Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2008 Proceedings

European Conference on Information Systems (ECIS)

2008

Using Levels of Difficulty to Judge Computer Ability: An Empirical Comparison of the Structure of Self-Efficacy

J Downey
University of Central Arkansas, jdowney@uca.edu

Summer E. Bartczak
University of Central Arkansas, sbartczak@uca.edu

USING LEVELS OF DIFFICULTY TO JUDGE COMPUTER ABILITY: AN EMPIRICAL COMPARISON OF THE STRUCTURE OF SELF-EFFICACY

Downey, James, University of Central Arkansas, 201 Donaghey St., Burdick Business Building, Conway, AR., 72034, USA, jdowney@uca.edu

Bartczak, Summer, University of Central Arkansas, 201 Donaghey St., Burdick Business Building, Conway, AR., 72034, USA, sbartczak@uca.edu

Abstract

Although self-efficacy (SE) and computer self-efficacy (CSE) in particular has received much research interest in recent years, organizations have been deterred from using it because of the confusing number of extant instruments and the lack of publicity by researchers. CSE is an extremely useful construct in that it provides not only a measure of competence, but more importantly it is an indication of employee motivation to learn and use technology. Two problems have exacerbated this condition: a misunderstanding of how individuals actually judge their computing efficacy and conceptual confusion regarding task versus domain SE. While most extant CSE instruments include the structural dimension of magnitude (degree of task difficulty), this study questions this inclusion both conceptually and empirically. We compare four CSE measures (one without magnitude), in three computing domains, in their relationship with four common outcomes of CSE. Using a sample of over 300 and structural equation modeling, the findings are surprising in that for three outcomes, the CSE measure which did not include magnitude was significantly better. This suggests that perhaps magnitude is not as important as cognitive researchers have conjectured and further clarifies how individuals judge their computing ability. Other implications and directions for further study are presented.

Keywords: Computer self-efficacy; social cognitive theory; self-efficacy magnitude; computer anxiety; computer affect; computing skills; structural equation modeling.

1 INTRODUCTION

There has been increasing interest and research effort in the area of self-efficacy (SE) and in particular computer self-efficacy (CSE). SE serves as a motivator of action and can be influenced and enhanced by both training (Yi & Davis, 2003) and experience (Igbaria & Iivari, 1995). Self-efficacy stems from the work of Bandura and his Social Cognitive Theory (1986, 1997). According to this theory, grounded by impressive empirical research (cited throughout this study), human behavior is predictable and reciprocally influenced by both environmental and cognitive factors. Self-efficacy, one of the cognitive factors, is an individual's confidence that he or she can successfully accomplish a given task. Bandura maintains that self-efficacy beliefs are not merely "passive foretellers" of one's ability level (1997, p. 39); they also help govern and stimulate the motivation necessary to conduct the behavior. Self-efficacy helps determine what activities an individual engages in, the effort in pursuing that activity, and persistence in the face of adversity.

Computer self-efficacy is defined as an individual judgment of one's capability to use a computer (Compeau & Higgins, 1995a). It is a multileveled construct; that is, one can judge their computing ability at a specific level (for example, spreadsheet or database CSE) and at a general, or computer-wide level, referred to as general CSE or GCSE (Marakas, Yi & Johnson, 1998).

The issue of how to best measure an individual's perception of ability (i.e., their SE) for some task has been addressed extensively by cognitive theorists (Bandura, 1986, 1997; Compeau & Higgins, 1995a, 1995b; Gist, 1987; Gist & Mitchell, 1992; Marakas et al., 1998). The issue will be covered in the next section, as this study examines one of the fundamental assertions by theorists. First it is important to provide the motivation for this study. Self-efficacy, and CSE in particular, remains one of the strongest predictors of critical outcomes, such as computer performance (Colquitt, LePine, & Noe, 2000; Munro, Huff, Marcolin & Compeau, 1997), computer skills development (Martocchio & Webster, 1992), attitudes or beliefs with respect to computing, including anxiety (Hasan, 2006) and computer liking or affect (Rainer & Harrison, 1993), and usage (Hung, 2003). Whether a person's perception of their ability is accurate or not, self-efficacy still provides a motivational component that enhances effort in learning and performing computing tasks and persistence despite obstacles (Murphy, Coover, & Owens, 1989).

This is extremely important to organizations, because CSE is an indicator of future potential and one's ability to prosper on the job using technology, particularly with respect to training (Compeau & Higgins, 1995b). It is an indicator not only of competence, but of motivation; as such, it is can be more useful than an actual performance test which merely yields an individual's level of competence. The issue of how best to measure CSE is compounded by the multi-leveled nature of self-efficacy. CSE instruments typically measure some specific application domain (e.g., spreadsheets) or the entire computing domain. The former instruments, generally referred to as "application-specific CSE" (labeled AS-CSE), while usually effective in measuring CSE for a particular domain, have limited applicability. Instruments that measure self-efficacy for the entire computing domain are much more useful to organizations in that they have wider applicability. One's self-efficacy for the "entire computing domain" includes all applications and environments (Marakas et al., 1998).

Besides the domain measured, extant CSE instruments also differ in structure. Most (but not all) instruments include domain tasks with apparent differences in difficulty levels. Labeled magnitude (Gist & Mitchell, 1992), it refers to the difficulty level of the task; those with higher SE believe they can accomplish more difficult tasks. Based on the seminal conceptualization of Bandura (1986, 1997), most CSE instruments include magnitude; at issue is whether this inclusion actually adds value in computing.

This is important for two reasons. First, understanding the conceptual underpinnings of CSE is important for both practitioners and researchers. If individuals do not need to be presented computer task difficulty levels for a valid measure of their efficacy (despite current thinking), this would be an important contribution to understanding how individuals judge their computing ability. Secondly, we believe CSE is not used frequently by business organizations because there are too many instruments, each measuring different computing domains, with varying results, making it difficult for IT researchers to make the case for its importance to business managers. A simple, short, valid instrument that has high predictive power in explaining important outcomes would be extremely useful to organizations.

There is no known study which empirically tests the CSE structural issue of magnitude. Most studies follow the recommendation of cognitive theorists (Bandura, 1997; Gist & Mitchell, 1992; Marakas et al., 1998) and include this dimension in their operationalization of CSE. We question this inclusion first on a conceptual basis, and then empirically. To carry out this study, respondents' self-efficacy was assessed in three different domains (word processing-WP, spreadsheets-SS, and the entire computing domain), using four different instruments (one of these instruments did not include magnitude). Each resulting self-efficacy measure was then examined in its relationship with four outcomes established in the literature, including two attitudes and two performance-based measures. By comparing the strength of each relationship (using structural equation modeling), the inclusion of magnitude was directly tested.

2 MAGNITUDE AND THE STRUCTURE OF CSE

Self-efficacy has three distinct but related dimensions, including strength, generality, and magnitude.

2.1 Strength and Generality

Strength is an assessment of confidence in successfully completing a given task. It is the level of conviction about the self-efficacy judgment. Those with higher self-efficacy have more confidence in their ability to carry out the task. To operationalize strength, instruments generally use a ten point response scale that ranges from 1 to 10, as recommended (Bandura, 1997; Gist & Mitchell, 1992). Generality refers to the extent that a self-efficacy judgment in one area applies (generalizes) to a judgment in another area. Bandura recognized that SE judgments are rarely made from scratch; rather they are formed based on a variety of previous experiences (1997).

2.2 Magnitude

As stated, magnitude refers to the difficulty level of a given task an individual believes he/she can accomplish. Magnitude is traditionally measured dichotomously (Bandura, 1986; Gist & Mitchell, 1992), that is, by a yes/no as to whether or not the respondent could complete the task.

Bandura conceptualized tasks as having levels of difficulty. Using his example of individual's with a snake phobia (1986), there is a progression of difficulty levels that include looking at a snake, being close to one, touching one, and finally holding a snake. He did additional studies with individuals afraid of spiders (Bandura, Reese, & Adams, 1982). Because these types of tasks included inherent difficulty levels, it made intuitive sense to include magnitude in the structure of self-efficacy.

In the IT field, self-efficacy is typically measured for some domain, including an application (such as databases), an environment (such as UNIX), or the entire computing domain (GCSE). Task difficulty levels (i.e., magnitude) are included in extant instruments two ways. First, some instruments include actual tasks; the spreadsheet instrument of Johnson and Marakas (2000) includes ten separate tasks. Most application CSEs are of this type. The other type of instrument uses levels of assistance, such as the GCSE instrument of Compeau and (1995a). This instrument asks respondents if they could carry out an unspecified job using an unknown software application with ten levels of assistance, such as "if I could call someone if I got stuck". These levels of assistance provide levels of difficulty for a domain. This instrument is used predominately for the entire computing domain, though it has been used for application domains, including spreadsheets and word processing (Compeau & Higgins, 1995b).

We present the case of not including the dimension of magnitude based on two arguments. First, computing tasks are different than the tasks Bandura used. Most computing tasks do not really have intrinsic difficulty levels. For example, in the spreadsheet CSE of Johnson and Marakas (2000), one task concerns the ability to summarize numeric information. There aren't really levels of difficulty involved; it takes the same ability level to summarize two or five numbers (or integers versus decimal numbers, etc.). In general, most computing tasks are singular, without difficulty levels. Therefore computing tasks are intrinsically different than the tasks used by Bandura. What has confused the issue of magnitude is task versus domain. If computing tasks are essentially without difficulty levels, domains, such as spreadsheets or databases, appear to have difficulty levels. In the computing field, magnitude has come to stand for domain difficulty levels rather than task difficulty levels. This conceptual leap has largely been unexamined and untested.

Secondly, we argue that while computer applications do contain a variety of discrete tasks, these individual tasks do not usually have an absolute range of difficulty. Consider the spreadsheet CSE instrument of Johnson and Marakas (2000; see Appendix A). Tasks include formatting numbers in a spreadsheet, summarizing numeric information, using a simple formula, and making a graph. Are any of these intrinsically more difficult than the others? There is a relative ordering of difficulty, which depends on each individual. We recognize that domains do have task difficulty levels (putting a number in a cell is inherently easier than writing a formula), but most task-based CSE instruments measure beginning (or intermediate) skills in an application, where there is no absolute ordering. Even if we were to concede that domain versus task difficulty levels is an acceptable transition (issue one above), in computing domains CSE measures are actually an aggregation of discrete tasks making up the domain that do not have absolute difficulty levels.

For these reasons, we set out to test whether including task difficulty levels in domains adds any worthwhile explanatory predictive power. While most CSE instruments include magnitude in a domain (either by different tasks or by levels of assistance), one type of instrument does not. A global CSE instrument asks respondents to judge their ability in a domain without referring explicitly to differences in difficulty. One example is the global instrument of Hill, Smith & Mann (1987), a four-item instrument which focuses on understanding and using computers, but has no task differentiation. We point out that global instruments are judgments of ability for a particular domain (in this case the computing domain), and are not the global (or omnibus) instrument that Bandura (1986, 1997) derides, which has no context (domain) at all. Although a global instrument conceivably could be used for any domain, they have typically been used for the full computing domain (one exception is Vijayasarathy, 2004, which included a global measure for Internet shopping technology). Global measures have had some success in the computing field (e.g., Hill et al., 1987; Igbaria & Iivari, 1995; Vijayasarathy, 2004).

3 OUTCOMES OF CSE AND RESEARCH MODEL

Although this study empirically compares four CSE instruments in their relationship with four outcomes, we provide only minimal justification for using them. There is extensive extant literature documenting these relationships (for a summary, see Marakas et al., 1998). Our intent is to compare the strength of different CSE measures, not to provide a detailed theoretical exposition of the relationships.

3.1 Outcomes of CSE

Attitudes are dynamic, domain-specific individual differences that affect the conduct of the individual's activities within the domain (Thatcher & Perrewe, 2002). Both theory and research suggests that there is a statistically significant relationship between CSE and computer attitudes. Computing anxiety is a fear of computers or of computer use (Loyd & Gressard, 1984). Self-efficacy influences how individuals interpret their experiences, which influences anxiety and other emotions (Bandura, 1997). Studies show that persons with high CSE have less anxiety, while those with low CSE exhibit higher anxiety (Johnson & Marakas, 2000; Hasan, 2006). Affect is the feeling of like or dislike towards computing. A person's like/dislike towards computing is an important factor in user acceptance as well as computer usage (Al-Jabri & Al-Khaldi, 1997). Individuals tend to pursue activities they like while avoiding disliked activities (Loyd & Gressard, 1984). Like anxiety, affect has a statistically significant relationship with CSE. CSE has been shown to influence an individual's positive affect toward technology (Rainer & Harrison, 1993). It should be noted that affect (dislike) is a different construct than anxiety (Kernan & Howard, 1990).

The relationship between self-efficacy and performance is one of the strongest in the literature. Individuals with higher self-efficacy tend to perform better at tasks in question and have higher competence (Bandura, 1997; Munro et al., 1997). In a recent meta-analytic study, the correlation between self-efficacy and performance was .40 (Colquitt et al., 2000).

3.2 Research Model

Self-efficacy is predicted to significantly influence two attitudinal constructs, anxiety and affect, as well as performance (in the word processing and spreadsheet domains). Self-efficacy is measured four ways; two are application-specific instruments for word processing and spreadsheets. The other two measure SE for the full computing domain; one is the popular GCSE (Compeau & Higgins, 1995a) and the other is a global instrument, which does not include magnitude. The model used in this study is presented in Figure 1.

Another issue that must be considered is the potential mismatch between the level of the self-efficacy measure and related constructs. Labeled "specificity matching", it is the presumption that predictability is greatest when the level of self-efficacy (specific to general) matches the level of the construct to which it is being related (Chen et al., 2001, p. 65; see also Ajzen, 1991; Bandura, 1997). Therefore, an AS-CSE should have a stronger relationship with application performance (both at the specific level), etc. While this may impact study results, it is also true that SE "generalizes" to similar domains (Bandura, 1997), so mitigating this effect in the IT field.

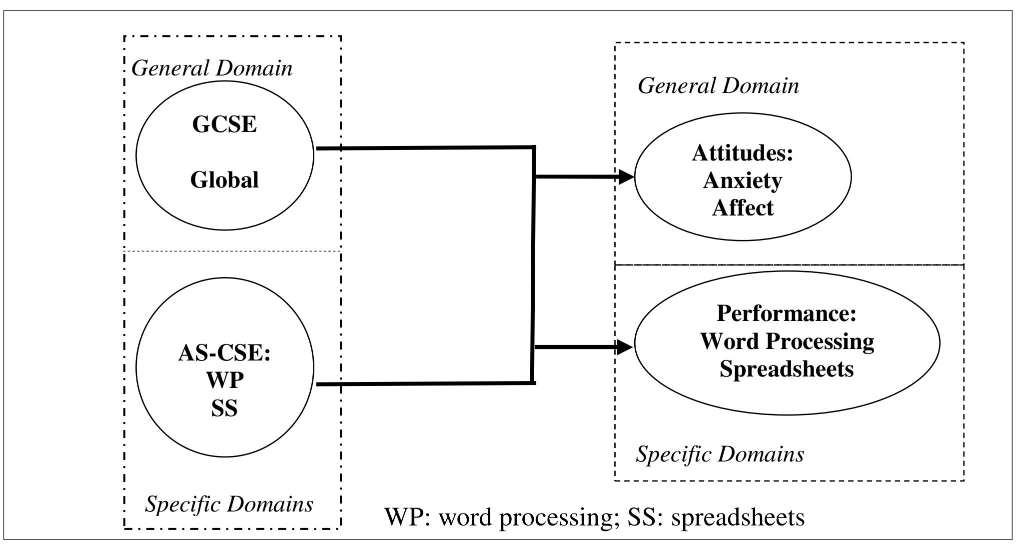

Figure 1. Theoretical Model

4 METHODOLOGY

The population chosen for this study is Midshipmen in the U.S. Navy's commissioning program. There are 57 US universities that currently have a Naval Reserve Officers Training Corps (NROTC) program as well as the US Naval Academy, with Midshipmen in the process of earning college degrees and receiving commissions in the Navy or Marine Corps. Thirteen universities with NROTC programs were chosen at random (from the 57 total) to participate. Because of its size, the US Naval Academy was also selected. Each university was sent 24 surveys, while the Naval Academy received 61. Of the 373 surveys sent, 308 completed responses were received for an overall response rate of 83%. Universities included South Florida, Florida, South Carolina, Missouri, Minnesota, Kansas, Penn State, Idaho, Ohio State, Washington, Purdue, Oregon State, and Vanderbilt.

The average age of respondents was 21.1 (sd = 2.91); 263 were male (86%) and 45 were female. On average, responders had 2.4 years of college (sd = .99). Respondents were evenly spread from freshman through seniors. Respondents' majors included liberal arts (30%), engineering majors (not computer science) (19%), IT/MIS (14%), math or science (12%), business but not MIS (12%), computer science (6%), with the rest a mixture of agriculture, education, health/nursing, and others (7%).

Study Measures: Most measures were derived from previously reported and validated instruments.

Attitudes. Anxiety and affect was measured using the anxiety and computer liking subscales of the Computer Attitude Scale developed by Loyd and Gressard (1984). Woodrow (1991) stated that the subscales were reliable enough to be administered separately.

Performance Tests. Two performance tests of fourteen questions each were developed for this study, for word processing and spreadsheets (Word and Excel). Both consisted of declarative knowledge items in a multiple choice format. In order to avoid a potential confound for those who may be skilled at a non-Microsoft application, we used only those who indicated they knew Word or Excel "best".

Self-efficacy Measures. There were four self-efficacy scales in the survey. For the two AS-CSE scales, word processing and spreadsheets, all of the survey items were actual tasks. All start with the same stem, "I believe I have the ability to…", followed by the actual task within the domain. Following the recommendation of Lee and Bobko (1994), each CSE score was derived from averaging the strength of only those tasks that the respondent believed they could accomplish.

The spreadsheet CSE instrument was developed by Johnson and Marakas (2000) and validated by the authors and others (Yi & Davis, 2003). The word processing instrument was self-developed, but similar in scope and design to the spreadsheet scale.

GCSE. GCSE, or efficacy for the entire computing domain, was measured using the GCSE instrument of Compeau and Higgins (1995a). It included magnitude and strength and the same score derivation.

Global self-efficacy. Adapted from an instrument by Hill et al. (1987), this two-item scale also measured efficacy for the entire computing domain. It asks respondents to rate their confidence in their "computer ability" and their ability to "learn computer applications". Scored like the other scales, it does not include difficulty levels and has no magnitude. See Appendix A for selected CSE instruments.

5 RESULTS

To test the model, we used confirmatory factor analysis and structural equation modeling (SEM) using AMOS (4.01) (Arbuckle & Wothke, 1999). AMOS is an advanced SEM technique that uses maximum likelihood estimation (MLE), among other estimation techniques and permits the simultaneous testing of both the measurement model and the structural model.

5.1 Measurement Model

Self-efficacy measures. Each respondent had four self-efficacy scores, two for applications (word processing and spreadsheets), one global, and GCSE. Each CSE measure was factor analyzed separately to assess dimensionality and reliability and then all four were factor analyzed simultaneously to test convergent and discriminant validity. When factor analyzed separately, three of the scales were one-dimensional. GCSE, however, extracted two factors. To maintain a one-dimensional construct, three of the ten survey items were eliminated. All scales had excellent reliability.

When factor analyzed simultaneously, an iterative process was used to eliminate items with high cross-loads. This process eliminated one item from the spreadsheet scale and three from the word processing scale. The resulting WP and SS scales had nine items, GCSE had seven, and the global instrument two. Discriminant validity is indicated if each item loads highest on its own construct, which was the case. A second way to assess it is by computing average variance extracted (AVE). AVE should be greater than .50 to justify using a construct and greater than other construct correlations (Fornell & Larcker, 1981). Descriptive statistics and results are presented in Table 1; on the diagonal in bold-face is AVE. The two items for the GCSE and global instrument loaded on the same factor. This was expected, since both scales purport to measure general self-efficacy, and this establishes some convergent validity. The correlation between GCSE and global was .68, indicating to some extent they measured the same domain. They differed, however, in their relationship with other constructs (next section) and the loadings for the seven items from the GCSE scale were all higher, between .70 and .85, compared to the two global items (.62 and .66). These results indicate adequate reliability and discriminant validity. This factor analysis is presented in Appendix B.

Construct	**Mean**	**SD**	**ά**	**Correlations and AVE**			
				(1)	**(2)**	**(3)**	**(4)**
(1) AS-WP	9.14	1.35	.94	**.88**			
(2) AS-SS	7.49	2.58	.97	.43**	**.90**		
(3) GCSE	6.77	1.84	.92	.39**	.47**	**.85+**	
(4) Global	7.72	1.92	.84	.48**	.48**	.68**	**.85+**

Table 1. Descriptive statistics and correlations of CSE constructs
*n = 308. Bolded elements along diagonal represent AVE. Off diagonal elements are correlations. ** p < .01.*
+ GCSE and Global loaded on same factor; hence, these were combined

Construct	**Mean**	**SD**	**ά**	**Correlations and AVE**	
				(1)	**(2)**
(1) Anxiety	1.82	.90	.91	**.88**	
(2) Affect	4.74	1.22	.90	-.58**	**.89**

Table 2. Descriptive statistics and correlations of attitude constructs
*n = 308. ** p < .01. Bolded elements along diagonal represent AVE.*

Attitude scales. The two attitude scales were examined in a like manner as the CSE scales. The anxiety and attitude scales were both one-dimensional when factored separately. When combined, one item from each scale was eliminated, due to cross-loadings. The resulting scales had excellent psychometric properties. The factor analysis is presented in Appendix B. Results are presented in Table 2.

Performance Tests. Respondents received a single performance test for either word processing or spreadsheets. Although they were optional, a total of 204 tests were received (66%), including 105 WP tests (mean = 7.8, sd = 2.5) and 99 SS tests (mean = 9.5, sd = 3.7).

5.2 Structural model

Given satisfactory measurement models, the structural models were tested next. Structural model testing was an iterative process. For each dependent variable (anxiety, affect, WP and SS performance), a separate model was created which included all applicable independent variables (CSEs). This we labeled the *base model*. Because each self-efficacy measure was expected to correlate with all other SE measures, we allowed these paths to be freely estimated. We ran the model, examining its goodness of fit, and tested the strength of each path. We then created and tested a nested sub-model, based on those paths that were statistically significant in the base model. We did this by eliminating (constraining to zero) all non-significant paths in the base model. The model thus created we called the *reduced model*.

Affect									
Model	**CSEs (Paths)**	**Path CVs**	**Constrained Paths**	**df/X^2**	**NFI**	**$\Delta \chi^2$**	**Δ df**	**p < .05**	**Significant CSEs**
Base ($R^2 = .47$)	Global (.49) GCSE (.15) AS-WP (.04) AS-SS (.06)	4.89** 1.83 (ns) .77 (ns) 1.20 (ns)	None	485/1475	.96				
Reduced ($R^2 = .49$)	Global (.70)	11.2**	AS-WP, AS-SS, GCSE	488/1480	.96	5.5	3	ns	**Global**
Anxiety									
Model	**CSEs (Paths)**	**Path CVs**	**Constrained Paths**	**df/X^2**	**NFI**	**Δ χ2**	**Δ df**	**p < .05**	**Significant CSEs**
Base ($R^2 = .46$)	Global (-.59) GCSE (-.02) AS-WP (-.13) AS-SS (-.05)	-5.56** .260 (ns) -2.14* -.91 (ns)	None	517/1551	.96				
Reduced ($R^2 = .46$)	Global (-.60) AS-WP (-.14)	-8.58** -2.37*	GCSE, AS-SS	519/1552	.96	.8	2	ns	**Global, AS-WP**
Word Processing Test									
Model	**CSEs (Paths)**	**Path CVs**	**Constrained Paths**	**df/X^2**	**NFI**	**$\Delta \chi^2$**	**Δ df**	**p < .05**	**Significant CSEs**
Base ($R^2 = .25$)	Global (.41) GCSE (-.13) AS-WP (.23)	1.70+ -.62 (ns) 1.91*	None	147/440	.94				
Reduced ($R^2 = .23$)	Global (.28) AS-WP (.26)	2.41* 2.37*	GCSE	148/440	.94	.4	1	ns	**Global, AS-WP**
Spreadsheet Test									
Model	**CSEs (Paths)**	**Path CVs**	**Constrained Paths**	**df/X^2**	**NFI**	**$\Delta \chi^2$**	**Δ df**	**p < .05**	**Significant CSEs**
Base ($R^2 = .76$)	Global (.04) GCSE (.10) AS-SS (.76)	.32 (ns) .87 (ns) 7.92**	None	148/513	.92				
Reduced ($R^2 = .76$)	AS-SS (.87)	13.9**	Global, GCSE	150/514	.92	1.0	2	ns	**AS-SS**

Table 3 Structural model results

*** p < .01 * p < .05 + p < .10. ns in the p < .05 column stands for not significant, which means there is no statistical difference between the base and reduced models, indicating that the constrained CSEs (those constrained to zero or excluded from the model) should be excluded*

If the eliminated variable(s) in the reduced model were indeed *not statistically significant*, the nested-model should *not* be a significantly *poorer fit* than the base model. Put another way, if the reduced model is indeed statistically the same, then the non-significant CSEs *should* be eliminated. Because the difference in chi-squares of two nested models is itself a chi square distribution, we statistically tested this difference between base and reduced models, given the difference in degrees of freedom. This step established that the eliminated variable(s) were not statistically significant and could be removed

For all analyses, each model was examined in two primary ways. First, we examined whether the model was a good fit for the data. We used two primary indicators for this analysis, χ^2 and NFI. The ratio of χ^2 to degrees of freedom should be less than three (Schumacker & Lomax, 2004). NFI, or normed fit index, is an incremental goodness of fit measure which compares the given model to a null model. NFI should be greater than .90 for a good model fit (Byrne, 2001). Secondly, we examined whether each path that was freely estimated (i.e., not constrained to zero) was statistically significant. This is the critical value (CV), which must be greater than 1.96 (.05 significance level) (Schumacker & Lomax, 2004). We retained variables in the base model at $p < .10$ (CV > 1.65), but not in the reduced model.

Results are presented in Table 3. For the domains of affect and anxiety (measures at a general computing level), both the base and reduced models exhibited a good fit for the data. The amount of variance extracted in these models was relatively high, even in the reduced models, with the lowest at 46%. For both DVs, the global measure was statistically significant. For anxiety both the global instrument and AS-CSE (word processing or AS-WP) were statistically significant. Although the global instrument had a much higher standardized path value than AS-WP (-.60 vs. -.14), the latter could not be constrained to zero without significantly reducing the model fit ($\Delta\chi^2$ of 5.5, $\Delta df = 1$, $p < .02$). For the two performance tests, the fit was satisfactory, although the χ^2 to degrees of freedom was slightly above three for the spreadsheet model. For the word processing model, both the global instrument and AS-WP were statistically significant (both with similar path strengths). AS-SS was the only statistically significant path for the spreadsheet performance test model.

6 DISCUSSION

6.1 Summary of Findings

While computer self-efficacy has proved to be an important and useful construct for both practitioner and researcher, clarification of the construct and the search for a reliable, short instrument prompted this study. In predicting employee success on the job, with respect to technology, self-efficacy provides an indication of computing competence, while simultaneously and more importantly presenting a useful indicator of internal individual motivational processes that significantly influences effort, persistence, and future performance. We believe that more organizations would use such a measure if there was a valid, easy to administer instrument and IT researchers promoted the same to practitioners. At a minimum it should be used as a tool for new IT hires as a way of discerning potential IT performance, similar to the way organizations test and interview for personality type to assess fit. This study clarifies CSE by empirically examining its structure, in particular the dimension of magnitude.

Study results demonstrate that the following CSE(s) were significantly influential: Affect: **Global;** Anxiety: **Global, AS-WP**; WP Test: **Global, AS-WP**; SS Test: **AS-SS**.

While the contention has been made that including a magnitude dimension is important, this study provided little support for such a claim. The only instrument without a magnitude dimension was the global instrument, and it demonstrated rather remarkable significance with all DVs except the spreadsheet test. These findings suggest that including magnitude in a CSE scale may not always provide the most accurate means of determining an individual's actual perception of ability (CSE) for the intended computing domain. Individuals seem quite capable of accurately judging their ability without explicitly being presented levels of task difficulty. As suggested, computing domains are different in that they each consist of a variety of sub-tasks with no inherent ordering by difficulty. We theorize that because of this, scales which *do include* an ordering by difficulty may actually mislead respondents by focusing their attention on tasks less important to their particular perception of ability for that domain. One of the findings which support this, at least at the general computing level, is the amount of variance explained by the global scale, which was 49% and 46% (affect and anxiety, respectfully). The global instrument was not merely better than the others; it was a very strong predictor in its own right. *For outcomes at the general computing level, these findings suggest that CSE instruments that include magnitude may not be well-suited; magnitude adds little explanatory power and may actually distort results*.

For the application domains of word processing and spreadsheets, each respective AS-CSE had a statistically significant relationship with its performance test. This was expected, given they were measured at the same application level. What was surprising was that the global instrument also had a significant relationship with the WP test. One possible explanation is that word processing has become almost ubiquitous (at least in this population). As an application becomes more pervasive, more widely used and known by respondents, competence in the domain increases. At this point it may be more susceptible to the influence of general computing measures (like global). Contrast this to the spreadsheet domain, where respondents had less competence. In that model, only spreadsheet CSE was significant, but accounted for an amazing 76% of the variability in the performance test. *For outcomes at the application-specific computing level, these findings suggest that magnitude adds explanatory power but is influenced by competence in the application; magnitude has a greater influence on relationships between CSE and performance for applications where respondents have less competence.*

6.2 Implications and limitations

Deciding on an acceptable instrument is an organization/researcher choice that is an important decision, according to study results. When the outcome belongs to the entire computing domain, such as computer anxiety or general computer performance (not some application performance), instruments without magnitude, such as the global instrument, are better. When the outcome is at an application level, the best instrument choice depends on the domain. For domains where respondents have less competence, application specific CSEs are clearly superior (particularly those based on actual tasks). For domains in which respondents have lots of competence, either a general (global) or an application-specific CSE is best. It is important, therefore, to have some knowledge of the population prior to testing.

Because of specificity matching, we expected general instruments to have a greater influence on general outcomes and application-specific CSEs to have greater influence on the specific performance tests. This study had mixed results, in that the global instrument was significant for WP test and AS-WP significantly influenced anxiety (though this influence was small). In *post hoc* testing (using SEM), every CSE had a significant relationship with all four outcomes when it was the *only predictor* in each model. This provides some evidence of CSEs generalizing to other domains; it also suggests that matching specificity may not always be necessary. This requires additional study.

One of the biggest surprises with important implications concerns the GCSE instrument of Compeau and Higgins (1995a). It was significant in none of the models (except in the *post hoc* testing when it was the sole predictor). This finding we view as alarming, particularly since it is commonly utilized. Marakas et al. (1998) suggest that GCSE is inherently confusing, while the authors themselves (1995a) note that it might really measure self-efficacy for *learning* computing. We believe this instrument does not adequately isolate an individual's perception of ability for a given domain. The levels of assistance appear problematic; they don't have a full range of assistance (there are no lots of assistance levels). The items are also ambiguous in terms of mutual exclusivity. For example, one of the levels of assistance is "…if I had never used a package like it before". But are other forms of assistance available? We suspect some respondents assumed there was, others that there was not.

As with any empirical study, there are several limitations. In any cross-sectional instrument, common method bias and other related limitations arise. Although reliabilities were high, anytime all measures are collected simultaneously validity may suffer. Of the dependent variables, this was true of both attitudes. However both performance tests were objective in nature, adding validity to this study. It should be noted that CSE *should be* self-reported. Generalizability to a general population must be approached with caution. This population is one in a Navy commissioning program though some report that it is no different from other student groups (Neiberg, 2000). Another potential limitation is the predominance of males in this survey. T-tests were conducted and results showed that there was no statistically significant difference between the male/female groups in terms of age, class, major, college, or any self-efficacy judgment. Thus differences in numbers between genders did not appear to adversely affect the results.

6.3 Conclusion

The motivation for this study was to clarify the conceptualization of computer self-efficacy; in particular we wished to examine the structural dimension of magnitude, and whether or not it adds any explanatory power in the relationship between self-efficacy and common outcomes. Results indicate that global-type instruments are quite acceptable and even better than the commonly used GCSE (Compeau & Higgins, 1995a), particularly when the outcomes apply to the entire computing domain or apply to applications in which respondents have lots of competence. This finding is at odds with leading cognitive researchers (Bandura, 1997; Compeau & Higgins, 1995a; Gist & Mitchell, 1992; Marakas et al., 1998). We believe there are two main reasons for this finding: self-efficacy in IT is not task-based, but domain-based and there are no absolute levels of task difficulty in domains. Individuals seem quite capable of judging their computer efficacy without being presented levels of task difficulty (i.e., magnitude), except when their competence in the domain is low (or the domain is new).

The global instrument in this study (Hill et al., 1987) has only two items, which can be problematic with some SEM estimation techniques, though not MLE (Byrne, 2001). It was reduced from the original four based on the concern that two of its items may not measure CSE (Compeau & Higgins, 1995a). The resulting instrument, however, was surprisingly robust in predictive power and should be studied further, along with other application and environment domains.

Self-efficacy has demonstrated capability in predicting computer performance, attitudes, and beliefs. Because it includes a motivational component, it is more useful than mere competence testing in predicting employee success. We suggest it is time for organizations to make use of this construct, choosing the appropriate instrument for the population to be surveyed.

References

Ajzen, I. (1991). The theory of planned behavior. Organizational Behavior and Human Decision Processes 50, 179-211.

Al-Jabri, I. M. and Al-Khaldi, M.A. (1997). Effects of user characteristics on computer attitudes among undergraduate business students. Journal of End User Computing 9 (2), 16-22.

Arbuckle, J. L. and Wothke, W. (1999). Amos 4.0 User’s Guide. SmallWaters Corporation, Chicago.

Bandura, A. (1986). Social Foundations of Thought and Actions. Prentice-Hall, Englewood Cliffs, N.J.

Bandura, A. (1997). Self-efficacy: the Exercise of Control. W.H. Freeman and Company, New York.

Bandura, A., Reese, L. and Adams, N.E. (1982). Microanalysis of action and fear arousal as a function of differential levels of perceived self-efficacy. Journal of Personality and Social Psychology 43 (1), 5-21.

Byrne, B.M. (2001). Structural Equation Modeling with AMOS. Lawrence Erlbaum Associates, London.

Chen, G., Gully, S.M., and Eden, D. (2001). Validation of a new general self-efficacy scale. Organizational Research Methods 4 (1), 62-84.

Colquitt, J.A., LePine, J.A. and Noe, R.A. (2000). Toward an integrative theory of training motivation: A meta-analytic path analysis of 20 years of research. Journal of Applied Psychology 85 (5), 678-707.

Compeau, D.R. and Higgins, C.A. (1995a). Computer self-efficacy: Development of a measure and initial test. MIS Quarterly 19 (2), 189-202.

Compeau, D.R. and Higgins, C.A. (1995b). Application of social cognitive theory to training for computer skills. Information Systems Research 6 (2), 118-143.

Fornell, C. and Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. Journal of Marketing Research 18 (1), 39-50.

Gist, M.E. (1987). Self-efficacy: implications for organizational behavior and human resource management. The Academy of Management Review 12 (32), 472-485.

Gist, M.E. and Mitchell, T.R. (1992). Self-efficacy: a theoretical analysis of its determinants and malleability. Academy of Management Review 17 (2), 183-211.

Hasan, B. (2006). Effectiveness of computer training: The role of multilevel computer self-efficacy. Journal of Organizational and End User Computing 18(1), 50-68.

Hill, T., Smith, N.D. and Mann, M.F. (1987). Role of efficacy expectations in predicting the decision to

use advance technologies: the case of computers. Journal of Applied Psychology 72 (2), 307-313.
Hung, S. (2003). Expert versus novice use of the executive support systems: An empirical study. Information and Management 40 (3), 177-189.
Igbaria, M. and Iivari, J. (1995). The effects of self-efficacy on computer usage. Omega 23 (6), 587-605.
Johnson R.D. and Marakas, G.M. (2000). Research report: the role of behavioral modeling and computer skills acquisition-toward refinement of the model. Information Systems Research 11 (4), 402-417.
Kernan, M. and Howard, G.S. (1990). Computer anxiety and computer attitudes: an investigation of construct and predictive validity issues. Educational and Psychological Measurement 50, 681-690.
Lee, C. and Bobko, P. (1994). Self-efficacy beliefs: comparison of five measures. Journal of Applied Psychology 79 (3), 364-369.
Loyd, B.H. and Gressard, C. (1984). The effects of sex, age, and computer experience on computer attitudes. AEDS Journal (Winter), 67-77.
Marakas, G.M., Yi, M.Y. and Johnson, R.D. (1998). The multilevel and multifaceted character of computer self-efficacy: toward clarification of the construct and an integrative framework for research. Information Systems Research 9 (2), 126-163.
Martocchio, J.J. and Webster, J. (1992). Effects of feedback and cognitive playfulness on performance in microcomputer software training. Personnel Psychology 45, 553-578.
Munro, M.C., Huff, S.L., Marcolin, B.L. and Compeau, D.R. (1997). Understanding and measuring user competence. Information and Management 33, 45-57.
Murphy, C. A., Coover, D., and Owen, S. V. (1989). Development and validation of the computer self-efficacy scale. Educational and Psychological Measurement, 49 (2), 893-899.
Neiberg, M.S. (2000). Making citizen soldiers: ROTC. Harvard University Press, Cambridge, MA.
Rainer, R.K. and Harrison, A.W. (1993). Toward development of the end user computing construct in a university setting. Decision Sciences 24 (6), 1187-1202.
Schumacker, R.E. and Lomax, R.G. (2004). A Beginner's Guide to Structural Equation Modeling. Second Edition. Lawrence Erlbaum Associates, London.
Thatcher, J.B. and Perrewe, P.L. (2002). An empirical examination of individual traits as antecedents to computer anxiety and computer self-efficacy. MIS Quarterly 26 (4), 381-396.
Vijayasarathy. L.R. (2004). Predicting consumer intentions to use on-line shopping: the case for an augmented technology acceptance model. Information and Management 41, 747-762.
Woodrow, J.E. (1991). A comparison of four computer attitude scales. Journal of Educational Computing Research 7 (2), 165-187.
Yi, M.Y. and Davis, F.D. (2003). Developing and validating an observational learning model of computer software training and skill acquisition. Information Systems Research 14 (2), 146-169.

Appendix A: Selected Survey Items

Global Self-efficacy Scale
…I believe I have the ability to learn computer applications.
…I have great confidence in my computer ability.

Word Processing Self-efficacy Scale
...to <u>effectively</u> use different fonts, including colors and sizes, in a word processing file.
…to apply character (letter) effects such as bolding, italicizing, or subscripting in a word processing document.
…to cut, copy and paste in a word processing document.
…to add symbols (such as foreign letters or yen symbol) to a document.
…to insert and manipulate endnotes or footnotes in a word processing program.
…to efficiently set different margins in a word processing document.
…to set page or section breaks in a word processing file.
…to effectively add bulletized (or numbered) lists to a word processing document.

…to add pictures, graphics, or objects into a word processing document.

Spreadsheet Self-efficacy Scale (from Johnson & Marakas, 2000)
…to manipulate the way a number appears in a spreadsheet.
…to use and understand the cell references in a spreadsheet.
…to use a spreadsheet to communicate numeric information to others.
…to use a spreadsheet to assist me in making decisions.
…to manipulate data using formulas that come with a spreadsheet program.
…to write a simple formula in a spreadsheet to perform math calculations.
…to summarize numeric information using a spreadsheet.
…to use a spreadsheet to share numeric information with others.
…to use a spreadsheet to display numbers as graphs.

Appendix B: Factor analyses of scales

CSE Scales					**Attitude Scales**		
	1	**2**	**3**			**1**	**2**
GCSE1	.13	.09	**.83**		Anxiety1	**.77**	-.21
GCSE2	.19	.04	**.82**		Anxiety2	**.73**	-.15
GCSE3	.17	.11	**.85**		Anxiety3	**.85**	-.27
GCSE4	.15	.16	**.82**		Anxiety4	**.69**	-.30
GCSE5	.14	.26	**.70**		Anxiety5	**.76**	-.27
GCSE7	.21	.22	**.71**		Anxiety6	**.77**	-.23
GCSE8	.28	.08	**.70**		Anxiety7	**.82**	-.24
GL1	.21	.34	**.62**		Liking1	-.26	**.80**
GL2	.30	.26	**.66**		Liking2	-.32	**.76**
WP1	.08	**.86**	.17		Liking3	-.16	**.85**
WP2	.07	**.88**	.14		Liking4	-.21	**.79**
WP3	.05	**.89**	.12		Liking5	-.33	**.70**
WP4	.21	**.74**	.24		Liking6	-.20	**.77**
WP5	.28	**.74**	.13				
WP6	.21	**.82**	.16				
WP7	.24	**.72**	.14				
WP8	.22	**.84**	.14				
WP9	.18	**.82**	.18				
SS1	**.78**	.22	.26				
SS2	**.82**	.20	.24				
SS3	**.86**	.21	.15				
SS4	**.89**	.18	.20				
SS5	**.87**	.13	.20				
SS6	**.91**	.14	.18				
SS7	**.90**	.20	.18				
SS8	**.86**	.17	.19				
SS9	**.78**	.14	.30				

GCSE (general instrument of Compeau and Higgins, 1995a); GL (global instrument derived from Hill et al. 1987); WP (application-specific, word processing; self-developed); SS (spreadsheets, from Johnson and Marakas, 2000)