Scaffolding Meta-Cognitive Skills for Effective Analogical Problem Solving via Tailored Example Selection

Kasia Muldner, Department of Psychology, Arizona State University, USA
Katarzyna.Muldner@asu.edu

Cristina Conati, Department of Computer Science, University of British Columbia, Canada
conati@cs.ubc.ca

Abstract. Although worked-out examples play a key role in cognitive skill acquisition, research demonstrates that students have various levels of meta-cognitive abilities for using examples effectively. The Example Analogy (EA)-Coach is an Intelligent Tutoring System that provides adaptive support to foster meta-cognitive behaviors relevant to a specific type of example-based learning known as analogical problem solving (APS), i.e., using examples to aid problem solving. To encourage the target meta-cognitive behaviors, the EA-Coach provides multiple levels of scaffolding, including an innovative example-selection mechanism that chooses examples with the best potential to trigger learning and enable problem solving for a given student. To find such examples, the mechanism relies on our novel classification of problem/example differences and associated hypotheses regarding their impact on the APS process. Here, we focus on describing (1) how the overall design of the EA-Coach in general, and the example-selection mechanism in particular, evolved from cognitive science research on APS; (2) our pilot evaluations and the controlled laboratory study we conducted to validate the tutor’s pedagogical utility. Our results show that the EA-Coach fosters meta-cognitive behaviors needed for effective learning during APS, while helping students achieve problem-solving success.

Keywords. Example-based learning, cognitive modeling, decision-theoretic tutoring, meta-cognitive support, self-explanation, min/max analogy

INTRODUCTION

An example is a problem whose solution is given to the student, along with the solution’s derivation. Research shows that students rely heavily on examples during problem solving when learning a new skill (Anderson, Farrell et al., 1984; Pirolli & Anderson, 1985; Reed, Dempster et al., 1985; LeFevre & Dixon, 1986; Novick, 1995; VanLehn, 1996; VanLehn, 1998; Renkl, Atkinson et al., 2003). Furthermore, there is evidence that examples are more effective aids to problem solving than general procedures alone (Reed & Bolstad, 1991) or hints on the instructional material (Ringenberg & VanLehn, 2006). However, research also indicates that an example’s pedagogical effectiveness strongly depends on a student’s ability to apply relevant meta-cognitive skills (VanLehn, 1998; VanLehn, 1999). Meta-cognition refers to “one’s knowledge concerning one’s own cognitive processes and products or anything related to them” (Flavell, 1976); more informally, meta-cognition has been referred to as “thinking about thinking”. Meta-cognitive skills are therefore domain-
independent abilities that are an important aspect of knowing how to learn in general. Here, we focus on the meta-cognitive skills needed to learn from a specific type of example-based activity, namely *analogue problem solving* (APS), i.e., using examples to aid problem solving. These meta-cognitive skills include:

- **Min-analogy**: solving the problem on one’s own as much as possible instead of by copying from examples, and referring to examples only to resolve problem-solving impasses or to check one’s problem solution (VanLehn & Jones, 1993; VanLehn, 1998).
- **Explanation-based learning of correctness (EBLC)**: deriving new domain principles via a form of self-explanation, i.e., the process of explaining instructional material to oneself (Chi, Bassok et al., 1989; Bielaczyc, Piroli et al., 1995). EBLC involves relying on one’s existing commonsense or overly-general knowledge to explain how an example solution step is derived (Chi & VanLehn, 1991; VanLehn, 1992; VanLehn, 1999).

Min-analogy and EBLC are complementary skills: min-analogy allows students to strengthen their knowledge through practise and discover knowledge gaps, while EBLC can be used to fill the gaps. Unfortunately, as is the case with other instructional activities, during APS some students prefer more shallow processes that hinder learning, such as copying as much as possible from examples without any proactive reasoning on the underlying domain principles (e.g., (VanLehn, 1998; VanLehn, 1999)). To help all students benefit from APS, we have been working on devising adaptive computer-based support, delivered by an Intelligent Tutoring System (ITS) we refer to as the Example Analogy (EA)-Coach. While some of this tutor’s scaffolding for meta-cognition is embedded in its interface, the primary form comes from its example-selection mechanism. This mechanism selects examples in the EA-Coach’s target domain of introductory Newtonian physics that have the best potential to stimulate min-analogy and EBLC in a given student during APS.

To select appropriate examples, a key factor that needs to be taken into account is the similarity between the problem the student is working on (target problem from now on) and a candidate example. If the example is too different from the target problem, both learning and problem solving are blocked (Novick, 1995). Given this, existing ITSs supporting APS select the most similar example (e.g., (Weber, 1996b)). However, while highly-similar examples help students generate the problem solution, they may interfere with learning because they enable shallow strategies such as copying (Reed, Dempster et al., 1985). We argue that certain problem/example differences foster learning by promoting min-analogy and EBLC, while enabling problem-solving success. This hypothesis is embedded into the EA-Coach’s example-selection mechanism, and here we describe how we validated it in a controlled experiment we conducted.

Our work brings the following three contributions:

1. We provide a novel ITS that fosters the meta-cognitive skills needed for effective APS by selecting examples with different levels of similarity to the problem, tailored to a student’s needs. To date, ITSs supporting APS via example selection have chosen an example with a solution most similar to the target problem’s solution, and did not foster meta-cognitive skills (e.g., (Weber, 1996b; Guin-Duclosson, Jean-Duabias et al., 2002)). Although some ITSs do target meta-cognitive skills (e.g., (Conati & VanLehn, 2000; Aleven, McLaren et al., 2006)), none do so during APS and so their design does not need to account for an example’s impact on problem solving.
2. We provide a probabilistic user model that infers how problem/example similarity and student characteristics impact learning and problem-solving outcomes during APS. Although there has been substantial work on devising probabilistic user models to infer learning and/or problem solving outcomes (Pek & Poh, 2000; Mayo & Mitrovic, 2001; Conati, Gertner et al., 2002; Murray, VanLehn et al., 2004; Ting, Zadeh et al., 2006), none of these models are suitable for the type of instructional activities targeted by the EA-Coach.

3. Through the evaluation of the EA-Coach, we provide preliminary insight regarding the impact of problem/example similarity on students’ APS behaviors and subsequent learning and problem-solving outcomes. While there is work in cognitive science on APS (e.g., (Dempster et al., 1985; Novick, 1995)), it does not address how problem/example similarity influences min-analogy and EBLC.

Our hypotheses on the impact of problem/example similarity were presented in (Conati, Muldner et al., 2006) and a condensed version of their evaluation was discussed in (Muldner & Conati, 2007). Here, we present (1) the rationale behind the hypotheses, including how we derived them from established cognitive theories of learning from examples; (2) substantially more detail on the EA-Coach evaluation, including pilot evaluations and analysis and discussion not included in (Muldner & Conati, 2007).

This paper is organized as follows. We begin by describing the cognitive science research on APS and then show how we leveraged this research to generate our own hypotheses regarding the impact of problem/example differences on APS outcomes. We next describe our pilot evaluations, and then present the EA-Coach and its evaluation. Finally, we present the related work from the ITS community and conclude with a discussion of future work.

COGNITIVE SCIENCE BACKGROUND

APS can be characterized by two phases: selection and application. During the selection phase, students need to choose an example that is similar to the target problem and therefore helpful during problem solving. Problem/example similarity is typically characterized via two dimensions, superficial and structural (Chi, Feltovich et al., 1981; Novick, 1988; Bassok, Wu et al., 1995), defined as follows: (1) superficial similarity is determined by comparing features that appear in the problem/example specifications and/or solutions but are not part of the domain knowledge needed to derive the respective solutions; (2) structural similarity is determined by comparing the domain principles needed to derive the problem/example solutions.

If an example is not structurally similar to the target problem then it is not appropriate for APS, because its solution cannot be applied to generate the problem solution. Unfortunately, some students rely only on superficial similarity during example selection, without considering structural similarity, which hinders their ability to find appropriate examples during APS. For instance, Novick and colleagues (Novick, 1988; Novick & Holyoak, 1991) showed that when ‘distracter examples’ are available that are only superficially similar to the target problem, students select them over structurally appropriate but superficially dissimilar examples. Students also fail to recognize when a structurally appropriate example is helpful, if the example is not superficially similar to the target problem.
Reliance on superficial similarity during example selection is particularly common among students with low domain expertise (Novick, 1988).

Once an example is selected, its solution needs to be applied to generate the target problem’s solution (application phase). How students do so and what they learn as a result has been the topic of substantial research, as we now describe.

The ACT-R Analogy Mechanism

John Anderson’s Atomic Components of Thought-Rational (ACT-R), a theory of human cognition, includes the ‘analogy mechanism’ as a fundamental means of human learning (Anderson & Thomson, 1989; Anderson, 1993). The mechanism, derived from studies on APS in the LISP programming domain, encodes how to use an example to generate a problem solution, and how learning occurs from doing so. As is illustrated in Figure 1(A), using the example to generate the problem solution involves: (1) mapping: finding a correspondence between the problem/example constants in the respective specifications; (2) application of mapping: transferring the example solution over to the problem, relying on the mapping to replace example-specific constants by those needed for the problem solution. The mapping and application processes are generalized by creating a domain rule, as follows. The rule’s antecedent is the conjunction of the problem/example specification elements, while its consequent is the example solution; all constants involved in the mapping phase are replaced by variables (illustrated in Figure 1(B)).
Impact of Meta-Cognitive Skills on APS

VanLehn’s work on the role of examples in skill acquisition investigates how learning outcomes are influenced by individual differences between students, and particularly by the meta-cognitive skills that students bring to bear (VanLehn & Jones, 1993; VanLehn, 1998; VanLehn, 1999). The work is based on protocol analysis of students studying physics examples and solving problems with access to the examples.

During APS, students have the choice of generating the problem solution on their own or by copying, i.e., transferring the example solution (or part of) over to the problem with minor or no changes, such as (1) revising the order of terms in the equations of the transferred example step; (2) minor omissions, e.g., leaving out brackets; (3) using different variable names to denote problem solution elements; (4) replacing example-specific constants with ones needed to solve the problem, a process VanLehn refers to as *transformational analogy* (see Figure 2). Note that transformational analogy is analogous to ACT’s mapping/application processes, but while it is associated with learning in ACT-R, in VanLehn’s work it is associated with a lack of learning.

How much a student copies during APS characterizes a meta-cognitive skill that VanLehn identifies as relevant to APS, namely *min-analogy*. Min-analogy entails generating the problem solution through the application of one’s own knowledge rather than by copying, relying on examples primarily to overcome impasses that block problem-solving progress. Min-analogy is beneficial to learning for two reasons. First, if students solve problems on their own without copying, they strengthen their existing domain knowledge through practice. Second, min-analogy provides students with opportunities to encounter problem-solving impasses, which expose missing knowledge, i.e., knowledge gaps. Gap discovery is an important prerequisite to learning of domain principles, since once a student becomes aware of a gap, she may take action to repair it. Unfortunately, although some
students prefer min-analogy, others have the opposite tendency, i.e., a max-analogy tendency. These students maximize copying without trying to generate the problem solution on their own, even if they have the knowledge to do so, and so miss the opportunity to strengthen their knowledge and/or discover their knowledge gaps.

Clearly, gap discovery is not sufficient to learn new domain principles. VanLehn postulates that learning occurs through a form of self-explanation referred to as Explanation-based learning of correctness (EBLC) (Chi & VanLehn, 1991; VanLehn, 1992; VanLehn, 1999). EBLC is a metacognitive skill that involves using one’s existing commonsense and overly-general knowledge, in conjunction with domain knowledge, to infer new rules that explain how a given example solution step is derived. Note that the definitions of self-explanation in general and EBLC in particular do not require an inference to be correct in order to be classified as a self-explanation (or EBLC); in fact, students may require several attempts before a correct principle is derived. Figure 3 shows an example provided in (VanLehn, 1999) of how EBLC can be used to explain the existence of the normal force mentioned in example solution step 3, Figure 2.

Unfortunately, some students miss learning opportunities altogether because they do not have a tendency for EBLC. These students resolve impasses during APS by resorting to strategies not as conducive to learning to generate the problem solution, such as guessing or copying from available examples. Note that min-analogy and EBLC are complementary skills, and that having one skill does not guarantee the presence of the other. For instance, a min-analogy student will strengthen her knowledge and discover her knowledge gaps, but this does not guarantee she will use EBLC to fill the gaps.

---

1 We will use a mix of feminine and masculine pronouns.

---

<table>
<thead>
<tr>
<th>Commonsense rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>If an object $O_1$ supports object $O_2$ then there is a force $F$ on $O_1$ due to $O_2$.</td>
</tr>
<tr>
<td>Since the ramp supports the crate, there is a force on the ramp applied by the crate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overly general rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $F$ is a force inferred by common sense reasoning then this force is an official physics force.</td>
</tr>
<tr>
<td>$F$ is an official physics force.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Newton’s Third Law rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>If an object $O_1$ exerts a force on object $O_2$ then $O_2$ exerts an equal and opposite force on $O_1$.</td>
</tr>
<tr>
<td>There is a reaction force exerted by the ramp on the crate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Normal-exists rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>If an object $O_1$ is supported by object $O_2$ then there is a normal force on $O_1$ due to $O_2$.</td>
</tr>
<tr>
<td>There is a normal force on $O_1$ due to $O_2$.</td>
</tr>
</tbody>
</table>

Fig. 3. Rule inferred via EBLC from step 4 of the physics problem/example in Figure 2 (abstract rule and student reasoning shown on left and right, respectively).
**EA-Coach Target Learning Mechanism.** ACT-R’s analogy mechanism is appropriate for learning from simple examples, such as the LISP examples on which the mechanism is based. Based on our analysis, however, with more complex multi-step examples, rules inferred by the analogy mechanism tend to be overly specific and do not reflect a true understanding of the underlying domain principles. This limitation is illustrated in Figure 4 (top) by the ‘Normal-exists’ rule inferred through the ACT-R analogy mechanism from the problem/example pair in Figure 2. Note how this rule reflects at best a shallow understanding of the normal force, because its structure is too closely tied to the problem/example specifications from which it is derived. In contrast, the rule inferred by EBLC (Figure 4, bottom) reflects the kind of knowledge students should possess with respect to the normal force. Thus, EBLC is the more appropriate learning mechanism for the EA-Coach domain of introductory Newtonian physics, and so is its target learning mechanism. Similarly, we adopt VanLehn’s view that although transformational analogy may enable problem solving during APS, it hinders learning.

**Impact of Problem/Example Differences on Example Application**

As we mentioned above, problem/example differences are commonly classified according to two similarity dimensions, *structural* and *superficial*. Reed, Ackinclose et al. (1990) compared the impact of two kinds of structural differences on students’ ability to apply the example solution to generate the target problem solution: (1) the problem solution includes steps derived by domain principles not in the example’s solution, (2) the example solution includes steps derived by rules not in the problem’s solution. Reed showed that the first type of structural difference impedes students’ ability to generate the problem solution more than the second type of difference. Several other studies have shown that the first type of structural difference hinders the application phase (Reed, Dempster et al., 1985; Reed, 1987; Novick, 1995); Novick (1995) found that this is particularly the case for students with low domain expertise.

There is also research exploring how superficial differences influence example application. Quilici & Mayer (1996) found that when students are given *several* examples to help solve a problem, they learn better when the examples are *not* superficially similar to each other. However, Quilici & Mayer did not explore the impact of varying the superficial similarity between the problems and the examples. This was investigated in other work (Ross 1987; Reed, Dempster et al., 1985); the results show that students are more successful in generating a problem solution when problem/example
superficial similarity is high, as compared to low, but that neither low or high similarity triggers learning of the underlying domain principles.

**LEVERAGING COGNITIVE SCIENCE FINDINGS: IMPACT OF PROBLEM/EXAMPLE DIFFERENCES ON APS**

Although research clearly shows that problem/example differences have an impact on APS, there is not yet a full understanding of how various kinds of differences influence APS behaviors and subsequent learning and problem-solving outcomes. To design the EA-Coach support, it was therefore necessary to formulate our own hypotheses, which we based directly on the cognitive science background just presented. Before we can present our hypotheses, however, we need to introduce some terminology.

**Terminology**

The terminology we present is based on the established approach of classifying problem/example relations as *superficial/structural*, but we extend this work by proposing a novel classification of superficial differences.

We consider a pair of problem/example steps as *structurally identical* if the steps are generated by the same rule, and *structurally different* otherwise. Two structurally identical steps may include different constants and so be *superficially different*. This is illustrated in Figure 5, which shows a fragment of the problem and example originally shown in Figure 2. For each problem and example solution step, Figure 5 shows a portion of the step’s textual description and the rule used to derive that step. Since problemStep3 and exampleStep3 are generated by the same rule (*Normal-force*), they are structurally identical. Likewise, problemStep4 and exampleStep4 are generated by the same rule (*Normal-dir*) and so are structurally identical. However, both problemStep3 and problemStep4 are superficially different from their corresponding example steps, in the following ways:

- **problemStep3** and exampleStep3 are superficially different with respect to both the object a normal force exists on (*block* vs. *crate*) and the object the force is due to (*floor* vs. *ramp*)
- **problemStep4** and exampleStep4 are superficially different with respect to the angle specifying the normal force (*90°* vs. *120°*)

Our approach entails classifying the superficial differences according to how easily they can be resolved: through in-depth reasoning such as EBLC or by copying from the example (e.g., via transformational analogy). Returning to our scenario, exampleStep1 can be copied to generate a correct step in the problem solution, because the two superficial differences between it and the corresponding problemStep3 can easily be resolved by transformational analogy. This is because both differences between problemStep3 and exampleStep3 correspond to constants that appear in the problem/example specifications, enabling the mapping necessary for transformational analogy on these steps. We classify such differences as *trivial*. In contrast, the superficial difference between problemStep4 and exampleStep4 cannot be easily resolved by transformational analogy because the example constant corresponding to the difference is missing from the problem/example specifications, and so a student
cannot generate the necessary mapping between the relevant constants. We classify such differences as non-trivial. The formal definitions corresponding to our classification, which the EA-Coach relies on to assess problem/example similarity, may be found in (Muldner, 2007).

Impact of Differences: Hypotheses

Given our classification of superficial and structural relations, the key question is: What impact do various kinds of differences have on APS behaviors and subsequent problem solving and learning? We argue that this impact depends on a student’s APS meta-cognitive skills and domain knowledge.

If the problem and example are structurally different with respect to a problem step, then a student cannot rely on the example to derive it. This hinders both problem solving and learning if the student lacks the knowledge to generate the problem step on his own. If, on the other hand, the student does have the knowledge to generate the step on his own, then the structural difference encourages min-analogy by blocking copying, which will help to solidify the student’s knowledge of the corresponding rule.

In contrast, superficial differences between structurally identical problem/example steps do afford students the opportunity to apply the example to generate the problem step, because the two steps are derived by the same rule that a student can infer and subsequently apply. However, as we already pointed out, what distinguishes trivial vs. non-trivial differences is how easily they may be resolved. Because trivial differences are easily resolved by transformational analogy, they encourage copying instead of learning for students with poor APS meta-cognitive skills. Copying increases the chances...
that a student will generate the problem solution, but it means she misses the opportunity to strengthen
her knowledge through practice, or fill knowledge gaps via EBLC.

On the other hand, because non-trivial differences are not easily resolved by transformational
analogy, they may encourage min-analogy and EBLC. To see how this could occur, let’s consider
the non-trivial difference between \(\text{exampleStep}_4\) and \(\text{problemStep}_4\) in Figure 5. Since the direction of the
normal force in \(\text{exampleStep}_4\) does not appear in the example specification, a student cannot generate
the problem solution step by copying from the example, because he cannot use transformational
analogy to resolve the difference between the problem/example steps. Consequently, if the student
knows the \(\text{Normal-dir}\) rule needed to generate this step, then the non-trivial difference may encourage
him to generate the step via the application of his own knowledge, i.e., to engage in min-analogy. If,
on the other hand, the student does not know the \(\text{Normal-dir}\) rule, then the non-trivial difference mayencourage him to explain through EBLC how the example step was generated, i.e., to learn the rule. If
the student is successful, then he can apply the newly-acquired knowledge to generate the solution
step, thereby resolving the difference between the problem and example on this step.

PILOT EVALUATIONS: DATA ON HYPOTHESES AND REFINING THE
EA-COACH INTERFACE

In order to see if we were on the right track in terms of our hypotheses presented above, we conducted
a pilot evaluation (primary pilot from now on), through which we gathered some first-hand
observations on how the various types of superficial differences in our classification influence APS.
We now describe this primary pilot, as well as subsequent pilots that we used to refine the EA-Coach
interface.

Preliminary EA-Coach Interface. In the primary pilot, eight students used a preliminary version of the
EA-Coach interface to solve problems and access examples in our target domain of Newtonian
physics (see problem and example windows, Figure 6a and 6b). This interface was based on two
existing ITSs for Newtonian physics: (1) Andes, which supports pure problem solving without
providing access to examples (VanLehn, Lynch et al., 2005), and (2) the SE-Coach, which supports
pure example studying without providing problems to solve (Conati & VanLehn 2000). Since the EA-
Coach supports APS, i.e., problem solving with access to examples, and the Andes and SE-Coach
interfaces had undergone extensive usability testing, we based the preliminary EA-Coach interface on
their respective designs.

The problem window in the preliminary EA-Coach interface (see Figure 6a) allowed students to
draw free-body diagrams and/or type equations. The window’s design was directly based on the Andes
design, the primary exception being that the Andes interface provides feedback for correctness. The
preliminary EA-Coach interface did not provide such feedback, for two main reasons. First, although
cognitive science work indicates that students are not always effective at diagnosing their own
misconceptions (Chi, Bassok et al., 1989), we wanted to gather first-hand information on how this
diagnosis occurred in our particular instructional context. Second, we wanted to investigate APS
behaviors in a context that was similar to existing cognitive science research (e.g., (VanLehn, 1998)),
which does not provide such feedback.
The example window in the preliminary EA-Coach interface (see Figure 6b) was directly based on the SE-Coach design. The example was presented using a format loosely based on the one used in physics textbooks. A masking interface covered the example specification and solution (moving the mouse over a region uncovers it and covers whatever region was previously uncovered). This interface serves two functions. First, it provides information to the tutor on which example steps students view. Second, in the context of the EA-Coach, the masking interface is a form of scaffolding for the metacognitive skills of min-analogy and EBLC, by (1) discouraging copying, because of the effort needed to explicitly uncover the example solution, and (2) encouraging EBLC, by helping to focus students’ attention on individual example solution steps. A second form of scaffolding we introduced to discourage copying corresponds to the lack of ‘Copy’ and ‘Paste’ functionality between the example and problem windows. This design is based on findings from an earlier pilot involving an interface that allowed cutting and pasting; some students abused these functionalities to copy entire example solutions.

Primary Pilot Conditions. In the primary pilot, each student used the above-described interface to solve two problems, with access to one corresponding example for each problem (opening a problem also opened the example). The problems and examples were based on typical “Newton’s Second Law” problems used in physics courses (as identified through on-line searches and textbooks, e.g., (Halliday & Resnick, 1988)). We manipulated the superficial similarity between the problem and its corresponding example so that for one problem, its example only included trivial superficial differences (trivial condition), while for the other problem, its example included both trivial and non-trivial superficial differences (non-trivial condition). The two problem/example pairs were statically hand-paired before run time and all students saw the same problem/example pairs.
Primary Pilot Results. Here, we present highlights of the primary pilot results; full details may be found in (Muldner, 2007). We found that students copied more in the trivial condition, as compared to the non-trivial condition (on average, 3.5 vs. 2.5 copy events; to identify copy events, we used the method described in the ‘Evaluation of the EA-Coach’ Section). This result is in accordance with our hypothesis that non-trivial differences should discourage copying. On the other hand, the pilot revealed that non-trivial differences lowered problem-solving success: students’ solutions had more errors in the non-trivial condition, as compared to the trivial condition (on average, 1.75 vs. 0.63 errors). Furthermore, contrary to what we hypothesized, non-trivial differences did not encourage students to self-explain more than trivial differences – the number of self-explanations was approximately equal between the conditions\(^2\) (on average, 1 vs. 0.75 in the trivial vs. non-trivial conditions; to identify self-explanation, we used the method described in the ‘Evaluation of the EA-Coach’ Section). We believe that a likely explanation for both of these findings, however, is related to the lack of feedback for correctness in the pilot. In our study, students did not provide any indication of being aware of the errors in their final problem solutions. Feedback for correctness would make the errors explicit and so encourage students to fix them, which could have two beneficial side effects: (1) increased problem-solving success and (2) trigger for EBLC, which is needed to resolve errors corresponding to knowledge gaps. Consequently, as our next step, we incorporated immediate feedback for correctness into the EA-Coach, realized by coloring student entries green or red for correct vs. incorrect entries, respectively. This feedback is another type of scaffolding for APS, since as we argue above, it may foster EBLC. Feedback for correctness may also encourage min-analogy. To illustrate, suppose a student with a tendency for max-analogy is generating the problem solution by indiscriminately copying from an example that includes some differences blocking ‘correct’ copying of its solution. Immediate feedback for correctness can make the student aware of the incorrectly-copied steps and so discourage excessive copying by highlighting its limitations.

Subsequent Pilots to Refine the EA-Coach Interface. The addition of feedback for correctness meant that students now had to include appropriate variable names in their equations generated in the EA-Coach problem window (this requirement is a function of the system’s algorithm for assessing correctness). To support students in doing so, we added a ‘variable definition’ pane to the problem-solving window, following the Andes design (VanLehn, Lynch et al., 2005) (see Figure 7a, top left). The subsequent pilot involving three students showed the need for two key refinements. First, subjects felt that the example window’s design was not sufficiently similar to the problem window’s. Consequently, we changed the example format to more closely mirror the problem format (see Figure 7b). Second, after we observed a pilot subject struggling because of syntax issues with her equation, we added one more layer of scaffolding to the system’s feedback, corresponding to having the EA-Coach inform students when their entry could not be interpreted due to syntactic errors (such as incorrect variable names). The next pilot involved four subjects and did not indicate the need for any

\(^2\) In the pilot we identified instances when students self-explained in general, without analyzing if the explanation included EBLC-style reasoning used to derive new rules. Since EBLC is a form of self-explanation, we felt this would provide initial data on how students reasoned during our pilot.
further refinements. The final EA-Coach interface is shown in Figure 7 (note that the masking interface that covers the example window as seen in Figure 6 is not shown in Figure 7, but is part of the final interface).

THE EA-COACH ARCHITECTURE AND EXAMPLE-SELECTION MECHANISM

Above, we described how the EA-Coach interface evolved from our pilot evaluations, culminating in the final version shown in Figure 7. We now present the remainder of the EA-Coach framework. Since cognitive science research indicates that students experience difficulties during example selection and application (e.g., (Novick, 1988; VanLehn 1998)), the EA-Coach 1) takes over the responsibility of example selection, to ensure that each student has access to appropriate examples, tailored to her needs, and 2) provides interface scaffolding to encourage students to use the examples effectively.

Note that the tutor is designed to complement rather than replace traditional classroom instruction (i.e., students use the EA-Coach to refine their domain knowledge obtained via regular curricular activities).

The EA-Coach Architecture

To automatically assess the impact of an example on a student’s APS behaviors, including how she will generate the problem solution and what she will learn from it, the EA-Coach needs a formal representation of the problems and examples in the target domain. Furthermore, to tailor the instructional support, the tutor needs a mechanism to model and monitor students’ domain knowledge
and meta-cognitive skills. These requirements are implemented in the EA-Coach architecture, shown in Figure 8.

The knowledge base contains a rule-based representation of the Coach’s target domain of Newtonian physics. The problem/example pools are populated before run-time with (1) the problem/example specifications and (2) solutions derived by the problem solver, an ‘off-the-shelf’ forward-chaining expert system. The solver uses a problem or example specification and the rules in the knowledge base to automatically generate the corresponding solution, which is stored in a structure called the solution graph. The solution graph is a dependency network representing how each solution step derives from previous steps and rules in the knowledge base – for details, see (Conati, Gertner et al. 2002). At run-time, students use the interface (see Figure 7) to solve problems and refer to examples in the problem/example pools. The examples students use are dynamically selected for them by the example-selection mechanism. The mechanism assesses the similarity between a problem and example by comparing their specifications and solutions, and leverages that with information on a student’s domain knowledge and meta-cognitive skills to choose the optimal example for her, as we now describe.

**The EA-Coach Example-Selection Mechanism**

When a student asks for an example, the EA-Coach example-selection mechanism chooses one from the example pool that best meets two objectives: (1) helps generate the target problem solution (problem-solving success goal), and (2) triggers learning by encouraging the meta-cognitive skills of min-analogy and EBLC (learning goal). To meet these two objectives, the mechanism quantifies the suitability of a candidate example via a two-phase decision-theoretic process: simulation and expected utility (EU) calculation (Muldner & Conati, 2007).
Simulation Phase

Simulation involves generating a prediction of how a student will solve a problem in the presence of a candidate example and what she will learn from doing so. This prediction is generated by the EA-Coach user model, which corresponds to a dynamic Bayesian network (Dean and Kanazawa, 1989). The network is automatically created when a student opens a problem and is based on that problem’s solution graph, following the approach in (Conati, Gertner et al., 2002). The network includes nodes and links representing how the various problem solution steps (shaded rectangular nodes in Figure 9) can be derived from domain rules (shaded elliptical nodes in Figure 9) and other steps. To illustrate, the simplified fragment of the user model in Figure 9, slice $t$ (pre-simulation slice) shows how the solution steps normal-force and normal-dir are derived from the corresponding rules and/or other steps (assuming the problem is the one in Figure 5). The network also contains nodes to model a student’s min-analogy and EBLC tendencies ($MinAnalogyTend$ and $EBLCTend$ nodes in slice $t$, Figure 9). Unless otherwise stated, all network nodes have Boolean values, as follows: (1) the step nodes represent the probability the student will generate that step in the EA-Coach interface, (2) the rule nodes represent the probability the student will learn that rule via EBLC, and (3) the tendency nodes represent the probability the student has the tendency for min-analogy or EBLC.

To model the impact of a candidate example, the framework automatically adds a special ‘simulation’ slice to the network (slice $t+1$, simulation slice, Figure 9, assuming the candidate example is the one in Figure 2, later shown in Figure 5). This slice contains all the rule and step nodes in the pre-simulation slice, as well as additional nodes that are included for each problem-solving action being simulated and account for the candidate example’s impact on APS, as follows: (1) $Similarity$ nodes, encoding the similarity between a problem solution step and the corresponding example step (if any); (2) $Copy$ nodes, encoding the probability that the student will generate the problem step by copying the corresponding example solution step (3) $EBLC$ nodes, encoding the probability that the student will infer the corresponding rule from the example via EBLC.

During the simulation phase, the only form of direct evidence for the user model corresponds to the similarity between the problem and candidate example (see $Similarity$ nodes, slice $t+1$, Figure 9). The framework relies on our definitions of problem/example differences presented above to set each solution step’s corresponding similarity node value to either: None (structural difference because the problem and corresponding example step are derived by different rules), Trivial or Non-trivial. Since the introduction of this evidence into the dynamic Bayesian network results in the propagation of belief in the network, similarity nodes are instrumental in allowing the model to generate a fine-grained prediction of copying and EBLC. This prediction in turn impacts the model’s prediction of learning and problem-solving success.

To illustrate, consider the node $Step_{normal\text{-}force}$ in slice $t+1$, Figure 9. The probability that a student will generate this step by copying is high (see node $Copy_{n\text{-}force}$, slice $t+1$), because the problem/example similarity allows for it ($Similarity_{n\text{-}force}$=Trivial, slice $t+1$) and the student has a tendency to copy (indicated in slice $t$ by the low probability of the $MinAnalogyTend$ node). In contrast, the probability that the student will generate $Step_{normal\text{-}dir}$ by copying is very low (see node $Copy_{n\text{-}dir}$ in slice $t+1$) because the non-trivial difference ($Similarity_{n\text{-}dir}$=Non-trivial, slice $t+1$) between the problem step and corresponding example step blocks copying. As another example, consider node $EBLC_{n\text{-}force}$. The model predicts that the student is not likely to reason via EBLC to derive $Step_{normal\text{-}force}$ (see node $EBLC_{n\text{-}force}$ in slice $t+1$) even though she does need to learn this rule (see low probability of node
This is because of the high probability that she will copy the step (see node $Copy_{n,force}$) and the moderate probability of her having a tendency for EBLC (see node $EBLCTend$ in slice t). In contrast, a low probability of copying (see node $Copy_{n,dir}$, slice t+1) increases the probability for EBLC reasoning (see node $EBLC_{n,dir}$ in slice t+1), but the increase is mediated by the probability that the student has a tendency for EBLC, which in this case is moderate. The model’s prediction of EBLC and copying behaviors influences its prediction of learning and problem-solving success (see Step and Rule nodes in Figure 9, slice t+1). For instance, learning is predicted to occur if the probability of a rule being known is low in the pre-simulation slice and the simulation predicts that the student will reason via EBLC to learn the rule (see node $Rule_{normal,dir}$, slice t+1).

The DBN structure and parameters are based on relevant APS literature and represent our best estimate of the impact of factors such as student characteristics/similarity on the variable under consideration. For instance, the network does not currently include a link between copying and knowledge. Although it may be the case that knowledge somehow mediates student copying along with analogy tendency, this has yet to be investigated, and thus for the time being our network does not include that link. Future work will involve refining the network as needed, for instance via sensitivity analysis and/or machine learning (e.g., learning of the conditional probability table...
parameters from data, for instance with Expectation Maximization (EM) (Dempster, Laird et al., 1977)).

**Expected Utility (EU) Calculation Phase**

The outcome of the simulation is used to assign an expected utility to a candidate example, quantifying its ability to meet the learning and problem-solving success objectives. Specifically, the framework relies on a decision-theoretic approach, where the probabilities of the *rule* and *step* nodes in the user model are used as inputs to a multi-attribute linearly-additive utility model. This model calculates an example’s expected utility by weighting the utility of each outcome (learning, problem-solving success) by the probability that it will occur. Currently, we hand-select the utility function and corresponding weights, an approach also used by other ITS (Murray et al., 2004; Mayo and Mitrovic, 2001).

To illustrate the utility calculation process, to obtain the utility of learning, first the EA-Coach calculates the expected utility (EU) for learning each individual rule in the problem solution. This is the sum of the probability $P$ of each possible learning outcome for that rule (i.e., value of a rule node), multiplied by the utility $U$ of that outcome:

$$EU(\text{Rule}_i) = P(\text{known}(\text{Rule}_i)) \cdot U(\text{known}(\text{Rule}_i)) + P(\neg\text{known}(\text{Rule}_i)) \cdot U(\neg\text{known}(\text{Rule}_i))$$

Since we define $U(\text{known}(\text{Rule}_i)) = 1$ and $U(\neg\text{known}(\text{Rule}_i)) = 0$, the expected utility of a rule corresponds to the probability that the rule is known. An example’s overall expected utility for learning is calculated as the weighted sum of the expected utilities for learning the individual rules:

$$\sum_{i} EU(\text{Rule}_i) \cdot w_i$$

Currently all the weights $w$ are assigned an equal value (i.e., $1/n$, where $n$ is the number of rules in the network), representing the assumption that all rules are equally important. The same approach is used to obtain the measure for the problem-solving success objective.

This two-phase process (simulation, EU calculation) is repeated for each example in the EA-Coach’s example pool, and the example with the highest expected utility is presented to the student. Our approach for example selection allows the EA-Coach to take into account problem/example similarity, as well as a student’s knowledge and meta-cognitive skills, during the example-selection process, and thereby tailor the choice of an example to a given student’s needs. Once an example is presented to the student, the EA-Coach user model generates an assessment of how a student’s knowledge and EBLC/min-analogy tendencies evolve as a result of using the example (Muldner and Conati, 2005; Muldner 2007). When the student closes the target problem, the posteriors for the rule and tendency nodes in the last slice of the DBN are stored in a long-term model (not shown in Figure 8); when he opens a new problem, these posteriors become the priors for the corresponding rule and tendency nodes in the new problem’s DBN. This process allows the student model to have up-to-date information on the student each time he opens a problem to work on.
EVALUATION OF THE EA-COACH

We now describe the laboratory study we conducted to evaluate the pedagogical effectiveness of the EA-Coach, and in particular its example-selection mechanism. To verify how well the mechanism meets its example selection goals, we compared it with the standard approach taken by ITSs that support APS, i.e., selecting the most similar example (Weber, 1996b; Alevén & Ashley, 1997; Nogry, Jean-Daubias et al., 2004). To compare how the two example-selection approaches influence learning, we focused on analyzing how each approach influences APS behaviors that impact learning outcomes, i.e., min-analogy and EBLC. The approach of assessing learning by performing a fine-grained analysis of behaviors that impact learning outcomes is advocated in (Chi, Bassok et al., 1989). Although this approach makes the analysis challenging because it requires that subjects’ reasoning is captured and analyzed, it does have the advantage of providing in-depth insight into how each example-selection strategy influences learning via the relevant meta-cognitive processes. Another way to measure learning is through pre/post test differences. While we did include this measure in our analysis, it was not the focus of our evaluation, primarily because this approach does not provide information on the learning process, i.e., it does not tell us why learning did or did not occur.

Study Participants

The participants were university students who were taking or had completed a course equivalent to a high-school grade twelve physics class (but had not taken any higher-level physics courses). Subjects were paid ten dollars per hour for their participation. Since we needed information on APS behaviors, and our pilot studies showed that students who have a high level of expertise are less likely to use examples while problem solving, we pre-screened subjects based on knowledge. Specifically, subjects were given a pre-test, and those who obtained 90% or higher did not participate in the remainder of the study (11 subjects). Furthermore, subjects who did not use any examples during the experimental phase did not complete the remaining study phases (three subjects).

Study Methodology

The study methodology evolved from the pilot evaluations described above. We used a within-subject design, because it increases an experiment’s power to detect differences between conditions. This is accomplished by exposing each subject to all the conditions, thus accounting for the variability between subjects arising from individual differences in, for instance, expertise, APS tendencies and verbosity (which impacts verbal expression of EBLC). Each participant: (1) completed a pencil and paper pre-test; (2) was given a 10 minute break while we initialized the example-selection mechanism (initialization phase); (3) was introduced to the EA-Coach Interface (training phase), which took approximately 10 minutes; (4) interacted with the EA-Coach (experimental phase), which on average took about 68 minutes (with a standard deviation of about 22 minutes); (5) was given a 5 minute break; (6) completed a pencil and paper post-test (subjects were given 30 minutes for the pre and post-tests).
Experimental Phase. During the experimental phase, subjects used the EA-Coach interface (see Figure 7) to solve two physics problems (p1 and p2, see appendix). Note that as mentioned above, the EA-Coach provides feedback for correctness on subjects’ problem-solving entries, and informs subjects when it cannot interpret their entries because of syntactic problems. The system does not, however, provide any other hints or interventions related to domain-specific help or APS strategies.

To control the overall experiment time while providing adequate opportunity to generate the problem solution, subjects were given up to 60 minutes per problem (this threshold was derived from our pilot studies). Once subjects finished a problem and closed it, they were not allowed to re-open it. For each problem, subjects had the option to refer to one example (opened by clicking the ‘Get Example’ button, Figure 7). Subjects were instructed to treat this phase of the study as a homework situation, where they had some problems to solve, had access to a worked-out example, and were trying to both do their homework and prepare for an upcoming test. Subjects were told that it was up to them to use the examples and/or how to do so.

The example-selection strategy was manipulated as follows. For one of the problems, subjects had access to an example that was hand selected before run-time and was highly superficially and structurally similar to the problem (static-selection condition). In contrast, for the other problem, the example was adaptively chosen at run-time by the EA-Coach selection mechanism from the pool of examples (adaptive-selection condition). In the static-selection condition, all subjects saw the same problem/example pairs; this was not the case for the adaptive-selection condition (details on problem/example similarity in the two conditions are presented below). To account for carry-over effects, the problem and condition presentation order was fully counterbalanced, with subjects assigned to the problem/condition combinations in a round-robin fashion. Since the study involved two problems (p1 and p2) and two conditions (static and adaptive selection), counterbalancing resulted in four problem/condition combinations.

To collect data during the experimental phase, we instrumented the EA-Coach to log all student actions in its interface. We also used the talk-aloud method (Ericsson & Simon, 1980) to capture students’ reasoning by having them verbalize their thoughts. An experimenter was present in the room during the experiment and prompted subjects to speak if they became silent. We videotaped and subsequently transcribed all sessions.

Initialization Phase. During the initialization phase that followed the training phase and preceded the experimental phase, we initialized the EA-Coach example mechanism for each student. Note that the mechanism still functions without being initialized with student-specific information, but its operation is not tailored to a given student’s characteristics at the onset of the interaction. Initializing the selection mechanism corresponds to setting the priors for knowledge (i.e., rule) and APS tendency nodes in the dynamic Bayesian network user model.

To initialize the rule node priors for a given student, we used information from that student’s pre-test. The pre-test contained several questions corresponding to each relevant physics rule involved in the problem solutions in the experimental phase, i.e., rule node in the Bayesian network. For each rule node, we assigned it a prior based on the percentage of correct responses from pre-test questions corresponding to that rule. Thus, the selection mechanism had information on student knowledge, and tailored the choice of example to it, meaning that the same example was not always selected for a given problem in our study.
Obtaining information to initialize the APS tendency priors proved more challenging. We are not aware of the existence of a cognitive test to assess a student’s APS tendencies. One possibility we considered to obtain this information involved adding a third problem-solving phase to our experiment. Specifically, we would use the above-described design, but prior to the experimental phase, we would introduce a ‘baseline’ phase. In this phase, students would solve a problem with access to a statically selected example, and the EA-Coach user model would assess the students’ tendency by monitoring their APS behaviors. Recall that statically selected examples are highly similar to the target problem, allowing students to copy and/or self-explain as much or as little as they desire, thus providing an opportunity to observe their inherent tendencies. Adding this phase, however, has a critical drawback, since it means introducing another static condition that always appears first. Consequently, our experiment would no longer be counterbalanced, since the adaptive condition would never appear as the first trial. This would make it difficult to draw conclusions about the adaptive condition in isolation from the baseline phase. Furthermore, adding the baseline phase would increase overall experiment time, meaning that either we would have to conduct the experiment over several days, and risk higher attrition that accompanies such a design, or risk subject fatigue resulting from a prolonged session. Given these considerations, we decided against this option and instead chose to initialize the tendency node priors to 0.5 for all the subjects.

The model did not update this assessment of knowledge/meta-cognitive tendencies during the study. This decision was based on the following two factors: (1) it allowed us to evaluate the example-selection mechanism in isolation from the model’s assessment; (2) due to the need to counterbalance the study conditions, allowing the assessment to be updated during the study would mean that the EA-Coach example-selection mechanism had varying amounts of information about a student’s tendency (i.e., depending on whether the adaptive condition was first, or followed the static condition), which could bias the results.

Study Materials

The evaluation involved two types of materials: (1) pencil and paper pre & post tests; (2) problem and examples used in the experimental phase of the study.

The pre and post tests included the same number of problems (7 problems, some with sub-parts). The tests were based on ones used in the primary pilot study described above, but expanded to allow for a fine-grained assessment of students’ physics knowledge (we piloted this refinement in a follow-up pilot). Specifically, the problems were designed to provide information on subjects’ knowledge of individual rules in the EA-Coach knowledge base, used to initialize the example-selection mechanism. To improve accuracy, the tests included several questions per concept to account for the possibility of slips, etc. To motivate students to answer the post-test questions, we designed the pre and post tests to have equivalent problems, but we varied the constants and objects in the problem specifications.

The problems and the examples used during the experimental phase of the study (see appendix) evolved from our pilot evaluations. To help clarify the discussion of the results, we report on the level of similarity between the statically vs. adaptively selected examples and corresponding problems; all problems/examples are provided in the appendix. In the static condition, examples e₁ and e₄ were paired with problem p₁ and p₂, respectively. In the adaptive condition, the EA-Coach never chose to make this pairing. In fact, the adaptively-selected examples were always less superficially similar to
the target problem than the statically-selected ones, since (1) the total number of trivial and non-trivial differences between the problem/example was always higher in the adaptive condition than in the static condition and (2) the adaptive condition always included more non-trivial differences (see Table 1). Note that the variance in problem/example similarity in the ‘adaptive-selection’ column in Table 1 arises from the fact that during the study, the example in the adaptive condition was tailored to a given student’s knowledge and so the same example was not always chosen for a given problem. In particular, although the adaptively-selected examples always included more differences than the statically-selected ones, the selection mechanism did not always select an example with the maximum number of non-trivial differences; instead, it chose examples such that the differences corresponded to student knowledge gaps.

Dependent Measures

Our key dependent measures included: (1) Min-Analogy, analyzed by assessing copy events: the number of copy episodes for a problem solution, (2) EBLC events: the number of EBLC self-explanations expressed while generating a problem solution; (3) Learning gains from pre to post test: the percentage of rules learned; (4) Problem-solving success: the number of students who generated a correct problem solution; (5) Task time: the time to generate a problem solution; (6) Errors: the number of errors made while generating a problem solution.

Results

The key analysis technique we used corresponded to the univariate repeated-measures ANOVA. We first verified that there was no significant difference in subjects’ pre-test performance between the four groups arising from the counterbalancing of problem/condition combinations (F(3,15)=0.23, p=0.874). We then verified that problem type (p1 vs. p2) did not have an impact: there were no significant main, interaction or order effects for problem for any of the dependent variables. Consequently, we collapsed across problem and performed the analysis with selection as the within-subject factor. In our analysis, we considered selection in combination with the between-subject factors resulting from the counterbalancing of selection and problem types, referred to as selection order and problem order, respectively.

Of the 16 subjects that completed all stages of the experiment, two accessed an example in only one of the two selection conditions (one subject accessed only the statically-selected example; another accessed only the adaptively-selected example). Although we originally intended to include these subjects in our analysis, after further consideration we decided not to, since it is hard to argue that

<table>
<thead>
<tr>
<th></th>
<th>Static Selection</th>
<th>Adaptive Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean # trivial differences (st. dev.)</td>
<td>8 (0)</td>
<td>8 (.81)</td>
</tr>
<tr>
<td>Mean # non-trivial differences (st. dev.)</td>
<td>1 (0)</td>
<td>4.75 (.95)</td>
</tr>
</tbody>
</table>
selection has an impact on APS if a student does not open the available example. Thus, all of the within-subjects statistical analysis is based on the data from the 14 subjects who used examples in both conditions (see Table 2 for a summary of the results). We did perform some analysis that did not involve within-subjects comparisons, for which we considered data from all 16 participants – when this is the case, it is indicated in the text.

Our sample size is comparable to some experiments relying on talk aloud methodology, which complicates the collection and analysis of participant data. For instance, 9 participants were used to study self explanation in the seminal work by Chi et al. (1989); in another study on learning from observing, 10 participants (or pairs of participants) were used per condition (Chi et al., 2008). In contrast to these studies, however, we used a within-subject design that is comparable to a 28 between-subjects experiment, since each subject completes both conditions. In addition, the within-subjects design increases experimental power over a between-subjects design of comparable size because it reduces variability by exposing each participant to all conditions.

We now present our results and corresponding discussion of them, organized according to the EA-Coach example-selection objectives: learning and problem-solving success.

**Results: Learning**

We begin by reporting how example selection influenced APS behaviors that impact learning, namely min-analogy and EBLC, and then provide findings on pre/post test differences.

**Min-Analogy.** To determine example selection’s impact on min-analogy, we analyzed by hand students’ copy events. To do so, we identified: (1) which steps students accessed in the example solution and (2) whether these steps were subsequently copied. To identify which example steps students accessed, we primarily relied on the information provided by the masking interface, which was stored in the EA-Coach log files, since students were not always reliable in verbalizing example solution steps they accessed. To identify if accessed example solution steps were subsequently copied, we checked for correspondences between accessed steps and students’ subsequent problem solution input. Problem entries that were identical to accessed example steps or that shared minor differences of the type listed in the ‘Impact of problem/example differences’ Section were flagged as ‘copied’. In our coding scheme, access to an example solution step corresponded to at most one copy event (i.e., each term in the step was not counted separately). To decide whether a given problem entry was

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th></th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Adaptive</td>
<td>Static</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy Episodes</td>
<td>5.9</td>
<td>8.1</td>
<td>7.978</td>
<td>0.018</td>
</tr>
<tr>
<td>EBLC Episodes</td>
<td>3.14</td>
<td>1.21</td>
<td>14.1</td>
<td>0.004</td>
</tr>
<tr>
<td>Task Time</td>
<td>42min, 23sec</td>
<td>25min, 35sec</td>
<td>31.59</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Errors</td>
<td>22.4</td>
<td>7.6</td>
<td>11.53</td>
<td>0.007</td>
</tr>
</tbody>
</table>
copied, we only considered recently viewed example steps, i.e., the last three example steps accessed before a given problem entry. Although it is conceivable that we missed some copy events because students may have remembered and copied example steps that they accessed earlier, we did not see indications of this in our study. In fact, students who relied on copying from examples tended to do so on a line-by-line basis (i.e., uncover an example step, copy it, uncover the next example step, copy it, etc.), in a fashion similar to that described in (VanLehn, 1998). We did not consider only the last step accessed in the example, for two reasons. First, when students accessed an example step, they sometimes inadvertently uncovered a line or two in the example as they were moving the mouse over to the problem window to copy the step. If we only considered the last accessed step, we would have missed the copy event. Second, after viewing a given example solution step, students would sometimes refer to the example specification or free-body diagram to identify example-specific constants, thereby uncovering additional steps that were related to the copied step.

The ANOVA revealed a significant main effect of example selection on copying (F(1, 10) = 7.978, p=0.018). On average, students copied less from the adaptively than from the statically selected examples (5.9 vs. 8.1; this was the case for 10 of 14 the subjects).

EBLC Events. To identify EBLC, we analyzed the verbal protocols. This coding was done by the first author, who was blind to the condition (i.e., condition information was removed from the transcripts). Recall that EBLC is a form of self-explanation that involves deriving a rule by using overly-general or commonsense reasoning, as opposed to explaining a solution step using an already-known domain rule, referred to as other explanations below.

We used a two-tier approach to analyze example selection impact on self-explanation. First, we identified in the protocols self-explanations, without trying to distinguish between EBLC and other explanations. This coding served two purposes: (1) it provided insight on example selection’s impact regarding how students reason during APS and (2) it enabled us to perform several layers of analysis related to EBLC, as we shall see shortly. To identify self-explanations, we looked for subject comments that contained domain-relevant information over and above what is stated in the instructional materials, based on the definition in (Chi and VanLehn, 1991).

The ANOVA revealed a significant main effect of example selection on self-explanation (F(1, 10) = 12.02, p= 0.006): students expressed significantly more self-explanations in the adaptive condition than in the static condition (on average, 4.0 vs. 2.2).

To identify EBLC-based self-explanations in our pool of explanations, we followed the approach in (VanLehn, 1999) by looking for instances where students generated explanations corresponding to domain principles (1) for which they did not have good domain understanding, suggesting that knowledge gaps existed which required EBLC style reasoning to generate the explanation, or (2) by relying on commonsense and/or overly-general reasoning. The latter is challenging to identify as students’ explanations are often fragmented or incomplete (other attempts to classify EBLC faced similar issues, e.g., (VanLehn, 1999)). The former condition, however, is straightforward to identify given that we have pre-test data.

Specifically, we first identified all student self-explanations that related to rules assessed by the pre-test, referred to as set A explanations below. While the pretest was designed to assess domain principles needed to solve problems in the experimental phase, during the experiment students would sometimes express an explanation that did not have a clear counterpart in the pre-test, referred to as set B explanations below. Of the total 87 explanations under consideration (56 in the adaptive
condition vs. 31 in the static), 75% of the explanations in the adaptive condition related to rules in the pretest (42 explanations total) and 77% of the explanations did so in the static condition (24 explanations total). To identify EBLC in set A, we looked for explanations corresponding to rules that students did not have prior knowledge of as assessed by pre-test questions (i.e., see condition 1 above): this gave us 37 total EBLC explanations in the adaptive condition and 14 EBLC explanations in the static condition (see Figure 10, #1 and #3-4 for sample EBLC explanations). The remaining set A explanations were classified as other, since they related to rules assessed as known by pre-test questions.

We then analyzed the set B explanations (i.e., ones without a clear counterpart in the pre-test). Of these, 7 pertained to algebraic manipulations related to physics content (5 in the adaptive condition and 2 in the static). We labeled these as other explanations, because subjects provided no indication of deriving a new principle when expressing them, and did not rely on commonsense/overly-general reasoning to do so (see Figure 10, #5). This left us with 14 explanations in set B (9 the adaptive condition vs. 5 in the static). To identify EBLC in these, we looked for instances where students either expressed confusion and/or struggled with deriving the rule, suggesting that they did not have prior knowledge, and/or appeared to rely on commonsense or overly-general reasoning (see Figure 10, #2, for an example). There were a total of 10 of such EBLC explanations: (7 in the adaptive vs. 3 in the static). To identify EBLC in these, we looked for instances where students either expressed confusion and/or struggled with deriving the rule, suggesting that they did not have prior knowledge, and/or appeared to rely on commonsense or overly-general reasoning (see Figure 10, #2, for an example). There were a total of 10 of such EBLC explanations: (7 in the adaptive vs. 3 in the static). The remaining set B explanations were labeled as other.

An ANOVA on all EBLC explanations (i.e., from set A and set B) revealed a main effect of selection on EBLC: (F(1,10) =14.2, p=0.004). On average, students generated significantly more EBLC explanations in the adaptive condition than in the static condition (3.14 vs. 1.21; this was the case for 10 of the 14 subjects). In contrast, there was no difference in the number of other explanations between the two conditions.

Pre/Post Test Gains. In general, students significantly improved from pre to post test (2-tailed t(13)=7.49, p<0.001; Table 3 provides information on subjects’ pre and post test performance). Given the within-subject design, however, to analyze how each selection condition influenced pre-to-post learning for a given rule, it is necessary to consider rules that only appeared in one of the two conditions. To gain insight into learning of individual rules, we first identified for each student the set of rules that: (1) were involved in only one of the two conditions and the student used the example in that condition; (2) were not known or partially known as indicated by the pre-test (the test included several questions per rule, so a student could have ‘partial’ knowledge by answering some questions correctly). We refer to the rules satisfying these conditions as ‘learning opportunities’. Over all 16 students, six rules met these criteria. We then identified, for each student and each rule under consideration, the condition in which the student experienced the rule. Over all 16 students who used an example in the given condition, this left us with 11 students who had learning opportunities in the
It’s zero because it is on the y axis (points to the example free body diagram) but this (points to her problem) can’t be because...it’s at an angle”. Expressed while looking at free body diagram in example e5 after reading example solution step “P_x=0”. The subject correctly infers why the x component of the pushing force is zero (i.e., because its component is perpendicular to the x-axis). Classified as EBLC, as pre-test indicated subject did not have pre-existing domain knowledge.

“I think force W will be force N because both are pointing down – will that be right?”. Expressed while solving problem p1. Subject then writes incorrect equation “F_n = F_w”, where F_n and F_w are the variables representing the magnitude of the normal and weight force, respectively, which the subject has drawn correctly in her free-body diagram. Derived principle is not assessed by pre-test. Since both the forces related to the explanation are drawn straight down, the explanation is derived from an overly general rule that the magnitude of two forces is equal if they point in the same direction, and is classified as EBLC.

“It’s negative because it’s under the x axis (referring to the tension vector while looking at free body diagram of example e5) my tension is above the x axis so it should be positive and it will be cos...the angle it makes with it will be 40”. Explanation expressed after reading the example solution step “T_x = -Tcos(5)” in example e4. Classified as EBLC, as pre-test indicated subject did not have pre-existing domain knowledge. Note that subject is relying on an overly general mathematics rule to infer that components for all vectors below the x-axis are negative.

“Py equals zero...why are you zero ... oohh it is because it is zero degrees” First portion expressed after reading example solution step “Py=0” in e3; second portion expressed while looking at the example’s free body diagram, which shows that the pushing force is parallel to the x-axis. The subject’s inference is not clear: a literal translation is that she infers the component is zero because the force direction is zero degrees from the x-axis, which is slanted with respect to the horizontal; an alternative is that she infers that the component is zero because the force is oriented zero degrees to the horizontal. Classified as EBLC, as pre-test indicated subject did not have pre-existing domain knowledge.

“Ok we don’t know acceleration – so I will have to solve for acceleration and use that to solve for net because I have two unknowns”. Expressed after reading problem statement. Classified as other explanation since explanation pertains to algebraic knowledge that does not involve commonsense or overly-general reasoning.

<table>
<thead>
<tr>
<th>Pre-Test</th>
<th>Post Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.6 (8.2)</td>
<td>29.3 (6.8)</td>
</tr>
</tbody>
</table>

Fig. 10. Sample EBLC and other self-explanations.

Table 3

Subjects’ pre and post test performance (N=14)
static condition (2 on average per student), and 11 who had learning opportunities in the adaptive condition (2.18 on average per student).

To identify ‘learning events’ (i.e., rules learned) given a student’s learning opportunities, we analyzed, for each student, the number of rules that the student showed gains on from pre to post-test questions (a gain occurs if, for a given rule, a student performs better on corresponding post-test questions than she did for the pre-test). For a given student, this percentage is calculated by dividing her ‘learning events’ by her ‘learning opportunities’. We found that the percentage of learning events was marginally significantly higher for rules students experienced in the adaptive condition than rules experienced in the static condition (t(20)=1.72, p=0.056, on average, 52% in the static condition vs. 77% in the adaptive condition).

Learning: Summary of Key Results and Discussion

Students copied less and self-explained with EBLC more when presented with adaptively-selected examples, as compared to statically-selected ones. These results show that in general, the EA-Coach example-selection mechanism meets its learning goal, thus supporting our assumption that certain superficial differences encourage effective APS behaviors. The statically selected examples were highly similar to the target problem and so made it possible to generate the problem solution by copying, of which students took advantage. As one subject stated after the study, “1st example (statically selected) was more straightforward ...you don’t have to think that much because it is same thing - step by step”. Conversely, the adaptively selected examples were less superficially similar to the target problem. This provided incentive for students to generate the problem solution without copying, i.e., to engage in min-analogy, when they realized that copying was not an effective strategy. For some students this realization happened after copy attempts resulted in errors, which discouraged subsequent copying. One subject articulated this by saying “I should just ignore this and solve the problem on my own” after several incorrect copy attempts.

To see if there was a difference between the two conditions in how successful students were in copying, we checked by hand each copy event and labeled it as correct or incorrect. On average per student, 66% of copying was correct in the adaptive condition, vs. 86% in the static condition. In the static condition, all of the copy errors (there were 20 of these across all students) corresponded to example solution steps that either had no difference or trivial differences with the problem, and were the result of typos or poor attempts at substituting examples constants with ones needed for the problem’s solution. In contrast, in the adaptive condition, 82% of the copy errors were the result of students copying example solution steps that had non-trivial differences with the problem (there were 23 such errors from a total of 28 copy errors and all but 4 were resolved; the other 5 copy errors were related to example solution steps that had no difference or trivial differences with the problem and all were resolved.) All of the correct copy events in both conditions corresponded to example solution steps that had trivial superficial differences or no differences with the problem.

Thus, for the most part students managed to resolve errors resulting from incorrect copying in both the selection conditions. However, what differentiates the two conditions is how easily the errors could be resolved. In the static condition, all the copy errors could easily be resolved because they corresponded to trivial differences or slips, making it unlikely that they discouraged copying. In contrast, in the adaptive condition the majority of copy errors could not easily be resolved because they corresponded to non-trivial differences, and so could not be reconciled through simple mapping of problem/example specifications and solutions. This, in conjunction with the finding that students
were more successful at copying correctly in the static condition suggests that the type of superficial similarity was responsible for facilitating/discouraging copying in the static/adaptive selection conditions.

In addition to influencing copying, example selection influenced how much students self-explained. According to one of our assumptions embedded in the EA-Coach’s example-selection mechanism, blocking the generation of the problem solution by copying encourages a student to self-explain with EBLC. Since students copied less in the adaptive condition, this may explain the higher number of EBLC explanations in that condition. To see if this was the case, we checked for a correlation between EBLC and copying. For this and subsequent correlation analysis, since we were interested in general about a given relationship, we collapsed across conditions (i.e., used data from both conditions in the analysis). We found that less copying was associated with more EBLC, but this correlation did not reach significance (pearson = -.207, p=.145). Another possibility for why students had more EBLC explanations in the adaptive condition is related to the ‘error’ dependent variable. In particular, errors may have triggered EBLC reasoning to fill the gap revealed by the error, as is proposed by some cognitive theories of learning (VanLehn, 1996). Since as we will discuss below students made significantly more errors in the adaptive condition than in the static condition, the adaptive condition provided more triggers for EBLC to resolve the errors. We found that errors were indeed associated with EBLC (pearson = .468, p=0.006).

Instances when the learning goal was not satisfied. Although in general the adaptively selected examples encouraged the target APS behaviors and thus satisfied the learning goal, this was not always the case. We begin by discussing when this happened with respect to the copy findings. Two students copied more in the adaptive condition than the static condition. One of these students had an above average number of copy events in both conditions (10 vs. 12 copy events in the adaptive and static conditions, compared to the average of 5.9 vs. 8.1, respectively), suggesting that she had a max-analogy tendency. Thus, it appears that more explicit scaffolding than the EA-Coach’s may be needed to encourage a shift over to min-analogy for students with a strong max-analogy tendency. The second student had an average number of copy episodes in the adaptive condition and about half of the average number in the static condition (6 vs. 4 copy events in the adaptive and static conditions, respectively). One explanation as to why this student copied more in the adaptive condition may be related to the order in which this student experienced the conditions. Specifically, the student solved the problem in the adaptive condition first. Although we did not find overall that selection order had an effect, for this student the adaptively selected example may have discouraged copying, but this did not become apparent until the subsequent (static) condition. Two other students had the same number of copy events in both conditions; both number of copy events were below average. This suggests that these students already had a tendency for min-analogy, and so adaptively selected examples did not influence how much they copied.

As far as the EBLC events are concerned, although none of the students expressed fewer explanations in the adaptive condition than in the static condition, four students generated an equal number of explanations in the two conditions. For two of these students, the number of EBLC explanations was similar to the group’s average; it is not clear why the adaptively selected examples did not encourage them to self-explain more. Two students expressed a below-average number of explanations, and so appeared to have a low EBLC tendency. This suggests that additional scaffolding may be beneficial to encourage EBLC for students with a low EBLC tendency.
Results: Problem-Solving Performance

We now report on how example selection influenced students’ success in terms of the final problem solution, and then provide findings on selection’s impact on performance during the problem-solving process.

Problem-solving success. Problem-solving success is achieved if students generate a correct problem solution. For the 14 subjects considered in the within-subjects analysis, problem-solving success was realized for (1) all subjects in the static condition; (2) 12 subjects in the adaptive condition (the two students who used an example in only one condition generated a correct problem solution in both conditions). This difference between the two conditions is not statistically significant (sign test, \( p=0.5 \)), indicating that overall, both statically and adaptively selected examples helped students generate the problem solution. We should point out that feedback for correctness delivered by the EA-Coach was likely a factor in helping students achieve a fully correct solution.

Problem-solving process. To analyze how example selection affected the problem-solving process, we analyzed students’ task time and errors in each condition. The ANOVA revealed a significant main effect of example selection on task time (\( F(1, 10) = 31.59, p<0.001 \)): students took significantly longer to generate the problem solution in the adaptive condition than in the static condition (on average, 42min, 23sec vs. 25min, 35sec; this was true for 13 of 14 subjects). There was also a significant interaction between example selection and selection order, indicating that condition presentation order disproportionately affected task time (\( F(1,10)=8.09, p=0.017 \)). The magnitude of the difference in task time between the two conditions was greater if the adaptive condition was first in the selection order than if it followed the static condition (see Figure 11a).

For the error analysis, we counted all errors a student made while generating a problem solution in each condition. The ANOVA revealed a significant main effect of example selection on error (\( F(1, 10)=11.53, p=0.007 \)): students produced significantly more errors while generating the problem solution in the adaptive condition than in the static condition (on average, 22.35 vs. 7.57; this was true for 11 of the 14 subjects). There was also a significant interaction between selection and selection order, indicating that condition presentation order disproportionally affected error (\( F(1,10)=6.89, p=0.025 \)). As was the case for the time dependent variable, the magnitude of the difference in errors between the two conditions was greater if the adaptive condition was first in the selection order than if it followed the static condition (see Figure 11b).

Problem-Solving Performance: Summary of Key Results and Discussion

As stated above, problem-solving success is achieved if the student generates the problem solution, and is not a function of performance (time, errors) while doing so. In general, students were successful in generating the problem solution in both the static and adaptive selection conditions.

Instances when the problem-solving success goal was not satisfied. Two students generated a correct but incomplete solution in the adaptive condition. Both students received an example with non-trivial superficial differences that blocked copying of some of its solution steps, because the EA-Coach user model predicted that this would trigger learning via EBLC. This prediction is mediated by the model’s assessment of the student’s EBLC tendency, to which we had assigned a generic prior probability of 0.5 for both students due to lack of more accurate information. This appeared to have been inaccurate for one of these students, who showed no desire to engage in any in-depth reasoning during the study.
and so likely had a very low EBLC tendency. The other student, however, generated a number of EBLC self-explanations, indicating that inaccurate prior on EBLC tendency was not the reason for suboptimal example selection in terms of problem-solving success. This student invested considerable effort and did learn some of the rules needed to solve the problem (as we found by comparing her pre and post-test answers on related questions). However, although the user model simulation predicted she would learn all the necessary rules and thus generate the full problem solution, she was unable to do so within the allotted 60 minutes. We can’t predict whether this student would have eventually generated a full solution if more time was available or whether she would have become overwhelmed and frustrated by the process. There is a fine line between taking extra time to learn from one’s errors, and entering floundering, i.e., engaging in too many trial and error attempts that obstruct learning. This suggests that the system could be improved adding more explicit scaffolding to help students learn rules via EBLC when they are floundering.

Task time & errors. Students took longer/made more errors in the adaptive condition than in the static condition. This was particularly the case in the adaptive-static selection order (i.e., task time and number of errors were higher if the adaptive condition preceded the static condition). The most plausible explanation for this finding is that first solving the problem in the static condition provided a scaffold that better prepared students to solve the problem in the adaptive condition. As one of the subjects in the static-adaptive order indicated, “I liked the transition between the two… one was very similar, exactly the same (i.e., statically selected example) – this one (i.e., adaptively selected example) you had to manipulate the set up… It’s not hard to manipulate but you’d have to think about the question and how to relate them - the first one showed you the basics”.

In general, however, the fact that students had a higher task time and more errors in the adaptive condition is not a negative result from a pedagogical standpoint, because these are by-products of learning. Research shows that learning takes time and may require multiple attempts before the relevant pieces of knowledge are inferred/correctly applied (e.g., (VanLehn, 1991; Chi, 2000)). In particular, it is reasonable to assume that not copying and trying to reason via EBLC can take longer and result in more errors before a solution is found. To gain more insight into the relationships between APS behaviors and problem-solving performance in our study, we checked for correlations between the corresponding variables (time/errors and copying/EBLC). As far as errors are concerned, we already reported above that errors were associated with EBLC. We also found a trend that generating a problem solution without copying is associated with more errors during the problem-
solving process (pearson =-.271, p=0.082). This analysis suggests why students had more errors in the adaptive condition than in the static condition: because they generated more of the problem solution on their own, without copying. As far as task time is concerned, the correlation analysis follows our conjecture stated above that (1) copying is associated with decreased task time (pearson=-.362, p=0.028); (2) EBLC is associated with increased task time (pearson=.508, p=0.003).

RELATED WORK IN INTELLIGENT TUTORING SYSTEMS

We now review a representative sample of related work from the ITSs community, starting with ITSs that support APS. Since cognitive science research shows that students have difficulty choosing appropriate examples, such tutors all perform example selection for students. CATO (Aleven & Ashley, 1997) helps students build legal arguments by dynamically generating examples of how an expert would argue about a particular legal case. SPIEL (Burke & Kass, 1995) helps students acquire social skills needed for effective salesmanship by presenting ‘stories’ corresponding to video clips. To recognize when to present a story, SPIEL relies on a set of rules encoding general story-telling strategies. AMBRE (Guin-Duclosson, Jean-Daubias et al., 2002; Nogry, Jean-Daubias et al., 2004) helps students solve algebra word problems, by selecting examples that are “nearest to the target problem” (no details are provided on the selection mechanism). ELM (Weber, 1996a; Weber, 1996b) supports LISP programming by selecting example for them. Like Ambre, ELM’s selection criteria is to find “the most similar example to the target problem” (Weber, 1996b), based on a comparison of problem/example solutions.

Of the ITSs described above, ELM and AMBRE include problem-solving domains that have a clear notion of correct and incorrect behavior, and so are most similar to the EA-Coach’s. However, none of the above-described tutors reason about how an example will impact a student’s learning/problem-solving outcomes from APS, or take the approach of assuming some problem/example differences are beneficial to learning while enabling problem-solving success.

In recent years, there has been increasing interest in developing ITSs that target general meta-cognitive skills (e.g., (Aleven & Koedinger, 2002; Aleven, McLaren et al., 2006; Baker, Corbett et al., 2006a; Roll, Aleven et al., 2006; Conati & VanLehn, 2000; Conati, Merten et al., 2005)). For instance, the Help-Seeking (HS) Tutor provides support for the meta-cognitive skill of help seeking (Aleven, McLaren et al., 2006). This tutor relies on a production rule model to detect ineffective help-seeking behaviors and generates prompts to correct them. One form of help abuse relates to “gaming”, i.e., attempting to succeed in an educational environment by exploiting properties of the system rather than by learning the material and using that knowledge to generate the solution (Baker, Corbett et al., 2006a). Since gaming can interfere with learning, there is interest detecting it (Baker, Corbett et al., 2006b) and preventing it (Roll, Aleven et al., 2006). All of this work involves ITSs that support pure problem solving by providing detailed hints instead of examples.

While effective help-seeking is related to min-analogy, since both skills involve not abusing available help, the manner in which students can abuse help in these tutors involves asking for detailed hints too frequently, instead of indiscriminately copying from examples. To date, however, existing tutors have not targeted meta-cognitive skills during APS, and thus none have targeted min-analogy (which is specific to APS). In contrast, self-explanation has been supported in a variety of instructional activities (e.g., (Conati & VanLehn, 2000; Aleven & Koedinger 2002; Mitrovic, 2003)). Instead of
selecting examples, however, the standard approach for fostering self-explanation has been to provide students with (1) tools that they can use to derive the explanations, including feedback for correctness on the self-explanations, and/or (2) prompts encouraging self-explanation, during a variety of instructional activities. For instance, the SE-Coach (Conati & VanLehn, 2000) fosters self-explanation during example studying with both types of support mentioned above (tools, prompts). Other tutors support self-explanation during pure problem solving without giving students access to examples. The Geometry Explanation Tutor (Alevrin & Koedinger, 2002) targets self-explanation during geometry theorem proving, while Normit-SE’s (Mitrovic, 2003) support is provided during database normalization. Both tutors provide interface tools that students use to generate self explanations on their problem-solving entries. There is also work on supporting self-explanation during exploration of open learning environments (Bunt et al., 2004).

Other tutors support pure problem solving without targeting meta-cognitive skills (e.g., (Conati, Gertner et al., 2002; Anderson, Conrad et al., 1989)). The most relevant tutor to mention is Andes (Conati, Gertner et al., 2002; VanLehn, Lynch et al., 2005), which provides students with Newtonian physics problems to solve and hints on how to do so, but does not provide access to examples. The EA-Coach bases its problem-solving interface on the Andes design, and also follows the Andes approach to construct its user model. However, the EA-Coach user model operates in a different instructional situation than the Andes model and so distinguishes itself in three key ways: (1) while the Andes Bayesian network is static, the EA-Coach uses a dynamic Bayesian network, needed to model the evolution of students’ knowledge from EBLC explanations; (2) the EA-Coach model includes additional nodes to infer the impact of examples on problem solving and (3) only the EA-Coach model generates a simulation of how a student will solve the target problem given a particular example.

In contrast to the ITSs described thus far, other tutors rely on a decision-theoretic approach for action selection. The INQPRO ITS (Ting, Zadeh et al., 2006) generates interventions aimed at helping students acquire scientific inquiry skills. To decide which intervention to generate, INQPRO uses a Bayesian network supplemented with utility and decision nodes, i.e., a decision network. The CAPIT (Mayo & Mitrovic, 2001) tutor help students acquire punctuation skills. CAPIT combines a Bayesian network’s prediction with a utility function to realize two types of support: (1) selection of problems for students to work on and (2) generation of hints. Instead of selecting problems for students, the Decision Theoretic (DT)-Tutor (Murray, VanLehn et al., 2004) selects tutorial actions (e.g., provide hint, provide affective feedback). DT-Tutor includes a dynamic decision network, used to calculate a candidate tutorial action’s expected utility in terms of its impact on a student’s knowledge, problem-solving progress and affective state. While all these tutors rely on a decision-theoretic approach, as does the EA-Coach, they do not (1) formalize the utility of an example in terms of its ability to foster learning and problem-solving success APS outcomes or (2) include a probabilistic user model for inferring how APS-specific factors, such as problem/example similarity, impact these outcomes.

**CONCLUSIONS & FUTURE WORK**

In this paper, we have described the EA-Coach, an ITS that encourages meta-cognitive behaviors needed for effective APS through tailored example selection. The underlying motivation for our work is that although examples play a key role in cognitive skill acquisition, some students fail to learn from them effectively. Thus, it seems highly valuable to provide computer-based support for meta-cognitive
skills relevant to APS, to complement the availability of support targeting meta-cognitive skills during other instructional activities.

Currently, the main form of support provided by the EA-Coach is its example selection mechanism. Recall that this mechanism takes into account information related to the following three factors: (1) problem/example similarity, (2) student knowledge and (3) student meta-cognitive tendencies. While the evaluation of the EA-Coach provided support for its pedagogical utility in general, thus far we have not teased apart the impact of individual factors on outcomes of interest. For instance, one such analysis could focus on the second factor (student knowledge). The selection mechanism aims to choose examples that include non-trivial differences corresponding to students’ knowledge gaps, in order to encourage students to fill the gaps with EBLC. In the evaluation, this is what occurred (i.e., the mechanism tailored the selection according to students’ knowledge gaps, which we identified via the pre-test – thus, the same example was not always selected for a given problem). In the future, it would be interesting to evaluate the impact of this adaptation strategy on outcomes of interest, as compared to, for instance, selecting an example with a fixed set of non-trivial differences that are not tailored to a student’s domain knowledge. We conjecture that tailoring to student knowledge is beneficial over choosing an example that maximizes non-trivial differences but does not match the differences to a student’s knowledge gaps, because only the former strategy encourages impasses, and impasses are known to trigger learning (e.g., (VanLehn, Siler et al., 2003)). This conjecture could be tested by adding another condition that maximizes non-trivial differences without tailoring selection to student knowledge; in general, ablation studies could be used to test the impact of individual system components.

It is important to note that even if the system failed to have any student-related information and only took into account problem/example similarity, choosing suitable examples still requires sophisticated techniques to enable real-time selection and is preferable to statically matching examples with problems by hand a priori, for a number of reasons. First, once the selection algorithm is tested, it is more reliable and/or consistent than human hand matching. Furthermore, automatic selection relieves the burden of having to hand match examples to problems, which can be considerable given a large example pool. Related to the last point, once a selection algorithm is designed and implemented, there is no cost in applying it to new problems as they are added to the problem pool, something hand-coding does not afford. Automatic techniques also facilitate modification of the example-selection algorithm, since once the modification is implemented, selection is automated (as opposed to hand-coding, which requires re-assigning the entire set of problem/example pairs). In fact, existing systems that perform example selection only take into account problem/example similarity, but still rely on automated selection strategies for these very reasons (Weber, 1996b).

We conclude with a summary of our contributions and a discussion of future work. Our work has three key contributions. First, we bring a contribution to Intelligent Tutoring Systems: the EA-Coach is the first ITS that supports meta-cognitive skills during APS and does so by adaptively selecting examples with various levels of similarity to the target problem. Second, we bring a contribution to User Modeling: the EA-Coach user model relies on our definition of similarity to infer how it, in conjunction with student characteristics, influences APS outcomes for a given student. Third, the EA-Coach evaluation is the first to explore the utility of providing examples that include adaptively-selected problem/example differences instead of providing maximally-similar examples. Although preliminary, we see our results as an important first step toward understanding the influence of
problem/example similarity on APS behaviors and subsequent learning and problem-solving outcomes.

More empirical studies will be necessary to replicate and generalize our findings, as well as obtain insights on other yet to be tested EA-Coach functionalities. For instance, in our evaluation, the meta-cognitive tendency nodes were fixed. As we mentioned above, the main reason for doing so was to avoid adding a third condition to assess meta-cognitive tendencies, since it would mean our within-subjects experiment would no longer be counter balanced. A challenge for our future work, therefore, is how we can obtain such information, so as to evaluate the impact of tailoring selection to meta-cognitive tendencies. One possibility is to devise our own tendency assessment instruments, since we are not aware of any existing tests of self-explanations and min-analogy tendencies. Another is allow the EA-Coach student model to perform the assessment of tendency, although this suffers from the disadvantage of not separating the evaluation of the system’s selection mechanism from its model’s assessment accuracy.

To date, we have focused on evaluating the EA-Coach example-selection mechanism, since it is the key form of support delivered by the EA-Coach. A future evaluation could investigate the system’s scaffolding embedded in its interface – for instance, does the masking interface discourage copying as it is intended to? Instead of a laboratory experiment, another avenue for a future evaluation we are interested in is a longitudinal field study, where students would use the EA-Coach for a longer period of time than is afforded by a laboratory experiment, for instance while enrolled in a physics course. A longitudinal study has the advantages of affording greater ecological validity than a laboratory experiment, and makes it possible to evaluate the EA-Coach’s long-term impact on students.

Our evaluation showed that some students need more explicit scaffolding for meta-cognition than what is currently provided by the system, and so we are working on designing additional support. One form could correspond to tools to help students infer the appropriate domain principles via EBLC. To date, ITSs that provide tools for self-explanation have supported only domain-based reasoning, without the ability to support the commonsense and overly-general reasoning that characterizes EBLC. Thus, it is an open question as to how EBLC tools should be designed, or how to incorporate the complex domain and user models needed to allow the system to capture and provide feedback on EBLC. In addition to tools, another form of scaffolding could correspond to system-generated meta-cognitive interventions for min-analogy and EBLC, which our evaluation suggested could be beneficial for some students. A common approach for realizing meta-cognitive interventions involves generating prompts, e.g., to encourage self-explanation (Conati & VanLehn, 2000). Another approach involves using animated pedagogical agents to express approval or lack of it, based on whether students are gaming the system in ways that interfere with learning (Baker, Corbett et al., 2006a).

Another extension to the EA-Coach we are interested in corresponds to the incorporation of eye-tracking technology. Currently, the tutor’s ability to track students’ example usage is due to the masking interface. The interface, however, provides less information than an eye tracker, both in terms of granularity of attention patterns and physiological data (e.g., an eye tracker could provide data on pupil dilation, which is related to cognitive load and affect (Marshall, 2007)). We have taken preliminary steps in this direction by showing that pupillary response reliably distinguishes different types of reasoning, such as self-explanation and shallow reasoning, and different types of affect (Muldner et al., 2009), as well as created a user model to recognize moments of excitement during interaction with the EA-Coach, based on information provided by an eye-tracker and other physiological sensors (Muldner et al., 2010). A challenge relates to how this type of information
should be integrated into the EA-Coach model’s assessment and how useful it would be in terms of improving the model’s accuracy.

ACKNOWLEDGEMENTS

We thank Andrea Bunt, Benjamin Phillips and the anonymous reviewers for their helpful comments and suggestions on earlier drafts of this paper.

REFERENCES


Appendix: Problems and Examples used during the Evaluation of the EA-Coach

Problems (P1-P2) and Examples (E1-E6): during the evaluation, in the static condition, problem P1 was paired with example E1 and problem P2 was paired with example E4 in the static condition (note that the problems were counterbalanced, so that some students experienced P1 in the static condition, while others saw P2).

<table>
<thead>
<tr>
<th>Problem/Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P1)</td>
<td>Barb has decided to attach a picture to her ceiling. To see what it would look like, she holds it against the ceiling. She is applying a pushing force of 190N at an angle of 21 degrees to the horizontal. The picture has a mass of 3 kg and is moving along the ceiling with some unknown acceleration. What is the magnitude of the normal force on the picture?</td>
</tr>
<tr>
<td>(P2)</td>
<td>A toy mouse of some unknown mass hangs from the ceiling on a string. A playful cat pushes the mouse with a force of 50N, at an angle of 30 degrees to the horizontal, as shown. At this point, the mouse is not moving and the string makes an angle of 40 degrees with the horizontal. What is the tension in the string?</td>
</tr>
<tr>
<td>(E1)</td>
<td>Jake is trying to install a light fixture of mass = 2 kg. He is pushing it into the ceiling with a magnitude of 180 N. He is applying his force at an angle of 33 degrees to the horizontal and the fixture is moving along the ceiling, with some unknown acceleration. What is the magnitude of the normal force on the fixture?</td>
</tr>
<tr>
<td>(E2)</td>
<td>Greg is pushing a crate of mass 40 kg along the floor. He pushes with a magnitude of 65 N. This force is directed downwards and applied at an angle of 30 degrees from the horizontal. The crate is moving with some unknown acceleration. Find the magnitude of the normal force on the crate.</td>
</tr>
<tr>
<td>(E3)</td>
<td>A child is pushing a sled of mass 4 kg up a hill inclined at 10 degrees to the horizontal. This pushing force is applied parallel to the horizontal (i.e., at 0 degrees), but we don't know its magnitude. The crate is moving with an acceleration of 1.2 m/s^2. Find the magnitude of the normal force on the sled.</td>
</tr>
<tr>
<td>(E4)</td>
<td>A pumpkin of some unknown mass is suspended by a cord and pushed by Ann with a 45 N force until the cord forms an angle of 32 degrees with the horizontal (as shown). At this point, the pumpkin is at rest. The pushing force is applied at an angle of 18 degrees with the horizontal. What is the tension in the cord?</td>
</tr>
<tr>
<td>(E5)</td>
<td>A yoyo (fully unwound) of mass 2 kg is hanging from a string. Jane is pushing the yoyo so that its string makes an angle of 25 degrees with the horizontal – at this point, the yoyo is at rest. Jane applies the pushing force at an angle of 60 degrees to the horizontal, as shown. We don't know the magnitude of the pushing force. What is the tension in the string?</td>
</tr>
<tr>
<td>(E6)</td>
<td>A block is attached to a string, which is attached to the ground. Bob pulls on the block with a force that has a magnitude of 40 N, applied at an angle of 35 degrees to the horizontal. At this point, the block is at rest. The angle the string makes with the horizontal is 25 degrees. We don't know the mass of the block... What is the tension in the string?</td>
</tr>
</tbody>
</table>