December 2001

Assessing Task-Technology Fit in Simulation Modeling

Mark Dishaw
University of Wisconsin, Oshkosh

Diane Strong
Worcester Polytechnic Institute

D. Brent Bandy
University of Wisconsin, Oshkosh

ASSESSING TASK-TECHNOLOGY FIT IN SIMULATION MODELING

Mark T. Dishaw
University of Wisconsin, Oshkosh
dishaw@uwosh.edu

Diane M. Strong
Worcester Polytechnic Institute
dstrong@wpi.edu

D. Brent Bandy
University of Wisconsin, Oshkosh
bandy@uwosh.edu

Abstract

This research examines task-technology fit in a simulation modeling context using the method of Dishaw & Strong (1998b, 1999). An instrument is developed for assessing task needs, technology characteristics, and the resulting fit for simulation tasks using simulation modeling tools.

Overview

Task-Technology Fit (TTF) is a well-known construct in the MIS literature. The core thesis of TTF Models is that technology, e.g., software, will be used if, and only if, the functions available to the user support (fit) the activities of the user. A software function supports an activity if it facilitates that activity. Rational, experienced users will choose those tools and methods that enable them to complete the task with the greatest net benefit. Software that does not offer sufficient advantage will not be used. One method of assessing TTF, first elaborated by Goodhue (1988), assesses TTF directly using twelve variables to measure fit, without assessing task needs or available technology features (Goodhue, 1988; 1995). While the relatively low explanatory power of this method is not unusual for social science models, it is much lower than TAM, an alternative model for studying user's choices about technology (Davis, Bagozzi, & Warshaw, 1989; Dishaw & Strong, 1999; Taylor & Todd, 1995).

A second, more recently developed method of assessing TTF uses a definition of TTF conceptually similar to that of Goodhue (1988), but measures task needs, technology features, and uses these data to derive a resulting fit measure (Dishaw & Strong 1998b). This method was tested in the context of software maintenance. It provides better explanatory power, equivalent to that of TAM, but requires specific models of task and technology.

This paper describes an extension of the Dishaw & Strong (1998b) study to develop measures for general models of task and technology that apply in any problem-solving context. We began with their original instrument and re-developed it for use in a simulation modeling context. The results, analyzed using structural modeling techniques, are consistent with Dishaw & Strong's (1998b, 1999) findings. Further work is planned to demonstrate the instrument's validity across a variety of problem solving venues.

Task–Technology Fit Models

The ability of software to support a task is expressed by the formal construct known as Task-Technology Fit, which is the matching of the capabilities of the technology to the demands of the task. Figure 1 is a general TTF Model. In earlier TTF studies, the dependent variable in the models of fit is performance, e.g, (Goodhue and Thompson, 1995). Dishaw & Strong (1998a; 1998b, 1999), however, focus on the performance antecedent, tool usage, as the dependent variable, which is most appropriate

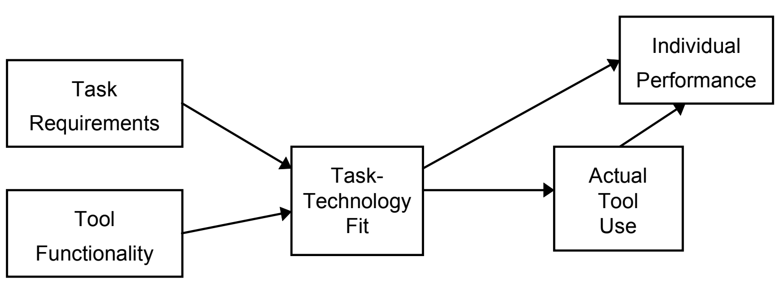

Figure 1. Task/Technology Fit Model

when the use of the tools is voluntary, as it was in their software maintenance context. This allowed them to consider a dependent variable that is closer, from the perspective of the causal chain, to the independent variable fit. This research continues tool usage as the dependent variable.

The model of fit between the maintenance task and software tool functionality (Dishaw & Strong 1998b, 1999), which serves as the basis for the development of a general TTF assessment method, is briefly described.

Maintenance Task Model. Dishaw & Strong (1998b) based their model of the maintenance task and the key dimensions involved on the empirical work of several MIS researchers. The specific actions that make up the major maintenance task activities of Understanding and Modification, i.e., planning, knowledge building, diagnosis, and modification activities, were identified during protocol analysis sessions of working maintainers (Vessey, 1985; 1986). The first three activities cover understanding, while the last one is the actual program transformation activity.

Maintenance Technology Model. The Henderson and Cooprider (1990) Functional Case Technology Model (FCTM) provided a description of the basic functions present in design support software (CASE). The functions that support an individual programmer developing or changing software include representation, analysis, and transformation functionality.

Task and Technology Models for Problem Solving

The goal in this research is to generalize the software maintenance TTF model and produce a general instrument and technique to access TTF for any problem-solving task and supporting technology. Previously, we argued that the software maintenance task and technology models are appropriate for general problem solving tasks and tools that support design and problem-solving tasks (Dishaw, Strong, & Bandy 1999). Specifically, we noted that Vessey's work is well grounded in the problem solving and cognitive science literature, and the technology model is grounded in the literature on information technology support functionality. Thus, the starting point for our general task and technology models are the software maintenance models used in their general form.

Research Method

Item and Scale Development. The items from Dishaw & Strong's long form maintenance instrument, which contains items for all of the factors identified in Vessey's (1985, 1986) debugging model, as well as items for the functional case tool model (Henderson and Cooprider, 1990), were used as the basis for a new general instrument. Items were rewritten to reflect problem-solving tasks by removing references to software maintenance and debugging, and rewriting items to address problem-solving activities. Similarly, the maintenance tool items were rewritten to reflect problem-solving support. Some items were deleted entirely.

Data Collection. The revised instrument was administered to undergraduate management students from several Operation Management classes. The instrument was administered after completion of an ordinary simulation modeling assignment. We obtained 109 valid (useable) data points for conducting the data analysis. All students completed the same assignment.

Data analysis. We refined the instrument and "culled" items that did not contribute to the scale. Confirmatory Factor Analysis (CFA) and tests of the overall fit of the model were accomplished using the AMOS package (Arbuckle & Wothke, 1999) and supported by additional analysis using SPSS for Windows package.

Preliminary Results

Measurement Model. The results of our analysis produced four items for the task activity variables, planning, knowledge building and modification, and three items for diagnosis activities. There were three items for each of the technology variables, analysis, representation and transformation. The utilization variables, construction and model checking, contained three items each. The measurement model details will be presented in August, but are omitted here to conserve space. The items for each variable were averaged for use in fitting the structural model.

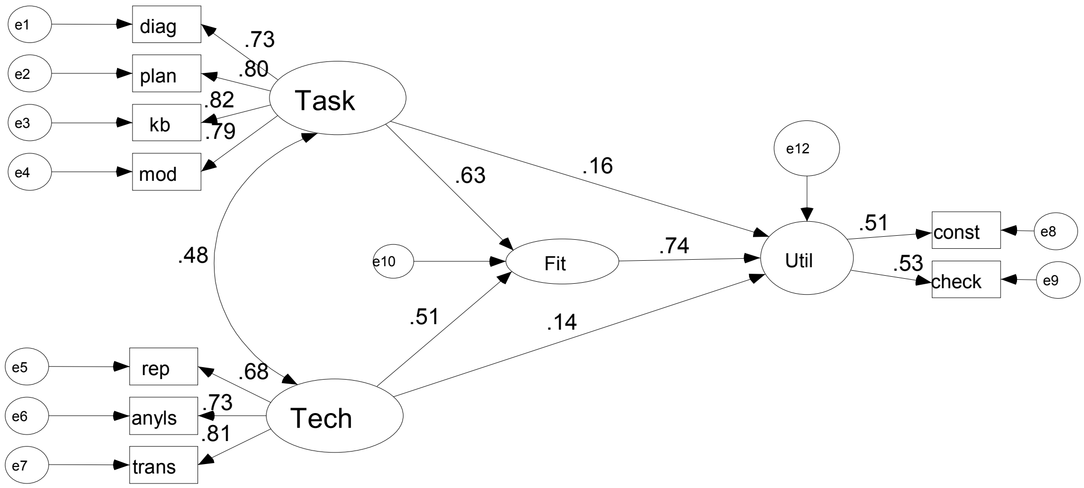

Figure 2. Structural Model with Standardized Regression Weights

Structural Model. The resulting structural model is shown in Figure 2. The fit measures for the model, χ^2=32.5, df=24, χ^2/df=1.4, p=0.12, GFI=0.94, AGFI=0.89, Incremental fit index=0.98, indicate an acceptable fit of the model to the data. The result is consistent with that reported by Dishaw & Strong (1999).

Conclusion. The results further demonstrate the feasibility of assessing task-technology fit using a method of measuring task needs, technology functionality, and deriving fit, an unobserved variable, from these measures. This study goes beyond the original context of software maintenance to demonstrate the method's use in a more general problem-solving context. Our next steps are to analyze further these simulation modeling data for reporting at the conference, and to apply the questionnaire in other problem solving contexts to validate the generalizability of the method.

References

Arbuckle, J.L and Wothke, W, *Amos Users' Guide Version*, Chicago: SmallWaters Corporation, 1999.

Davis, F.D., Bagozzi, R.P. and Warshaw, P.R. "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science* (35:8), 1989, pp. 982-1003.

Dishaw, M.T. and Strong, D.M. "Assessing Software Maintenance Tool Utilization Using Task-Technology Fit and Fitness-for-Use Models," *Journal of Software Maintenance: Research and Practice* (10:3), 1998a, pp. 151-179.

Dishaw, M.T. and Strong, D.M. "Supporting Software Maintenance with Software Engineering Tools: A Computed Task-Technology Fit Analysis," *Journal of Systems and Software* (44:2), 1998b, pp. 107-120.

Dishaw, M.T. and Strong, D.M. "Extending the Technology Acceptance Model with Task-Technology Fit Constructs," *Information and Management* (36:1), 1999, pp. 9-21.

Dishaw, M.T., Strong, D.M. and Bandy, D. B. "Developing a General Method to Access Task-Technology Fit", *Proceedings of the Fifth Americas Conference on Information Systems*, August 1999, Milwaukee, WI.

Goodhue, D.L. "Supporting Users of Corporate Data: The Effect of I/S Policy Choices," Unpublished Doctoral Dissertation, Massachusetts Institute of Technology, 1988.

Goodhue, D.L. "Understanding User Evaluations of Information Systems," *Management Science* (41:12), 1995, pp. 1827-1844.

Goodhue, D.L. and Thompson, R.L. "Task-Technology Fit and Individual Performance," *MIS Quarterly* (19:2), 1995, pp. 213-236.

Henderson, J.C. and Cooprider, J.G. "Dimensions of I/S Planning and Design Aids: A Functional Model of CASE Technology," *Information Systems Research* (1:3), 1990, pp. 227-254.

Taylor, S. and Todd, P.A. "Understanding Information Technology Usage: A Test of Competing Models," *Information Systems Research* (6:2), 1995, pp. 144-176.

Vessey, I. "Expertise in Debugging Computer Programs: Situation-Based versus Model-Based Problem Solving," *International Conference on Information Systems*, 1985.

Vessey, I. "Expertise in Debugging Computer Programs: An Analysis of the Content of Verbal Protocols," *IEEE Transactions on Systems, Man, and Cybernetics* (SMC-16:5), 1986, pp. 621-637.