the task. Although prior research has examined many factors that could serve as antecedents of CSE (see Table 1), our focus is on a recently described individual personality trait: personal innovativeness with information technology (PIIT) (Agarwal and Prasad 1998). The importance of examining cognitive factors is evident from Bandura's (1986) conceptualization of a triadic reciprocal causation in human agency, where (1) behaviors, (2) internal personal factors in the form of cognitive, affective, and biological events, and (3) the external environment interact to produce an interdependent causal structure.

Indeed, Bandura (1997) suggests that the individual personality exerts an *indirect* influence on performance via its effect on self-efficacy.

Personal innovativeness in the domain of information technology is defined as the "willingness of an individual to try out any new information technology" (Agarwal and Prasad 1998). As a relatively stable descriptor of individuals over time, this trait captures the risk-taking propensity of an individual with regard to information technology. Prior research, notably by Jones (1986), suggests that self-efficacy is related to an individual's openness to experience—an element underlying the conceptual definition of PIIT. Consistent with the arguments offered by cognitive psychologists (Gist et al. 1992), we suggest that personal innovativeness will influence self-efficacy beliefs. Individuals' conceptions about themselves as being more likely to voluntarily experiment with new technologies tend to create more frequent opportunities for enactive mastery. In other words, high levels of PIIT will result in more instances of technology use and experimentation, thereby providing for more occasions for trial-and-error learning. Thus, feelings of CSE in the general domain of computing as well as SSE, or the ability to interact with specific new software, will be greater.

Hypothesis 3. *After controlling for prior experience, personal innovativeness in the domain of information technology will significantly influence pre-training general CSE beliefs as well as post-training SSE beliefs with specific software.*

Although prior research has associated CSE with a variety of outcomes including learning, satisfaction, affect, anxiety, and usage, the primary objective of this research is to examine more fully the role played by CSE in the acceptance of new information technologies. Therefore, the key dependent variable in our model is an important belief about new technology—cognitive complexity beliefs, or perceived ease of use. This belief is central to the TAM and has received considerable theoretical and empirical support in prior work as a highly salient determinant of the acceptance of new information technology (see, for example, Tornatzky and Klein 1982, Davis et al. 1989, Moore and Benbasat 1991).

In recent work, Venkatesh and Davis (1996) found that general computer self-efficacy had a direct effect on ease of use perceptions, both before and after hands-on experience with software. Arguably, specific software self-efficacy is a more proximal predictor of ease of use because it refers to a focused, particularized judgment as opposed to a global feeling of confidence. Bandura (1997) notes that the optimal level of specificity at which self-efficacy should be assessed is a function of the specificity inherent in what one is seeking to predict. Therefore, SSE with a specific software should exhibit a significant effect on perceived ease of use of that software. In order to isolate the effects of SSE, we test Hypothesis 4.

Hypothesis 4. *After controlling for the effects of general CSE, SSE for a specific software package will significantly influence the perceived ease of use of that software package.*

Guthrie and Schwoeter (1994) recently observed that although a significant amount of research attention has been focused on self-efficacy, we still have only a limited understanding of its development. The research model tested here is an attempt to extend and replicate prior work in several ways. First, our model draws a theoretical and empirical distinction between notions of general self-efficacy and specific software self-efficacy. We posit relationships that unfold over a period of time within a software training environment and assess the carryover effects of specific software self-efficacy. Second, we draw upon cognitive psychology to posit a plausible antecedent of SSE that has yet to be subject to empirical scrutiny. Finally, we suggest that specific self-efficacy will influence perceived ease of use: a relationship that has been described in the context of general self-efficacy, but not specifically for a particularized judgment.

3. Methodology

Study Context and Sample

Data were gathered in the fall of 1996 and the spring of 1997 at a large university located in the southeastern United States as part of an introductory business computing concepts course. The course, taught by one of the authors, introduced Windows 95 and Lotus 1-2-3, with each being taught for approximately 7 weeks of a 14-week semester. Instruction involved lectures and hands-on lab experience, with the instructor present to respond to queries.

Data were gathered in three stages. At the beginning of the semester, the individual trait of PIIT was measured because PIIT is conceptualized as a relatively enduring and stable descriptor of individuals that is likely to be invariant in the short term. Data on pre-training general computer self-efficacy were also gathered at this point. Subsequently, students received instruction on the Windows 95 operating system for about seven weeks, at the end of which they submitted a major assignment that required them to utilize all the software's functionality. At the completion of this assignment, a questionnaire was used to gather data on software self-efficacy relative to the Windows 95 package and the perceived ease of use of the Windows 95 system. The strategy of anchoring data collection to the completion of assignments ensured the recency of experience with the software in the minds of the students; thus, this research design is equivalent to a natural field experiment. Finally, students received instruction on the Lotus 1-2-3 for Windows package for another seven weeks. Then, they completed another major assignment on this package. At this stage, data were again gathered on software self-efficacy relative to Lotus 1-2-3 and the perceived ease of use of this software package.

A total of 186 completed responses were obtained through this data gathering strategy. Sample demographics are provided in Table 2. Although respondents indicated that they had fairly extensive experience with personal computers (almost four years on average), they had considerably less experience with DOS and Windows 3.1. For the entire sample, the mean level of prior use of Windows 95 was just over five months, suggesting that the technology was still fairly new for the subjects. Subjects were randomly distributed across all the majors offered in the College of Business, and approximately three-quarters of the sample reported having some prior work experience.

Table 2 Sample Demographics

Variable	Mean	SD	Valid N
Prior PC use*	47.8	37.7	179
Prior use of DOS*	19.1	25.5	162
Prior use of Windows 3.1*	17.2	17.5	172
Prior use of Windows 95*	5.3	5.7	178
Software Usage Pattern in Last One Year^			
Spreadsheets	0.72	0.83	176
Wordprocessing	1.51	0.77	179
Databases	0.74	0.87	171
Electronic Mail	1.31	1.08	176
The Internet	1.33	0.97	178
College Major			182
Undeclared	3		
Finance	29		
Marketing	39		
Accounting	40		
Management	31		
Risk Management	4		
Hospitality Management	4		
Information Systems	32		
Work Experience			183
No	30		
Yes	153		

Notes. *Measured in number of months
^On a scale of 0–3: 0 = never, 1 = once a week, 2 = 3 to 10 times/week, 3 = >10 times/week

Operationalization of Research Constructs

All the research constructs were operationalized using multi-item scales previously developed and rigorously validated.[4] PIIT was measured through a 3-item scale based on the recommendations of Agarwal and Prasad (1998). CSE was operationalized using Compeau and Higgins' (1995a) 10-item scale. To develop measures for software self-efficacy constructs, the recommendations made by Bandura (1997) and Marakas et al. (1998) were closely followed. First, a set of tasks associated with Windows 95 and Lotus 1-2-3 were identified on the basis of an assessment of the key features and functionality of the specific packages. Finally, as discussed earlier, students were asked to complete a significant assignment that required them to use specific features of the two software packages; these features were included as items for measuring software self-efficacy. In all, nine items were used to construct measures of software self-efficacy with respect to both Windows 95 and Lotus 1-2-3.

Perceived ease of use was measured using the scale developed by Davis et al. (1989). The same set of items

[4]The instrument is available from the first author.

were used to capture these beliefs separately about the Windows 95 and the Lotus 1-2-3 packages at the second and third stages of data collection. Table 3 summarizes descriptive statistics including scale reliabilities and correlations between all research variables. All scales had a reliability (Cronbach's alpha) greater than 0.7, a level considered acceptable for field research (Nunnally 1978). A principal components factor analysis with varimax rotation on the items comprising all constructs extracted six factors that collectively explained 63.4% of the variance. A majority of the items loaded on the correct latent constructs, with the exception of two items related to Windows self-efficacy. These items were removed from subsequent analysis. Collectively, these data suggest that the scales used in this study exhibit appropriate convergent and discriminant validity and have good psychometric properties.

4. Results

Structural equation modeling using the partial least squares (PLS) method was utilized to test the hypothesized paths in the model (Figure 1). PLS, a latent structural equations modeling technique, uses a component-based approach to estimation. Therefore, it places minimal demands on sample size and residual distributions (Lohmoller 1989, Fornell and Bookstein 1982, Chin 1998). Paths are interpreted as standardized beta weights in a regression analysis. The path coefficients and explained variances for the model are shown in Figure 2.

Consistent with prior research, each multi-item construct except prior experience was modeled as a reflective latent variable (Chin 1998). As Chin notes, reflective indicators are affected by the same underlying latent variable, while formative indicators cause the creation of or change in the corresponding latent variable. Four items were used to measure prior experience: prior experience with Windows 95, prior experience with Windows 3.1, prior use of DOS, and prior use of personal computers. Because each one of these items can independently contribute to the overall intensity of prior experience, this construct was modeled as formative.

As stated in Hypothesis 1, after controlling for the effects of experience, general CSE significantly influenced SSE beliefs about the first software package in the training sequence, Windows 95. However, CSE did not have a significant influence on SSE beliefs about the second software package in the training sequence. As theorized in Hypothesis 2, significant support was found for the carryover effects of SSE: Beliefs about self-efficacy with Windows 95 (the first software) influenced self-efficacy beliefs about Lotus 1-2-3 (the second software).

Hypothesis 3 asserted that PIIT would influence both general and specific self-efficacy, after the effects of prior experience has been factored out. PIIT had a strong, positive relationship with general CSE as well as Windows 95 CSE, but not with SSE for Lotus 1-2-3,

Table 3 Descriptive Statistics

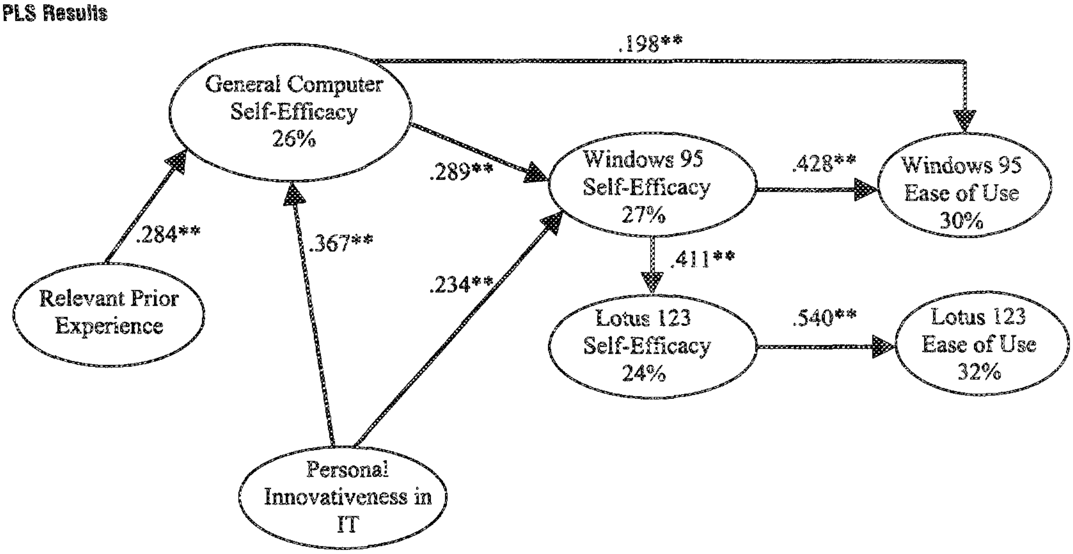
				Correlation^					
Construct	Reliability# (Number of items)	Mean*	SD	PIIT	CSE	WSE	WEOU	LSE	LEOU
Personal innovativeness with IT (PIIT)	0.74 (3)	4.93	1.20	1.00					
General computer self-efficacy (CSE)	0.92 (10)	6.90	1.86	0.43	1.00				
Windows 95 self-efficacy (WSE)	0.80 (7)	8.63	1.33	0.40	0.45	1.00			
Windows 95 ease of use (WEOU)	0.93 (4)	5.20	0.96	0.43	0.37	0.47	1.00		
Lotus 123 self-efficacy (LSE)	0.89 (9)	8.13	1.48	0.26	0.31	0.48	0.34	1.00	
Lotus 123 ease of use (LEOU)	0.92 (4)	4.65	1.03	0.19	0.19	0.22	0.25	0.54	1.00

Notes. # Adjusted Cronbach alpha is reported

* CSE, WSE, and LSE are measured on 10-point scales. All other constructs are 7-point scales.

^ Pearson correlation coefficients are reported. Coefficients >0.18 are significant at $p < 0.01$; >0.15 are significant at $p < 0.05$

Figure 2 PLS Results



Notes.
1. Relevant prior experience is measured using formative indicators, while all other constructs are modeled reflectively.
2. Compared to Figure 1, only significant paths are displayed along with the magnitude of the path coefficients. R^2 values are displayed below construct labels.
3. ** $p<0.05$

thereby providing partial support for Hypothesis 3. Finally, Hypothesis 4 is strongly supported for both packages: After controlling for general CSE beliefs, specific SSE for each software package had a highly significant effect on the perceived ease of use of that software package.

5. Discussion and Conclusions

The goal of this research was to empirically extend our insights into the unfolding effects of self-efficacy in a training process where individuals sequentially learn how to use two different software packages. In particular, we provided an empirical test of the differential effects of two alternate forms of self-efficacy: computer self-efficacy, which is a generalized individual belief about the ability to use information technology; and software self-efficacy, which is a particularized individual belief about the ability to use a specific information technology.

Prior to discussing the implications of the results, limitations that circumscribe their interpretation must be acknowledged. With regard to the external validity of this study, although the subjects did not constitute a random sample, there is no reason to believe that the sample was biased in any particular direction regarding their prior background and predilections. Further, the use of student subjects limits the generalizability of the results to similar populations. Finally, we postulated that specific self-efficacy was a predictor of perceived ease of use; however, the data for both these constructs were gathered at the same point in time. Thus, there might be a potential for common method variance, although such a concurrent measurement strategy is consistent with several other studies that have examined identical constructs (e.g., Venkatesh and Davis 1996, Compeau and Higgins 1995a).

Our study demonstrates that while CSE beliefs have a significant influence on SSE beliefs about software applications introduced earlier in the training sequence, the effects of such pre-training CSE diminish on SSE assessments about other software applications subsequent in the training sequence. In fact, the SSE beliefs developed with the earlier software package tend to significantly influence SSE beliefs with subsequent software packages, a phenomenon we term "carryover" effects. Not only do SSE beliefs developed with earlier software influence SSE beliefs about the subsequent software, but also the emerging SSE beliefs become the most significant predictors of beliefs about ease of use of the subsequent software. That is, with the passage of time, the effects on ease of use of a more focused confidence in abilities dominate generalized perceptions of abilities. Therefore, the accumulated SSE beliefs with different software packages begin to displace the effects of CSE beliefs that the individuals possess at the beginning of the training sequence.

Our study also examined the influence of prior experience and personal innovativeness with information technologies on beliefs about CSE and SSE with the two software packages. Although we used prior experience as a control, the findings are consistent with a majority of prior research that has empirically demonstrated that an individual's past interactions with the target performance behavior help shape self-efficacy beliefs. Interestingly, prior experience had a significant effect only on CSE, and not on SSE, for either software package. Intuitively, one would expect to observe such a result only when all the aspects of experience examined referred to broad-based, as opposed to specific, computing tasks. However, we also included experience with one of the to-be-learned systems, i.e., Windows 95 in the measurement of prior experience. One plausible explanation for these findings might be that compared with the magnitude of other experiences (PC, DOS, and Windows 3.1), the sample's experience with Windows 95 was relatively low. Therefore, such experience only contributed to amplifying a global as opposed to a particularized, specific confidence in abilities.

An individual's propensity to try out a new information technology, PIIT, was significantly related to CSE beliefs and SSE beliefs for Windows 95, but not to SSE beliefs about Lotus 1-2-3. A possible interpretation of these results is that they might be idiosyncratic to the specific software examined in this study: Windows 95 and Lotus 1-2-3. While the former represents a computing environment, the latter is application software. Note, however, that Agarwal and Prasad (1998) theorized that PIIT is reflective of innovativeness with any new type of information technology. The nonsignificant association between PIIT and SSE for Lotus 1-2-3 observed here is suggestive of either a need to further refine the conceptualization of PIIT, or of replication with additional studies to ascertain if a similar pattern of effects is observed.

Several important implications for research and practice follow. From the perspective of theory development, although we found evidence for personal innovativeness and general computer self-efficacy as two individual traits that significantly influence subsequent judgments of specific software self-efficacy, these two predictors explain just over one quarter of the variance in SSE. Therefore, there is a need for further investigation of other determinants. Moreover, the effects of these two traits dissipate over time because neither was a significant direct influence on SSE developed with the second software package in the training sequence. Given that our study assessed individual traits at the advent of training, these findings allow us to offer some practical recommendations. In essence, we have characterized individuals who, a priori, are likely to feel more confident about their ability to use new software after being trained in using it. Managers desirous of implementing new IT may wish to target individuals who exhibit greater personal innovativeness and general computer self-efficacy because these individual traits will increase the likelihood of their technology acceptance.

A second interesting implication of our study, revealed again by its longitudinal design, is related to the design of training programs. Carry-over effects underscore the need for designing training programs where successive reinforcing opportunities are provided for building skills. Indeed, the training literature points out that a key objective of training programs is to assist trainees in developing a mental model into which learned concepts can be situated (Davis and Bostrom 1993). The sequence of training provided—with the platform (Windows 95) taught first—may have helped to ensure that trainees were able to construct such a mental model. As noted by Gist (1987), mastery is facilitated when gradual accomplishments help build skills. Research examining the development and transfer of skills in several other domains such as user interfaces (Whiteside et al. 1985), text editors (Singley and Anderson 1985, 1988), and programming languages (Scholtz and Weidenbeck 1990) has shown that positive transfer effects will be obtained only when successive to-be-learned technologies share common elements. In this case, the common elements include a similar look and feel across the two pieces of software. Although we did not manipulate training sequence in our study, these prior findings when juxtaposed with our results allow us to offer the following recommendation. In organizational environments where it is necessary to train potential users on a variety of different software packages, training plans may need to account for not only the specific content of the training delivered, but also the sequence in which such packages are introduced.

Several avenues for future work remain. To further extend the external validity of our findings, the model proposed here needs to be tested in a variety of contexts with a range of technologies. It would also be interesting to study the relative effects of general versus specific software efficacy when users are successively trained to use packages that are dissimilar in their functionalities and appearance, such as a GUI-based system followed by a command-line interface. Finally, in our study we did not specifically assess what source of self-efficacy information was more salient in the construction of self-efficacy estimates. Others might wish to examine, in the spirit of the study reported by Compeau and Higgins (1995b), how alternate sources of efficacy information crystallize into self-efficacy beliefs over time.

References

Agarwal, R., J. Prasad. 1998. A conceptual and operational definition of personal innovativeness in the domain of information technology. *Inform. Systems Res.* 9(2) 204–215.

Ajzen, I., M. Fishbein. 1980. *Understanding Attitudes and Predicting Social Behavior.* Prentice-Hall, Englewood Cliffs, NJ.

——, T. J. Madden. 1986. Prediction of goal-directed behavior: attitudes, intentions and perceived behavioral control. *J. Experiment. Social Psych.* 22(5) 453–474.

——. 1977. Self-efficacy: Toward a unifying theory of behavioral change. *Psych. Rev.* 84(2) 191–215.

——, N. E. Adams, J. Beyer. 1977. Cognitive processes mediating behavioral change. *J. Personality and Social Psych.* 35 125–139.

——. 1986. *Social Foundations of Thought and Action: A Social Cognitive Theory.* Prentice-Hall, Englewood Cliffs, NJ.

——. 1997. *Self-Efficacy: The Exercise of Control.* W. H. Freeman, New York.

Brody, E. B., B. D. Hatfield, T. W. Spalding. 1988. Generalization of self-efficacy to a continuum of stressors upon mastery of a high-risk sport skill. *J. Sport and Exercise Psych.* 10 32–44.

Burkhardt, M. E., D. J. Brass. 1990. Changing patterns or patterns of change: The effects of change in technology on social network structure and power. *Admin. Sci. Quart.* 35 104–127.

Chin, W. 1988. Issues and opinion on structural equation modeling. *MIS Quart.* 22(1) vii–xvi.

Compeau, D. R., C. A. Higgins. 1995a. Computer self-efficacy: Development of a measure and initial test. *MIS Quart.* 19 189–211.

——, ——. 1995b. Application of social cognitive theory to training for computer skills. *Inform. Systems Res.* 6(2) 118–143.

Davis, S. A., R. P. Bostrom. 1993. Training end users: An experimental investigation of the roles of the computer interface and training methods. *MIS Quart.* 17 61–85.

Davis, F. D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quart.* 13(3) 318–339.

——, R. P. Bagozzi, P. R. Warshaw. 1989. User acceptance of computer technology: A comparison of two theoretical models. *Management Sci.* 35(8) 982–1003.

Ellen, P. S., W. O. Bearden, S. Sharma. 1991. Resistance to technological innovations: An examination of the role of self-efficacy and performance satisfaction. *J. Acad. Marketing Sci.* 19(4) 297–307.

Fornell, C., F. Bookstein. 1982. Two structural equation models: LISREL and PLS applied to consumer exit-voice theory. *J. Marketing Res.* 19 440–452.

Gist, M. 1987. Self-efficacy: implications for organizational behavior and human resource management. *Acad. Management Rev.* 12(3) 472–485.

——, T. R. Mitchell. 1992. Self-efficacy: A theoretical analysis of its determinants and malleability. *Acad. Management Rev.* 17(2) 183–211.

——, C. Schwoerer, B. Rosen. 1989. Effects of alternative training methods on self-efficacy and performance in computer software training. *J. Appl. Psych.* 74(6) 884–891.

Guthrie, J. P., C. E. Schwoerer. 1994. Individual and contextual influences on self-assessed training needs. *J. Organ. Behavior* 15 405–422.

Henry, J. W., R. W. Stone. 1994. A structural equation model of end-user satisfaction with a computer-based medical information system. *Inform. Resources Management J.* 7(3) 21–33.

Hill, T., N. D. Smith, M. F. Mann. 1986. Communicating innovations: convincing computer phobics to adopt innovative technologies. R. J. Lutz, ed. *Advances in Consumer Research.* Association for Consumer Research, Provo, UT, 419–422.

——, ——, ——. 1987. Role of efficacy expectations in predicting the decision to use advanced technologies: The case of computers. *J. Appl. Psych.* 72(2) 307–313.

Igbaria, M., J. Iivari. 1995. The effects of self-efficacy on computer usage. *Omega* 23(6) 587–605.

Jones, G. R. 1986. Socialization tactics, self-efficacy, and newcomers' adjustments to organizations. *Acad. Management J.* 29 262–279.

Lohmoller, J. 1989. *Latent Variable Path Modelling with Partial Least Squares.* Physica-Verlag, Heidelberg, Germany

Marakas, G., M. Yi, R. Johnson. 1998. The multilevel and multifaceted character of computer self-efficacy: Toward clarification of the construct and an integrative framework for research. *Inform. Systems Res.* **9**(2) 126–163.

Martocchio, J. J. 1992. Microcomputer usage as opportunity: The influence of context in employee training. *Personnel Psych.* **45** 529–551.

——. 1994. Effects of conceptions of ability on anxiety, self-efficacy, and learning in training. *J. Appl. Psych.* **79**(6) 819–825.

——, J. Webster. 1992. Effects of feedback and cognitive playfulness on performance in microcomputer software training. *Personnel Psych.* **45** 553–578.

Mathieson, K. 1991. Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Inform. Systems Res.* **2**(3) 173–191.

Mathieu, J. E., J. W. Martineau, S. I. Tannenbaum. 1993. Individual and situational influences on the development of self-efficacy: Implications for training effectiveness. *Personnel Psych.* **46** 125–147.

Moore, G. C., I. Benbasat. 1991. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Inform. Systems Res.* **2**(3) 192–222.

Nunnally, J. C. 1978. *Psychometric Theory*. McGraw Hill, New York.

Rogers, E. M. 1995. *Diffusion of Innovations*, 4th ed. The Free Press, New York.

Scholtz, J., S. Weidenbeck. 1990. Learning second and subsequent programming languages: A problem of transfer. *International J. Human-Comput. Interact.* **2** 51–72.

Singley, M. K., J. R. Anderson. 1985. The transfer of text-editing skill. *Internat. J. Man-Machine Stud.* **22** 403–423.

——, ——. 1988. A keystroke analysis of learning and transfer in text editing. *Human-Comput. Interact.* **3** 223–274.

Taylor, S., P. A. Todd. 1995a. Understanding information technology usage: A test of competing models. *Inform. Systems Res.* **6**(2) 144–176.

——, ——. 1995b. Assessing IT usage: The role of prior experience. *MIS Quart.* **19**(4) 561–570.

Tornatzky, L. G., K. J. Klein. 1982. Innovation characteristics and innovation adoption-implementation: A meta-analysis of findings. *IEEE Trans. Engrng. Management* **EM-29** 28–45.

Venkatesh, V., F. D. Davis. 1996. A model of the antecedents of perceived ease of use: Development and test. *Decision Sci.* **27**(3) 451–481.

Webster, J., J. J. Martochhio. 1992. Microcomputer playfulness: Development of a measure with workplace implications. *MIS Quart.* **16**(2) 201–226.

Whiteside, J., S. Jones, P. S. Levy, D. Wixon. 1985. User performance with command, menu, and iconic interfaces. *CHI'85 Proc.: Human Factors in Comput. Systems*. San Francisco, CA 185–191.

Peter Todd, Associate Editor. This paper was received on August 10, 1998, and was with the authors 5 months for 2 revisions.