Using Similarity to Infer Meta-Cognitive Behaviors During Analogical Problem Solving

Kasia Muldner and Cristina Conati

Department of Computer Science, University of British Columbia,
2366 Main Mall, Vancouver, BC, V6T1Z4, Canada
{muldner, conati}@cs.ubc.ca

Abstract. We present a computational framework designed to provide adaptive support aimed at triggering learning from problem-solving activities in the presence of worked-out examples. The key to the framework's ability to provide this support is a user model that exploits a novel classification of similarity to infer the impact of a particular example on a given student's meta-cognitive behaviors and subsequent learning.

1 Introduction

Research demonstrates that students rely heavily on examples, especially in the early phases of learning (e.g., [3, 4, 14, 15]). Therefore, there is a substantial amount of work in the cognitive science and Intelligent Tutoring Systems (ITS) communities exploring how examples impact learning, and how computer-based adaptive support can be provided so that examples are used effectively (e.g., [1, 8, 18]). This support typically takes one of two forms. One form involves selecting examples for students during problem-solving activities (e.g. [1, 18]). A second form involves providing guidance on skills needed to learn from examples effectively, based on evidence that different learners have various degrees of proficiency for using examples (e.g., [4]). For instance, the SE-Coach provides support for the meta-cognitive skill of *self-explanation* (the process of explaining instructional material to one self) during example studying, *before* students start solving problems [8]. Here, we describe the E-A (Example-Analogy) Coach, a computational framework designed to provide adaptive support for meta-cognitive skills required for effective *analogical problem solving* (APS), i.e., using examples *during* problem-solving activities.

A key factor that must be taken into account when providing support for APS is the similarity between the problem and example, since there is evidence that this similarity impacts the problem-solving process. For instance, students have difficulty using examples that are not similar enough to the target problem (e.g., [10, 12]). Thus, systems that select examples for students typically aim to minimize the differences between a problem and the chosen example [18]. Although this approach has been shown to be effective, we believe that student characteristics should also play a role in a system's analysis of the example's impact on APS. For instance, problem / example differences which the student has the knowledge to reconcile do not have the same impact on learning as differences which correspond to a student's knowledge gaps. In addition, there is evidence that even given very similar examples, students do not necessarily learn well, possibly because they engage in excessive copying that interferes with learning [15, 17]. Although it is clear that problem / example similarity affects APS, there is not much understanding on *how* this happens for different types of learners (personal communication, M. T. Chi). Here, we propose that certain kinds of similarity can have a positive impact on students who lack meta-cognitive skills needed for effective APS. We incorporate this assumption into the E-A framework, and thus extend existing work on supporting APS by: 1) proposing a novel classification of similarity, and 2) devising a user model that relies on this classification, as well as student knowledge and meta-cognitive skills, to assess the impact of various examples on APS. This assessment is used by the framework to provide tailored interventions (including example selection) to improve this process.

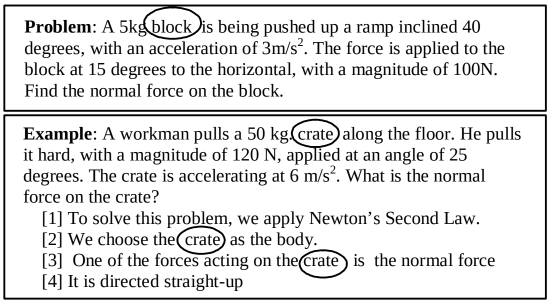
In the rest of the paper, we first discuss the skills needed for APS. We then describe the overall E-A framework. Finally, we present the E-A user model, and discuss how it can be used to generate adaptive support for effective APS.

2 Skills Needed for Analogical Problem Solving

Analogical problem solving consists of the example *retrieval* and *transfer* phases. The *retrieval* phase involves the selection of an example to help solve the target problem. This phase is governed by expertise, in that novice students tend to experience difficulties finding examples that both facilitate problem solving and support learning (e.g., [10]). The *transfer* phase involves incorporating information from an example into a problem's solution [2, 15, 16]. The learning outcomes from this phase are influenced by meta-cognitive skills which can be categorized along two dimensions: *analogy-type* and *reasoning*.

The *analogy-type* dimension characterizes a student's preferred style of problem solving when examples are available (e.g., [15, 17]). Min-analogy identifies students who try to solve a problem on their own, and refer to an example only when they reach an impasse. Max-analogy identifies students who copy as much as possible, regardless of whether they have the knowledge to solve the problem on their own. Students who prefer min-analogy tend to learn more, because they have opportunities to 1) strengthen their knowledge through practice, and 2) uncover knowledge gaps.

The *reasoning* dimension is characterized by how a student tries to understand the example solution prior to using it to solve the problem. A behavior that is believed to result in good learning is *explanation based learning of correctness* (EBLC), a form of self-explanation used to overcome impasses when existing domain knowledge is insufficient to understand the example solution [6, 16]. This process involves using common-sense knowledge (instead of domain knowledge), in conjunction with general rules, to derive new rules that can justify an unclear step in the example. For instance, Fig. 1 shows a problem and example in the domain of Newtonian physics, while Fig. 2 (top) shows how EBLC can be used to explain the existence of the normal force mentioned in line 3 of the example in Fig. 1 [16]. This reasoning can be compressed into a rule (Fig. 2, bottom) that the student can then use to solve the

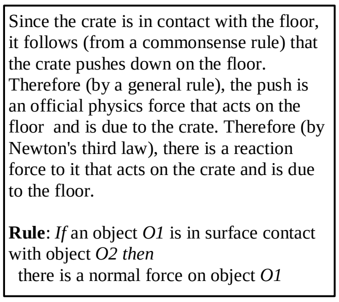


Problem: A 5kg block is being pushed up a ramp inclined 40 degrees, with an acceleration of 3m/s^2. The force is applied to the block at 15 degrees to the horizontal, with a magnitude of 100N. Find the normal force on the block.

Example: A workman pulls a 50 kg crate along the floor. He pulls it hard, with a magnitude of 120 N, applied at an angle of 25 degrees. The crate is accelerating at 6 m/s^2. What is the normal force on the crate?

[1] To solve this problem, we apply Newton's Second Law.
[2] We choose the crate as the body.
[3] One of the forces acting on the crate is the normal force
[4] It is directed straight-up

Fig. 1: Sample Problem & Example



Since the crate is in contact with the floor, it follows (from a commonsense rule) that the crate pushes down on the floor. Therefore (by a general rule), the push is an official physics force that acts on the floor and is due to the crate. Therefore (by Newton's third law), there is a reaction force to it that acts on the crate and is due to the floor.

Rule: *If* an object *O1* is in surface contact with object *O2 then*
there is a normal force on object *O1*

Fig. 2: Reasoning via EBLC

problem in Fig. 1, top. There some indication that certain students have an inherent tendency for this type of reasoning [6]. Unfortunately, many students employ more shallow processes when using examples during APS, which either do not result in learning, or result in shallow forms of knowledge (e.g., [13, 15, 16]). For instance, rather than reasoning via EBLC, students could focus on adapting example line 3 (Fig. 1) to copy it over to the problem. This can be done by substituting example constants by problem ones (i.e. *crate* by *block*) to generate the correct answer in the problem (this process is known as *transformational analogy* or *mapping-application* [2, 15]). Although this reasoning does accomplish the adaptation needed for correct transfer, it doesn't lead to learning the appropriate rule. Given that learners have various degrees of proficiency for using examples (e.g., [4, 6, 15]), the overall goal of our work is to provide a framework that encourages min-analogy and EBLC and discourages its ineffective counterparts. We begin by describing the E-A architecture.

3 The E-A Architecture

The overall architecture of the E-A Coach is shown in Fig. 3. The system contains two data bases of problems: worked-out examples and problems for students to solve. The solutions to these are automatically generated by the problem *solver*, using the *problem specification* and the rules found in the *knowledge base* component. The E-A *interface* allows students to interactively solve problems from the *problem pool* and to

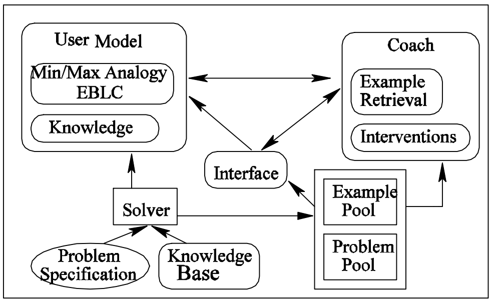


Fig. 3: The E-A Architecture

refer to worked-out examples in the *example pool*. The E-A *coach* relies on the *user model's* assessment of a given student's knowledge and APS behaviors to provide adaptive support for APS. This support includes suggesting appropriate examples and generating hints to encourage EBLC and min-analogy when needed.

Our approach for providing tailored support for APS is domain independent and applies to any problem-solving domain for which a rule-based representation is applicable. However, the current instantiation of the E-A framework is embedded in Andes, a tutoring system for Newtonian Physics [7]. Andes provides support for problem solving and example studying in isolation. The E-A Coach is designed to provide a bridge between these two modes, allowing for a smooth transition from pure example studying to independent problem solving. We now describe the E-A user model.

4 The E-A User Model

The E-A user model allows the framework's coach to provide individualized support to students during APS, by operating in two modes: *assessment* and *simulation*. In *assessment* mode, the model generates an appraisal of how well a student is using an example to solve a given problem, based on that student's interface actions. This allows the E-A coach to generate interventions encouraging EBLC and min-analogy only when it becomes apparent that the student is not learning effectively. In *simulation* mode, the model generates a prediction of student behavior and consequent learning for a particular problem / example pair. This allows the framework to find an example in its example pool that maximizes learning for a particular student. To perform both assessment and simulation, the model reasons about the student's cognitive skills (knowledge) and meta-cognitive traits (analogy and EBLC tendencies). To do so accurately, it takes into account the impact of problem / example similarity on these student characteristics, as we discuss below.

4.1 Impact of Similarity on the E-A User Model's Assessment

To show how the E-A user model incorporates similarity into its assessment, we need to first describe how similarity impacts APS. The similarity between a problem and example is typically classified as either *superficial* or *structural* (e.g., [5, 10]). *Superficial* similarity is assessed using features not part of the underlying domain knowledge, such as the objects in the problem specification and/or its solution (e.g., *block* and *crate* in the problem and example in Fig. 1). *Structural* similarity is assessed using the domain principles (rules) needed to generate the solution (e.g., the rule derived via EBLC, Fig. 2).

Let's now look at how these two kinds of similarity impact problem solving, and in particular, how they can be used to encourage effective APS. One of the downfalls of using examples is that some students copy from them without trying to learn the principles that generated the example solution. This could be prevented by introducing structural differences into the example. However, the benefit of doing so strongly depends on whether the student knows the rules involved in these differences. If the student knows the rules, the lack of similarity with the example forces her to do pure problem solving, which can be highly beneficial. If, however, the student does not know these rules, the example will not be helpful for acquiring them to generate the problem solution, and no learning gains will occur.

On the other hand, superficial differences do not prevent students from learning the underlying concepts, which increases the chances that they can carry on in their problem-solving. However, as we already discussed, some students do not reason effectively from superficially-similar examples. Although there is some evidence that superficial similarity impacts example retrieval and classification [5, 11], it is still not clear how different levels of superficial similarity influence students' meta-cognitive behaviors necessary for effective reasoning. In the process of investigating this issue, we realized that we needed a finer-grained classification of superficial similarity than one currently available in the literature. Thus, we developed one, based on further categorizing superficial differences as:

- *trivial*: differences between problem / example solution elements which correspond to constants that appear in *both* the example specification and its solution, and have a corresponding constant in the problem specification. In addition, in order for a difference to be classified as trivial, simple substitution of the example constant by the corresponding problem constant is sufficient (i.e. requires no further inference) to generate a correct solution step in the problem. For instance, a *trivial* difference between the problem and example in Fig. 1 corresponds to the objects chosen to be the body in their solutions: *crate* (line 2, example solution) and *block* (problem solution, not shown);
- *non-trivial*: differences between problem / example solution elements corresponding to constants that do not appear in both problem / example specifications, or that require additional inference to generate a correct problem solution. One such *non-trivial* difference in Fig. 1 relates to the problem solution step requiring that a normal force be drawn perpendicular to the ramp, as opposed to straight up for the example (line 4, example). This difference depends on a constant defining the incline of the surfaces on which the block and crate rest, which only appears in the problem specification, but which also requires additional inference in order to be reconciled (i.e. that the force is directed 90 degrees away from the surface's incline).

Note that the classification is based on comparing solutions, which the E-A Coach has access to (students only have access to the example solution). Given this classification, we have two hypotheses regarding the impact of superficial similarity on APS behaviors, which are based on cognitive theories of learning from examples [2, 15, 16]. First, trivial differences do not stimulate EBLC and min-analogy for students who do not spontaneously engage in these processes and have poor knowledge. There is some evidence backing up this assumption: students do not have difficulty reconciling trivial differences during APS to generate the problem solution, but do not always learn from doing so [13, 15]. Second, non-trivial differences can have a positive impact on learning for students with poor APS skills. This assumption is based on our observation that only the 'good' APS processes (i.e. EBLC and min-analogy) make it possible to resolve the non-trivial difference and generate a correct problem solution. For instance, students can not apply transformational analogy to

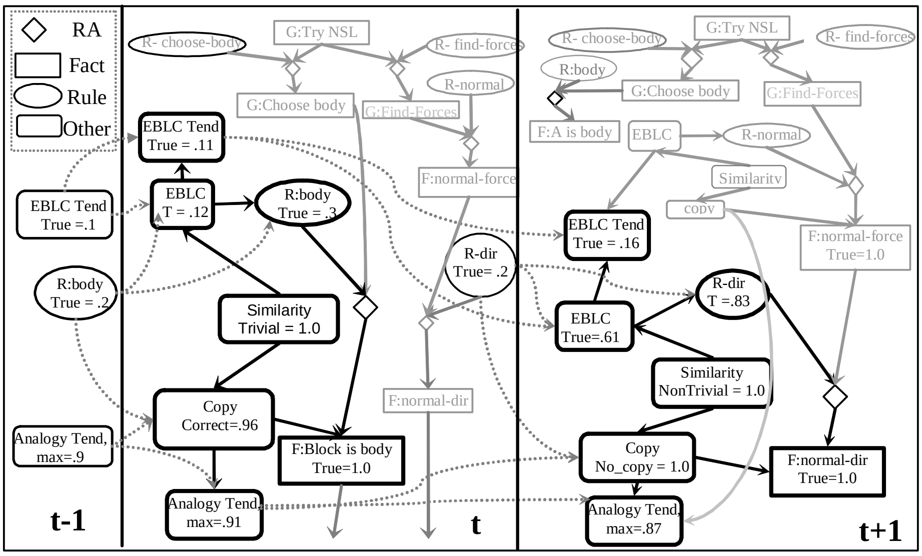


Fig. 4: Fragment of the E-A User Model (not all probabilities/links between slices are shown)

transfer example solution line 4 in Fig. 1, and still obtain the correct solution. To correctly continue problem solving, they need to generate the rule via EBLC.

Our proposed classification allows the E-A user model to reason about the impact of superficial similarity on students' APS behaviors. This extends existing approaches to providing such support, since these do not make an explicit distinction about different *kinds* of superficial similarity, or their impact on students' meta-cognitive behaviors [1, 18]. We will now discuss in more detail how the classification and related hypotheses are incorporated into the E-A user model.

4.2 Assessment Mode

We first describe how the E-A user model operates in assessment mode to evaluate a student's APS behavior. The model relies on the Andes approach, which involves: 1) automatically generating a solution graph consisting of all the steps and corresponding knowledge needed to solve the problem, as well as paths between these steps, and 2) automatically converting this solution graph into a Bayesian network, used in real-time to perform assessment [7]. The Andes model, however, does not account for the presence of examples during problem-solving activities. Thus, we have extended this model, as is described below.

The Andes model assesses student knowledge based on problem-solving actions, but does not assess how knowledge can *evolve* through these actions. Since the E-A model does need to assess learning resulting from EBLC reasoning, we have switched to using a fully dynamic Bayesian network. In this network, each problem-solving action results in the addition of a new slice. Fig. 4 shows a small portion of this network, assuming that a student 1) is solving the problem and has access to the example in Fig. 1, and 2) has generated two problem-solving actions (key nodes of interest are shown in bold). First, we describe the semantics of each type of node (unless otherwise stated, all nodes have *True/False* values):

- *Fact*: facts and goals (F & *G* prefixes in Fig. 4) corresponding to solution steps
- *Rule*: whether the student knows the corresponding rule
- *RA*: whether the student can generate the corresponding fact, either by copying or by reasoning
- *Copy*: whether the student copied a step. This node has three values: *Correct, Incorrect, NoCopy*
- *Similarity*: the similarity between a problem fact node and the corresponding fact in the example solution. This node has three values: *Trivial* (no difference or trivial superficial difference), *NonTrivial* and *None* (i.e. structural difference)
- *Analogy Tend*: whether the student has a tendency for *min* or *max* analogy
- *Eblc*: whether the student has explained the step through EBLC
- *EBLC Tend:* a student's EBLC tendency.

Each slice in the network contains the solution graph and the two tendency nodes (analogy and EBLC). Each student problem-solving action is entered as evidence into the network, and results in the addition of corresponding copy, similarity and EBLC nodes. For instance, in slice *t*, the student chose the block as the body to solve for in the problem in Fig. 1 (*'F:Block is body'* node). In slice *t+1*, the student drew a normal force (*'F:normal dir', 'F:normal-force'* nodes*)*. We now describe how the model performs its various types of assessment during APS.

Assessment of Copy Episodes. The model tries to assess whether a student copied a step to evaluate: 1) the evolution of student knowledge, since self-generated entries provide a stronger indication of knowledge than copied ones, 2) student analogy tendency. If direct evidence of copying is available, the model uses it to observe the copy node to the appropriate value. In the absence of direct evidence of copying, the model uses information about student knowledge, and analogy tendency from the previous time slice, as well as problem / example similarity, to assess the probability that a copy took place in the current slice. Fig. 4 demonstrates this process. In slice *t*, where the student produced an entry corresponding to the '*F:Block is body'* node, the model's belief that this step was copied is high (*'Copy'*, *Correct* =.96), due to the similarity with the corresponding example step (*'Similarity'*, *Trivial*=1.0), this student's tendency for max-analogy (slice *t-1*, *'Analogy'*, *max*=.9) and low knowledge of the rule necessary to generate the copied step (slice *t-1*, '*R:body*', *True*=.2). In slice *t+1,* the student produced a correct entry specifying the normal force, including its direction ('*F:normal-dir*' node). The corresponding copy node is observed to *NoCopy*. This happens because the superficial similarity (*'Similarity'*, *NonTrivial* = 1.0) makes it impossible to copy and still generate a correct solution entry.

Assessment of EBLC Episodes. These episodes are used to assess the evolution of student knowledge. Currently, the model does not have direct evidence of positive instances of EBLC, and so aims to assess it by taking into account the following factors: 1) similarity, encoding our assumption that non-trivial superficial differences have a higher potential to stimulate EBLC than trivial ones, 2) student tendency for

EBLC, and 3) knowledge, in that students who already know a rule do not need to generate it via EBLC. The impact of these factors is demonstrated by the differences in the model's assessment of EBLC in slices *t* and *t+1* in Fig. 4. In slice *t*, the probability of EBLC is low (*'EBLC'*, *True*=.12), because although the student has low prior knowledge of the rule (*'R:body'*, *True*=.2, slice *t-1*), she has a poor tendency for EBLC (*'EBLC Tend'*, *True*=.1, slice *t-1*) and the similarity type is trivial, allowing for the correct generation of the solution step even in the absence of the appropriate rule. In slice *t+1*, the probability of EBLC has increased (*'EBLC'*, *True* =.61), because this is the only process that would allow the student to overcome the non-trivial difference between the problem and example to generate the solution step correctly.

Analogy and EBLC Tendency Assessment. An assessment of these two meta-cognitive tendencies allows the E-A Coach to generate tailored interventions when needed. To assess analogy tendency, the model uses its appraisal of students' copying behaviors. For instance, lack of copying (slice *t+1*) decreases the model's belief in the student's tendency for max analogy (*'Analogy'*, *max*= .91 in slice *t* decreases to *max* =.87 in slice *t+1).* To assess EBLC tendency, the model uses its appraisal of EBLC episodes. For instance, given belief in occurrence of EBLC, belief in EBLC tendency increases (*'EBLC Tend'*, *True*=.11 in slice *t* increases to *True* =.16 in slice *t+1).*

Knowledge Assessment. The model assesses knowledge both diagnostically and causally. Knowledge is assessed diagnostically from students' problem-solving actions. Specifically, the corresponding fact nodes are observed, resulting in belief propagation to either the parent rule node or the copy node. If there is a high probability of copying, the copy node explains away much of the evidence coming from student input. For instance, in slice *t* in Fig. 4, the high probability of a correct copy (*'Copy'*, *Correct*=.96) associated with the fact node *'F: Block is body'* explains most of the evidence away from the corresponding rule node. In slice *t+1,* the copy node is observed to *NoCopy*, so the evidence does propagate up from the fact node to the rule (*'R:dir'*) and prerequisite nodes. The model also aims to assess student learning from EBLC in a causal fashion through the link between EBLC and rule node. For example, in Fig. 4, belief in EBLC in slice *t+1* increases the probability that the *'R:dir'* rule has been generated by the student.

4.3 Simulation Mode

One of the ways in which the E-A Coach supports effective APS is through example selection for students, the goal being to choose an example that maximizes learning while helping the student achieve problem-solving success. To meet this goal, the framework relies on its user model. Specifically, the network described in Section 4.2 is used to predict the impact of each example found in the E-A example pool on the student's problem-solving (PS) success and subsequent learning during APS.

To generate this prediction, the model simulates the student's reasoning and actions, as if the student was solving the target problem and had access to the candidate example. This means that for each problem-solving step required for the target problem's solution, appropriate nodes are added to the network to assess copy and EBLC behaviors, as in assessment mode. Unlike in assessment mode, however, the only evidence available in simulation mode corresponds to the similarity between the current problem and candidate example. This evidence is combined with the model's belief in the student's knowledge and tendency for analogy and EBLC to generate a prediction of that student's problem-solving success (the probabilities for the fact nodes, i.e., problem-solving steps) and consequent learning (the value of the rule nodes). Note that problem-solving success is achieved either if the student has the knowledge to generate the problem solution, or if the example helps her do so (through the absence of structural differences). Both predictions (problem-solving success, learning) are a factor of student characteristics (knowledge, EBLC and analogy tendencies), as well as the similarity between the problem and example.

Given the model's predictions of learning and problem-solving success, to choose an example in a principled manner, the framework relies on a decision theoretic approach. Specifically, to calculate the utility of a candidate example, utility nodes are added and linked to the network's rule and fact nodes (fragment shown in Fig. 5). The '*Learning'* utility node reflects the objective to maximize learning, and is calculated using individual utility nodes for each rule in the network. The '*PS Success'* utility node reflects the objective to ensure problem-solving success, and is calculated using individual utility nodes for each fact in the network. Finally, the overall utility is calculated by combing values of the individual utility nodes using a linearly-additive multi-attribute utility (MAU) node ('*Overall'* utility node, Fig. 5). This process is repeated for each example in the framework's example pool to find the one with maximum utility. A similar approach has been proposed to select tutorial actions [9]; here, we extend it to the example selection task.

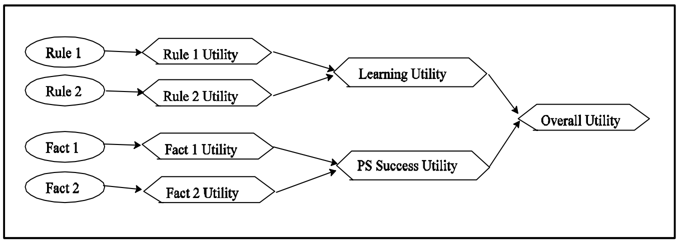


Fig 5: E-A Utility Model

5 Summary and Future Work

We have presented a framework aimed at providing support for meta-cognitive skills needed for effective analogical problem solving. To realize this support, the framework relies on its user model, which exploits a novel classification of similarity to infer the impact of an example on a student's relevant meta-cognitive skills. This assessment allows the framework to generate tailored interventions, including hints and example retrievals for students.

The next step in our research will be to evaluate the effectiveness of the proposed model with human participants. We plan to use the outcomes of these evaluations to refine the E-A user model, as well as to determine if additional factors should be incorporated into its assessment. We also intend to use these evaluations to assess the suitability of the proposed classification of similarity.

References

1. Aleven, V. and K.D. Ashley. *Teaching case-based argumentation through a model and examples: Empirical evaluation of an intelligent learning environment.* in *Artificial Intelligence in Education*. 1997. Kobe, Japan.
2. Anderson, J.R., *Rules of the Mind*. 1993, Hillsdale, NJ: Lawrence Erlbaum Associates.
3. Atkinson, R., S. Derry, A. Renkl, and D. Wortham, *Learning from Examples: Instructional Principles from the Worked Examples Research.* Review of Educational Research, 2002. **70**(2): 181-214.
4. Chi, M.T.H., M. Bassok, M.W. Lewis, P. Reimann, and R. Glaser, *Self-explanations: How students study and use examples in learning to solve problems.* Cognitive Science, 1989. **13**: 145-182.
5. Chi, M.T.H., P. Feltovich, and R. Glaser, *Categorization and representation of physics problems by experts and novices.* Cognitive Science, 1981. **5**: 121-152.
6. Chi, M.T.H. and K. VanLehn, *The content of physics self-explanations.* The Journal of the Learning Sciences, 1991. **1**: 69-105.
7. Conati, C., A. Gertner, and K. VanLehn, *Using Bayesian Networks to Manage Uncertainty in Student Modeling* . Journal of User Modeling and User-Adapted Interaction, 2002. **12**(4): 371-417.
8. Conati, C. and K. VanLehn, *Toward Computer-based Support of Meta-cognitive Skills: A Computational Framework to Coach Self-Explanation.* International Journal of Artificial Intelligence in Education, 2000. **11**: 389-415.
9. Murray, C. and K. VanLehn. *DT Tutor: A decision-theoretic dynamic approach for optimal selection of tutorial actions*. in *5'th Int. Conf. on ITS.* 2000. Montreal, Canada.
10. Novick, L.R., *Analogical transfer, problem similarity and expertise.* Journal of Experimental Psychology: Learning, Memory and Cognition, 1988. **14**: 510-520.
11. Quilici, J. and R. Mayer, *Role of Examples in How students Learn to Categorize Statistics Word Problems.* Journal of Educational Psychology, 1996. **88**(1): 144-161.
12. Reed, S.K., *A structure-mapping model for word problems.* Journal of Experimental Psychology: Learning, Memory and Cognition, 1987. **13**: 124-139.
13. Reed, S.K., A. Dempster, and M. Ettinger, *Usefulness of analogous solutions for solving algebra word problems.* Journal of Experimental Psychology: Learning, Memory and Cognition, 1985. **11**: 106-125.
14. Sweller, J. and G.A. Cooper, *The use of worked examples as a substitute for problem solving in learning algebra.* Cognition and Instruction, 1985. **2**(1): 59-89.
15. VanLehn, K., *Analogy Events: How Examples are Used During Problem Solving.* Cognitive Science, 1998. **22**(3): 347-388.
16. VanLehn, K., *Rule-Learning Events in the Acquisition of a Complex Skill: An Evaluation of Cascade.* The Journal of the Learning Sciences, 1999. **8**(1): 71-125.
17. VanLehn, K. and R.M. Jones, *Better learners use analogical problem solving sparingly*, in *Machine Learning: Proceedings of the Tenth Annual Conference*, P.E. Utgoff, Editor. 1993, Morgan Kaufmann: San Mateo, CA. 338-345.
18. Weber, G., *Individual Selection of Examples in an Intelligent Learning Environment.* Journal of Artificial Intelligence in Education, 1996. **7**(1): 3-33.