Training for Generalization: The Role of Integrated Skills and Knowledge in Technology Domains

Dorrit Billman[1], Richard Catrambone[2], Jolene Feldman[1],
Zachary Caddick[3], Sky Eurich[1], Jonathan Leventhal[1], Rob Martin[1], & Kasia Sliwinska[1]
San Jose State University Research Foundation @ NASA's Ames Research Center[1]
Georgia Institute of Technology[2], University of Pittsburg[3]

Training is of little value if trainees can only do the exact tasks on which they were trained, in the identical context of training. Rather, the value typically comes from the ability to apply skills and knowledge across novel variation in contexts and tasks. Training in dynamic technical domains can be particularly challenging because the future tasks can rarely be fully anticipated. We hypothesize that generalization in technology domains will be facilitated when principles (such as device models) are taught in addition to operational procedures, and, particularly, when principles and procedures are integrated. We conducted an exploratory study, including method development, using a micro-world with simulated International Space Station Habitat systems. We compared the effects of Integrated versus Component-wise Training Conditions on generalization to varied tasks, quite different from those in training. Exploratory analyses suggested better generalization and transfer in the Integrated Condition.

INTRODUCTION

Training is of little value if trainees can only do the exact tasks on which they were trained, in the identical context of training. Rather, the value typically comes from the ability to apply skills and knowledge across novel variation in contexts and tasks. Our research focuses on *generalizing* skills and knowledge to new tasks and problems.

Sometimes, the scope of generalization intended from training is clear, yet there are practical reasons why not every situation can be included in training (e.g. multiplication problems under a million). In other cases, the complexity of the domain means all situations cannot be anticipated; thus, successful training must rely on teaching people how to generalize to the unexpected. Goals, constraints, and resources may change from training. This may be particularly true for rapidly changing socio-technical systems. A radar systems technician using one equipment variant in the schoolhouse may be faced with different equipment shipboard. Astronauts on future long-distance missions will be faced with an unfamiliar environment, working to novel goals and in situations unimagined during training. Successful problem solving requires generalization from the content presented in training. We study this problem in the domain of operating complex equipment.

Background

Experts' ability to generalize (adaptive expertise) relies on schemas, principles, and more general procedures, which abstract away from details of examples (Carbonell, et al, 2014). Various training methods foster learning these types of generalization-promoting skills and knowledge. Collectively, research suggests learning activities that target both a) declarative knowledge of principles and b) procedures for solving example problems are more effective than learning activities that target just one of these.

Much training centers on working example problems and learning solution procedures. How this is done influences generalization. Comparing superficially different examples can promote analogy and schema formation, as can exposure to varied examples; in turn schema formation may promote generalization to quite different cases, or far transfer (reviewed in Nokes-Malack & Richey, 2015). Providing less detailed, more general instructions can also aid transfer, but learning from examples is often not sufficient for generalization (Catrambone, 1990; Van Der Meij, Blijleven, & Jansen, 2003; Wiedenbeck, 1989, Wittwer & Renkl, 2010). Structuring procedures or worked examples into subgoals helps students deconstruct a procedure into functional parts and aids generalization (Catrambone, 1998; Margulieux & Catrambone, 2016). In addition, instruction focused on principles and explanations is widely used. Self-generation of explanations is particularly valuable, especially for learners with relevant background knowledge (Chi, Bassok, Lewis, Reimann, & Glaser, 1989). Providing principles to enhance examples also aids transfer (Catrambone, 1995).

Generalization of technical skills (e.g., Bibby & Payne, 1993; Frederiksen & White, 1993; Karreman, J., Ummelen, N., & Steehouder, M., 2005) is particularly relevant to our work. Procedures often specify how to run equipment and principles describe internal workings of the system. Principles can help a learner understand why things operate the way they do and reason out what actions need to be taken in an unfamiliar situation. Kieras and Bovair (1984) showed that providing a mental model of a device in addition to procedure-based training produced better performance on novel problems. Schaafstal, Schraagen and VanBerl (2000) found that combining a specific, functional-style model of a device with a general "structured trouble-shooting" procedure dramatically improved transfer relative to standard training.

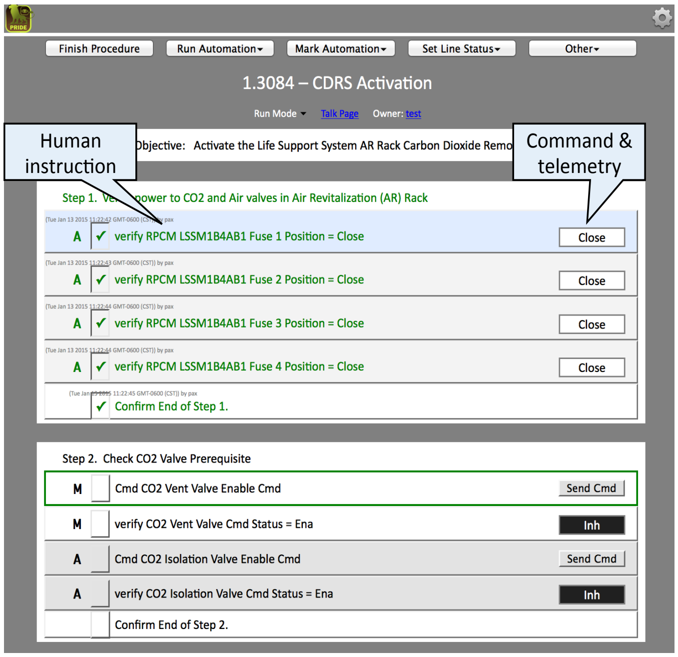


Figure 1. Example screenshot of procedure automation software for commanding the (simulated) habitat equipment.

We hypothesize that learning in technology-rich domains will be facilitated when principles of operation (e.g., device models) are taught in addition to procedures, and, further, when these principles and procedures are integrated. Integrated principles and procedures may provide rich, diverse retrieval cues allowing flexible access, may provide linkages among knowledge components that support inference; and may show the value of connecting information. Learners may be better able to use information in a new but related situation "transferring in" skills and knowledge to guide generalizing (Schwartz, Bransford, & Sears, 2005).

Our theoretical goal was investigating the effects of integrated device knowledge and procedural skills on generalization. Our required methodological goal was developing both training methods and transfer tasks. We compared two groups that were trained on the same component principles and procedures. In the Integration Condition, training was designed to integrate these principles and procedures, while the Component-wise (C-W) Condition kept principles and procedures separated during training.

METHOD

Design and Subjects

This exploratory study simultaneously a) developed the domain methods and measures relevant to integration and generalization and b) investigated the integration hypothesis. The between-subjects factor was training method. Dependent variables were solution success and secondarily, time, on a variety of generalization tasks. The generalization tasks were developed for this domain to assess different aspects of operational skill and conceptual knowledge and their scoring rubrics were created to capture variation in observed behavior.

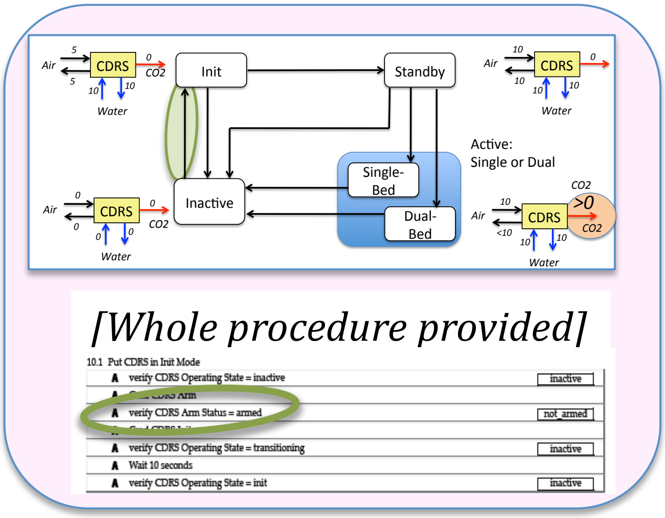
Participants were Aeronautics-Astronautics engineering students, from two universities, at undergraduate, masters, and Ph.D. levels. There were 13 in the Integrated Condition and 14 in the Component-wise Condition. We balanced conditions for the university and education level of participants. One Component-wise participant declined to complete some tasks.

Materials

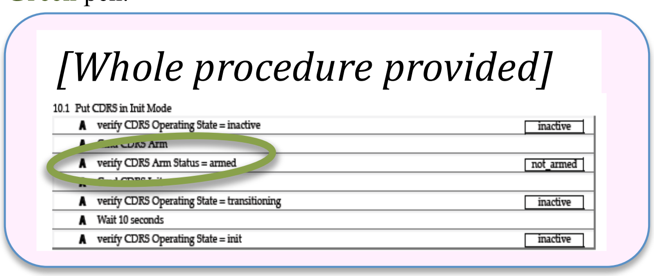
Our training environment was a micro-world providing simulated habitat equipment from the International Space Station (ISS) and software for executing operational procedures. It modeled the Carbon Dioxide Removal System (CDRS), the Active Thermal Control System (ATCS) providing cooling water, and the Remote Power Control modules (RPCMs) distributing power. Users operated equipment through the procedure automation software and associated procedures activating and deactivating equipment (Figure 1). The equipment simulator and procedure automation software have been used in prior studies and they provided guidance on training content and on varied tasks operating the system (Billman, Schreckenghost, & Billinghurst, 2015; Schreckenghost, Milam, & Billman, 2014; Schreckenghost et al. 2014).

Table 1. Condition Differences in Training

Characteristic	Integrated	Component-wise
Topic Order	Interleaved: information about systems controlled, software, and procedures interleaved to build relationships.	Segregated: information about software before procedures before policies before controlled system.
Question Types	Recall plus inference and prediction	Recall only
Markup task [both coloring]	Mapping between procedures and schematics: 1a) For an element marked on a schematic, find (and color) the element in a procedure; 1b) For an element marked on a procedure find (and color) the element in a schematic.	Separate Identification Tasks (different times): 1) Find & color a named component in a device schematic; 2) Find & color the step affecting a named component in a procedure.
Procedure Format	Instructions grouped into sub-goal steps	Instructions as a list



2A. Integrated Mark-Up Example: Model-> Procedure
Q1: Compare the state transition diagram and procedure. Find the transition highlighted in **Green** on the diagram. Look at the procedure and identify the actions that correspond to this transition.
Mark the corresponding lines of the procedure with the **Green** pen.



2B. Component-wise: Markup Procedure Alone.
Q1: Identify the transition that corresponds to switching the CDRS to the Init state.
Circle the corresponding section of the diagram with the **Green** pen.
[Component-wise Markup questions used schematic diagrams and procedures, but never together.]

Figure 2A: Integrated mark-up related procedures and device models.
Figure 2B: Component-wise markup separates these representations.

Training materials were presented as slides on a computer combined with a) hands-on use of the software and b) telling the experimenter answers to questions distributed through training. The content for both conditions was matched in information about the device, software, and procedures and in practice running procedures. They differed in information organization and in key activities designed to aid integration (summarized in Table 1). Figure 2A shows the Markup Task for the Integrated Condition; in this example, users identify the circled element (a component or process) in a procedure and then identify and color the corresponding representation in a device schematic, here the state transition diagram.

Training was also designed to ensure that novel problem types to-be-encountered in the generalization phase were kept genuinely novel. During training, participants were told what they should do with the procedures. They did not need to decide what procedures to run, they were not exposed to procedures running in mismatched situations where actions might fail, they did not have to troubleshoot, nor was there any mention of reconfiguring or writing procedures. Such situations were included in generalization.

Generalization included conceptual and procedural tasks. We discuss the 5 procedure execution tasks and one of the conceptual tasks. In the procedure execution tasks, participants were given 5 tasks that required running procedures and varied in how similar they were to the simple tasks presented in training versus how much additional inference was needed. Procedural Task 1 was highly similar to training tasks, requiring little generalization, and we expected little difference between conditions. Task 2 required identifying components of procedures to use together to accomplish the goal; no such analysis a use of procedure components had occurred in training. Tasks 3 and 4 were set up so that the user had to change existing conditions so procedures could execute successfully and if this was not done, the procedure would fail. Participants had not encountered fails, nor did they need to infer what unnamed procedure had to be executed to accomplish the goal. After these 4 tasks, participants did the conceptual task, Write-a-Procedure, Task 5A. In Task 5A users wrote a procedure for conducting a valve test activity. Both the task of writing a procedure and the goal of doing any testing were novel. Thus this task was very different from any learning activity. Participants were then asked to try executing their procedure to accomplish the testing goals, Task 5B. We expected better performance in the Integrated Condition on the procedural Tasks 2, 3, 4, and 5B and on the conceptual Write-a-Procedure, Task 5A.

Procedure

The experiment was run in a lab at NASA; a session lasted about four hours with short breaks. Users continued from training materials to the generalization tasks. Each procedural task required the state of the simulation to meet certain conditions (e.g. water over a certain temperature to create a failure). As participants did the procedural tasks, they changed system state. The changes in one task were designed to produce the state intended for the next task. We used checkpoints for experimenters to check and if necessary adjust the simulation so it was in the intended state. However, these checkpoints were not frequent enough to catch and correct all unintended states.

CODING AND RESULTS

For log file data recording the procedural tasks, we built scoring rubrics to characterize what the participants did, including whether the user succeeded in accomplishing the specified task goals. Despite correcting at checkpoints, sometimes users started a task when the system was not in the intended state. Such trials were not scored, as the

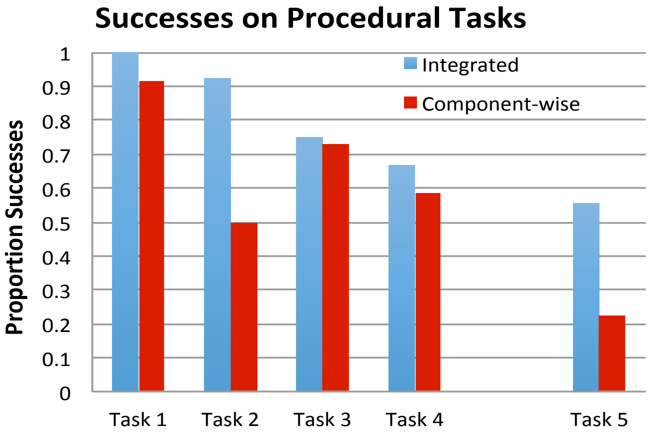


Figure 3. Proportion of successes on execution Generalization tasks. Participants contributing trials ranges from all 27 to 18 (in Task 5B).

participant then faced a distinct, unintended task. Six (of 65) trials had unintended initial states in the Integrated Condition and 12 (of 70) trials in the Component-wise Condition; this only happened if the participant had not accomplished the prior task and the divergence was not caught at a checkpoint. For successful trials we assessed the completion time.

For the conceptual Write-a-Procedure Task 5A, we scored 16 Basic Elements; we added good Extra Elements (e.g., conditionals to address alternative states) and subtracted bad elements (e.g., specifying an incorrect valve). Basic and Extra Elements were summed as Total Elements. We took a "data-mining" analysis approach, looking for patterns of difference between conditions and among measures. Our statistical comparisons are best viewed as filters for what differences to take most seriously.

Figure 3 shows that from early to late procedural tasks (1-5) successes tend to decrease, suggesting tasks identified as differing more from training were indeed more difficult. Between conditions, the ability to reach a solution was similar

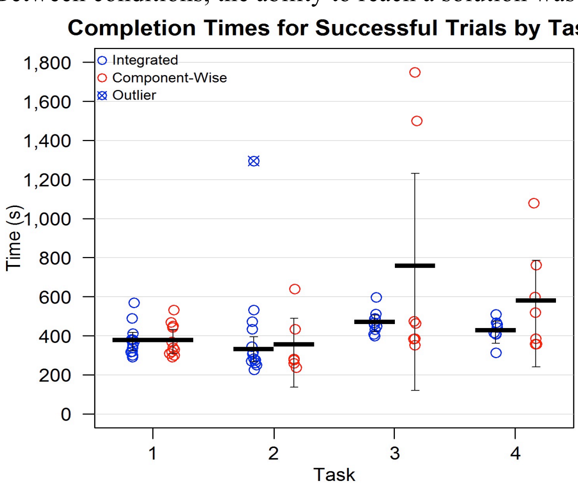


Figure 4. Completion times successful trials by task.

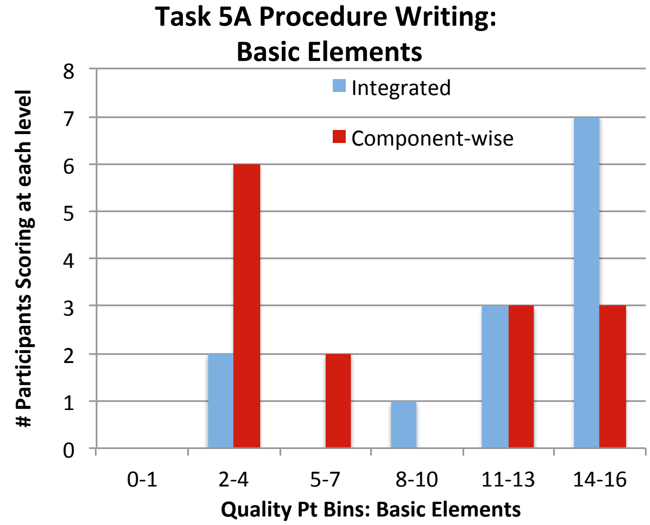


Figure 5. Condition distributions of Basic Elements Score on Procedure Writing Task 5A

on Tasks 1, 3, and 4, while Task 2 and 5 favored the Integrated Condition. The Task 2 proportion was the only task to differ significantly (Fisher exact probability $<.05$) between conditions. Although the Integrated Condition had a 33% success advantage on Task5B, performance from only the 18 users with the upgraded software version was available.

Figure 4 shows completion times of successful trials. Mean times for Task 5B were 5min 59s for Integrated (n=8) versus 7 min 51s for Component-wise (n=10). Integrated completion times tended to be shorter.

For the Write-A-Procedure Task5A, Figure 5 shows the bimodal distributions of scores on the 16-point Basic Elements Score. Median scores for Total Elements were 15 Integrating versus 5 in the Component-Wise Condition, Wilcoxon rank test ($W=50$, $p=.049$). The Condition effect was also significant ($p=.02$, CHI2 >5 , condition df=1) when fitting with Poisson distribution and analyzing the effect of condition (and school) using the GLMER module of R in each of several analysis approaches. Condition strategy differences are suggested by components within the Extra Elements score: system-monitoring procedures were included by 5 Integrated and no Component-wise participants while conditionals specifying order of actions were included by 5 Integrating and 2 Component-wise participants. Integrated users may have better understanding of the implications for effective procedure design.

DISCUSSION

Our research goal is to understand what makes effective training for generalization, particularly where it is infeasible to train for all types of situations that people will encounter. Our initial findings suggest that training to integrate operational procedures with the principles of technology may lead to better generalization on some novel, generalization tasks. Our work is methodologically innovative because it investigates a complex, work-relevant domain in controlled conditions. Specifically, we have developed and are further evolving a suite of procedural and conceptual tasks that differ from training tasks in multiple dimensions and degrees of difficulty. Investigation requires the joint development of training activities that are able to manipulate integration of tasks to measure differences in generalization.

The exploratory study reported here shows promise for our hypothesis and our methods. We provide initial evidence that our transfer tasks can measure and our training methods can alter the degree that people are able to generalize. Tasks 2 and 5 required users to decompose and reconfigure parts of procedures to serve new goals, even when no training or experience about reconfiguration or parts was provided. Tasks 3-5 required users to reason about what procedures are needed to reach old goals in new conditions. Task 5 also required users to engage in a completely different type of activity--writing a procedure for a new device where procedure writing was never hinted at in training. The tasks also required trouble-shooting from participants to avoid or recover from failures, although trouble-shooting situations or skills were not introduced in training. The patterns favoring the Integrated Condition are sometimes in numbers succeeding on task (i.e., accomplishing the task goal), sometimes by the speed of success, and sometimes by the use of a more general or comprehensive strategy.

Summarizing, our study provides initial evidence that our transfer tasks can measure, and our training methods can alter how effectively participants generalized. We found patterns favoring the Integrated Condition on several tasks and measures, with two trends significantly favoring the Integrated Condition. Measurement of our tasks is not very sensitive: success outcomes are important but binary measures are not very sensitive, our sample size is modest, and our individual differences are large.

In ongoing work, we are expanding our set of generalization tasks and our scoring rubrics. We are working to both measure and reduce participant variability. Individual differences in how readily our training engaged meta-cognitive processes may be important as well as differences in engineering knowledge. We plan to assess generalization over longer retention periods. We wish to investigate what aspects of the integrated training are effective, to explore the roles of multiple possible cognitive mechanisms involved, and to relate integration as a process and product to other factors aiding generalization.

ACKNOWLEDGEMENTS

Research supported by NASA HRP grant NNX15AP26G (Billman PI). Thanks to Debra Schreckenghost and TRAClabs for use and support of the PRIDEview software.

REFERENCES

Bibby, P. A., & Payne, S. J. (1993). Internalization and the use specificity of device knowledge. *Human-Computer Interaction*, *8*, 25–56.

Billman, D., Schreckenghost, D., & Billinghurst, S. (2015). Problem Solving when Procedure Conditions Are Not Met: Using Procedure-Automation Software for Support. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 59, pp. 294–298). Los Angeles.

Carbonell, K.B., Stalmeijer, R.E., Könings, K.D., Segers, M., & van Merriënboer, J.J.G. (2014). How experts deal with novel situations: A review of adaptive expertise. *Educational Research Review*, 12, 14-29.

Catrambone, R. (1990). Specific versus general procedures in instructions. *Human-Computer Interaction*, *5*, 49–93.

Catrambone, R. (1995). Following instructions: Effects of principles and examples. *Journal of Experimental Psychology: Applied*, *1*, 227–244.

Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General, 127,* 355-376.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*, 145–182.

Eiriksdottir, E., & Catrambone, R. (2015). The effects of timing of exposure to principle and procedural instruction specificity on learning an electrical troubleshooting skill. *Journal of Experimental Psychology: Applied*, *21* (4), 383-394.

Frederiksen, J.R. & White, B.Y. (1993). The avionics job-family tutor: An approach to developing generic cognitive skills within a job-situated context. Artificial intelligence in education, Al-ED 93, World Conference on Artificial Intelligence in Education, Edinburgh, Scotland.

Karreman, J., Ummelen, N., & Steehouder, M. (2005, July). Procedural and declarative information in user instructions: What we do and don't know about these information types. Paper presented at the *IEEE International Professional Communication Conference*, Limerick, Ireland

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science*, *8*, 255–273.

Margulieux, L. E., & Catrambone, R. (2016). Improving problem solving with subgoal labels in procedural instructions and worked examples. *Learning and Instruction, 42*, 58-71.

Noles-Malack, T.J. & Richey, J.E., (2015). Knowledge Transfer. In *Emerging Trends in the Social and Behavioral Sciences*. R. Scott and S. Kosslyn (Eds). John Wiley & Sons, Inc. New York, NY.

Schaafstal, A., Shraagen, J.M., & Van Berl, M. (2000). Cognitive Task Analysis and Innovation of Training: The Case of Structured Troubleshooting. In *Human Factors: The Journal of the Human Factors and Ergonomics Society*. 42, 75-86.

Schreckengost, D., Bonasso, R. P., Kortencamp, D., Bell, S., Milam, C., & Thronesbery, C. (2008). Adjustable Autonomy with NASA Procedures. Presented at the The 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Pasadena,CA.

Schreckenghost, D., Milam, T., & Billman, D. (2014). Human performance with procedure automation to manage spacecraft systems (pp. 1–16). IEEE.

Schwartz, D. L., Bransford, J. D., & Sears, D. (2005). Efficiency and innovation in transfer. In J. Mestre (Ed.), *Transfer of learning from a modern multidisciplinary perspective* (pp. 1–51). Greenwich CT: IAP

Van Der Meij, H., Blijleven, P., & Jansen, L. (2003). What makes up a procedure? In M. J. Albers & B. Mazur (Eds.), *Content & complexity* (pp. 129–186). Mahwah, NJ: Lawrence Erlbaum.

Wiedenbeck, S. (1989). Learning iteration and recursion from examples. *International Journal of Man-Machine Studies*, *30*(1), 1–22.

Wittwer, J., & Renkl, A. (2010). How effective are instructional explanations in example-based learning? A meta-analytic review. *Educational Psychology Review, 22*, 393-409.