A computer framework to support self-explanation*

Cristina Conati
Intelligent Systems Program
University of Pittsburgh
3939 O'Hara St.,
Pittsburgh, PA, 15260

Jill H. Larkin
Departments of Psychology and Computer Science
Center for Innovation in Learning
Carnegie Mellon University
Pittsburgh, PA, 15213

Kurt VanLehn
Learning and Research Development Center
University of Pittsburgh
3939 O'Hara St.,
Pittsburgh, PA, 15260

Abstract

We present a computational framework for improving learning from examples by supporting self-explanation - the process of clarifying and making more complete to oneself the solution of an example. Many studies indicate that self-explanation can improve problem solving performance, and that guiding self-explanation can extend these benefits. Our goal is developing and testing a computer tutor — the SE (Self-Explanation) Coach — that can elicit and guide correct and effective self explanation, and thus improve problem solving performance in university-level Newtonian physics, a particularly complex and psychologically challenging domain. The self-explanations elicited by the SE Coach address how each component in the example solution can be justified in terms of (a) the theory of the instructional domain, and (b) the goal accomplished in the plan underlying the example solution.

The SE Coach provides the student with a Workbench that interactively presents examples and provides tools to construct self-explanations using the instructional domain theory. To guide self-explanation responsively, the SE Coach relies on a probabilistic student model, from which it assesses the student's understanding of an example.

The student model consists of a Bayesian network that generates its predictions by integrating information on self-explanations performed in the Workbench with information on the student's general domain knowledge and on the structure of the current example. By examining this network, the SE Coach identifies deficits in the student's self-explanations and can provide guidance to remedy them.

1. Introduction

Studying examples is a natural and common way of learning, and students use examples extensively when acquiring new skills [1, 2]. However, the learning benefits of examples are unclear, some studies showing great benefits [3], but others indicating only ability to solve problems similar to the original examples [4, 5].

However, self-explaining examples, that is analyzing and generating justifications of example steps, ordinarily produces both better problem solving performance and better understanding of domain

* This research is supported by AFORS grant F49620-96-1-0180, by ONR's Cognitive Science Division grants N00014-96-1-0260 and N00014-95-1-0950, and by ARPA's Computer Training and Education Initiative grant N66001-95-C-8367.

principles [6, 7, 8, 9, 10]. Furthermore, student-generated explanations seem to produce better learning than externally-provided explanations [7, 11].

This paper presents a computer tutor, the SE (Self-Explanation) Coach, which guides self-explanation, and thus aims to improve understanding and problem solving in university-level Newtonian physics, a particularly complex and psychologically challenging domain.

While human tutors have successfully elicited self-explanation [9, 12], the SE Coach is the first computer tutor aiming to improve learning by guiding self-explanation. Although ELM-PE [13] also uses examples to facilitate learning, it does not support self-explanation because the system, not the student, generates the explanations.

The SE Coach is part of the Andes tutoring system for physics, designed for use by the U.S Naval Academy Introductory Physics class. Andes guides students in example study and problem solving. Within Andes, the SE Coach ensures that students generate appropriate self-explanations so as to understand each component of an example solution In particular, the SE Coach focuses on two kinds of explanations that have appeared useful in self-explanation experiments: (a) explaining step correctness using the domain theory (here Newtonian physics); and (b) explaining step utility by identifying what goal each step satisfies in the plan underlying the example solution.

This paper describes the following components that contribute to the SE Coach's capacity to monitor and guide the student in the targeted self-explanations:

(1) A *Workbench* (described in Sec. 2) that interactively presents examples, prompts self-explanations, and provides principled tools for building them.

(2) A probabilistic Student Model (described in Sec. 3) that allows the SE Coach to decide when to elicit further self-explanations. The student model, (evolving from [14, 15]) consists of a Bayesian network [16], which assesses the student's example understanding by integrating: (a) Workbench actions reflecting the student's self-explanations, (b) estimates of the student's general domain knowledge, and (c) a representation of the current example.

Sec. 4 illustrates supported self-explanation, including how the SE Coach uses the assessments in the student model to prompt self-explanations that improve example understanding.

2. The Workbench

The SE Coach does not process natural language input, and needs other means to assess students' self-explanation. These include (a) a simple device to monitor the content and time durations of students' attention, and (b) an interface that reminds students to self-explain and provides raw materials for constructing self-explanations.

2.1 Attention monitoring and control

The Workbench hides example elements with opaque covers (Fig. 1). As a student moves the cursor over the screen, the corresponding covers immediately disappear and the text or diagram becomes visible. This device has two functions. First, it allows the student to focus on one element at a time, thus counteracting a tendency to read rapidly, without thorough processing. Second, it gives the SE Coach a record of viewed items with corresponding times.

Low viewing time suggests that little self-explanation has occurred. Although high viewing time may reflect more extensive explanations, it may also reflect confusion due to lack of knowledge, or merely doing something else (such as talking with a friend). Thus viewing times are suggestive evidence for self-explanation, with more explicit evidence coming from the additional mechanisms discussed next.

2.2 Supporting useful self-explanations

To remind students that self-explanation may be useful, an explain icon appears near the uncovered example part (Fig. 1). Clicking on this icon produces simple prompts to initiate self-explanations of correctness and utility.

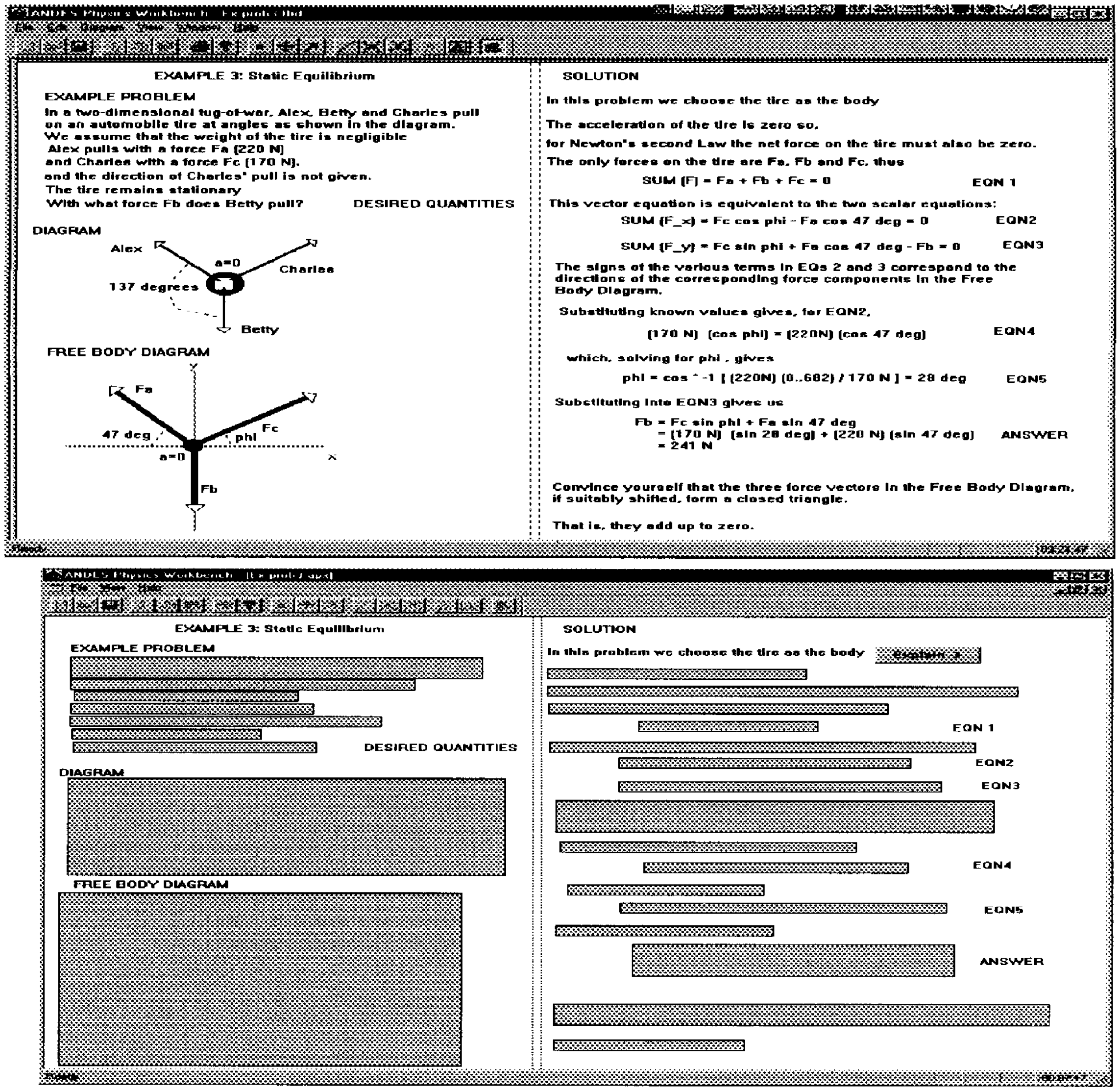

Figure 1: An example, including a problem and its solution (top). The Workbench presentation with attention-control panels and explain button (bottom).

For example, if the student clicks on the explain icon while viewing the solution line 1 in Fig 1, *we choose the tire as the body*, the SE Coach presents the prompt menu:

A. This step is useful in solving the problem because …
B. This choice of a body is appropriate because …

Similarly, for solution line 2, *The acceleration of the tire is zero*, the SE Coach presents:

C. This step is useful in solving the problem because …
D. This value for acceleration is correct because …

2.3 Building correct self-explanations

The Andes tutoring system includes physics and planning rules sufficient to solve problems in the instructional domain, and thus providing a complete specification of that domain knowledge. The Workbench guides correct and effective self-explanations by providing explanation-building tools linked to the Andes' rule-set. Fig. 2 illustrates some of these tools. A rule index shows Andes' planning and physics rules (Fig. 2a and 2b). To illustrate rule-index use, suppose a student reads *we choose the tire as the body* and selects prompt A above.

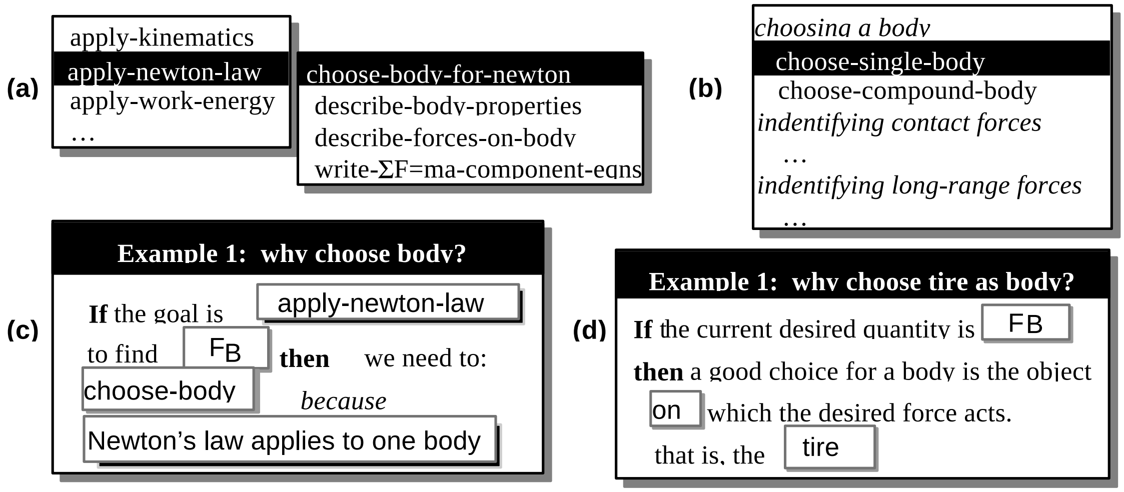

Figure 2: (a) Plan-rule index. (b) Physics-rule index. (c) and (d) Templates for applying rules to a particular example.

The student can use the Planning-rule index (Fig. 2a) to select the goal path leading to "choose-body-for-newton", thus explaining how this step fits in the goal structure of the plan to apply Newton's law to this example. To self-explain more extensively, the student may complete a template (Fig. 2c) describing how the planning rule behind the selected goal-stack applies to this example.

If the student chooses prompt B above (explaining why the tire is an appropriate choice for the body), then the hierarchical physics-rule index (Fig. 2b) allows finding a rule that generates this choice of body. Again, a rule template (Fig. 2d) supports more extensive self-explanation.

Workbench actions and the student model. All student's Workbench actions go to the SE Coach's student model, a Bayesian network which uses these actions to provide principled, integrated and continuously updated estimates of the student's understanding of the current example, as described in the next section.

3. The student model for the SE Coach

Fig. 3 shows a student-model fragment for the example in Fig. 1. The network *structure* (node content and links) represents Workbench actions (Fig. 3a), rules in the Andes system (Fig. 3b), and the example structure (Fig. 3c), including solution steps and rule applications producing them. Node *probabilities* represent dynamic estimates of related student's knowledge or actions. Value triplets in Fig. 3 reflect estimates at three points during a student's work (see Sec. 4). This section first describes the node types in the student model, and then their dynamic interactions.

3.1 Node types and corresponding knowledge in the Bayesian network

Workbench actions: Read-example and Self-explanation (SE) nodes. Read-example nodes (re- nodes in Fig. 3a) represent viewing example parts. Their three values indicate viewing time: *None* (not viewed), *Quick* (time sufficient to encode content, but too-short for self-explanation), and *Ok* (a plausible time for self-explanation).

SE nodes represent explicit self-explanation actions in the Workbench, and have binary values (*done/not-done*) indicating the corresponding actions have been performed. Some SE nodes (sel- nodes in Fig. 3a) represent selecting rule names (as in Fig. 2a and 2b). Others (inst- node in Fig. 3a) represent the instantiation of templates (as in Fig. 2d)[1].

General domain and planning knowledge: rule nodes. Rule nodes (r- nodes in Fig. 3b) correspond to rules in the Andes' rule-set, the physics and planning rules that produce example solutions and

[1] The inst-node for goal-setting (Fig. 2c) does not appear in Fig. 3 because it has complex connections in the network, and because the SE templates for planning rules are least well developed.

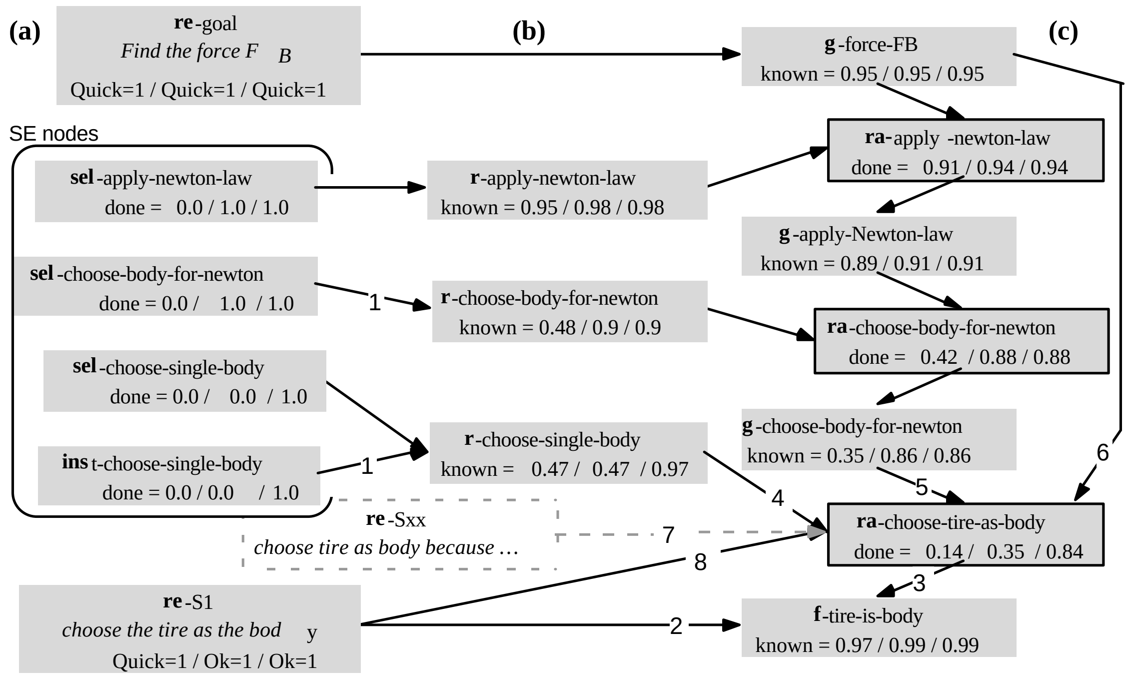

Figure 3: Fragment of the student model relevant to the example in Fig. 1. (a) Workbench actions (b) General domain knowledge. (c) Example structure. Node-value triplets correspond to the process-illustration in Sec. 4.

explanations. Probabilities for the binary values (*known/unknown*) of these nodes represent estimates that a student knows how to use the corresponding rules in the given problem solving context (for an more detailed description of the semantics of rule nodes see [15]). These values come from long-term observations of student's performance, in both example study and problem solving [15].

Example structure: goal, fact, and rule-application nodes. Andes includes a problem solver that uses Andes' rule-set to generate a principled example structure (see Fig. 3c), including (a) nodes representing facts and goals in the example solution (g- and f- nodes in Fig. 3c) and (b) rule-application nodes (ra- nodes in Fig. 3c) representing application of rules to produce these facts and goals.

The binary values (*known/unknown*) of goal and fact nodes encode the student-model estimate of whether a student knows the corresponding goals and facts. The binary values (*done/not-done*) of rule-application nodes encode estimates of whether the student has self-explained how goals and facts derive from general domain knowledge. As discussed below, the student-model produces these estimates by integrating knowledge of Workbench actions with estimates of the student's long-term knowledge of the instructional domain.

3.2 Structure of the Bayesian Network

Conditional probabilities for rule nodes. SE nodes represent Workbench actions designed to guide the student in using Andes' rules to generate self-explanations. Therefore each SE node connects to the rule node representing the rule that generates the corresponding explanation (e.g. links 1 in Fig. 3). This link, with the associated conditional probability table, reflects the fact that doing the Workbench action increases the estimate that the student knows the corresponding domain rule.

Conditional probabilities for fact and goal nodes. Fact and goal nodes often have two sources of input. First, read-example nodes connect to a fact or goal node reflecting the semantic content of the viewed element (e.g. link 2 in Fig. 3). These links indicate that viewing time influences the probability of knowing the related content. Second, a rule-application node connects to the fact or goal produced when the corresponding rule is applied (e.g. link 3 in Fig. 3). When a fact or goal node has input from both a read-example node and rule-application node (like f-tire-is-body in Fig. 3), then it may be known either through reading or through applying physics knowledge to earlier results. These alternate possibilities are represented in the conditional probability table for fact and goal nodes, as shown in Table 1a.

Table 1: Illustrative conditional probability tables for (a) goal and fact nodes and (b) rule-application nodes.

(a) g-/f-nodes, p(known)

Rule-appl-	Read-example None	Quick	Ok
done	1	1	1
not-done	0	<1	<1

(b) ra-nodes, p(done)

Rule	*Goal / Fact (pre-cond.)*	*Read-example (Rule result)* None	Quick	Ok
known	known	0.5	0.7	0.9
	unknown	0.0	0.0	0.0
unknown	known	0.0	0.0	0.0
	unknown	0.0	0.0	0.0

Conditional probabilities of rule-application nodes.

Since rule-application nodes reflect application of rules to generate an example solution, each rule-application node receives input from the corresponding rule node (e.g. link 4 in Fig 3) and from the goal and fact nodes representing the rule's preconditions (e.g. link 5 and 6). If an example describes the application of a rule, then the corresponding rule-application node gets input from the read-example node representing attention to this description (e.g. link 7 in Fig. 3, dotted because no such description appears in our example - see Fig. 1). If, however, the example solution presents only the rule-application result (often the case in real examples), then the rule-application node gets input from the read-example node representing attention to this result (link 8). This is because students generally start self-explaining an example part when they read it.

Table 1b, the conditional probability table for rule-application nodes, indicates that the probability of a rule-application node (i.e., the probability that the student has self-explained the corresponding inference) depends on both: (a) time spent attending to the result of the rule application, and (b) student's knowledge of both of the rule that generated the line and of its preconditions..

Model assessment of self-explanation. As probabilities automatically update through the Bayesian network, rule-application nodes reflect the model's assessment of how well the student understands the inferences required to explain the example. Thus, these nodes provide direct input to the SE Coach, from which it can decide when to prompt students for further self-explanation. The next section illustrates this process.

4. Coaching self-explanation

This section describes a hypothetical interaction between a student and the SE Coach, using the student model to suggest useful self-explanations.

Stage 0 - initializing the model. For the example in Fig. 1, the student model is set-up by clamping all the read-example nodes to *None* and the SE nodes to *not-done*. These probabilities, propagated in the network, reduce to *0.0* the probabilities of fact and goal nodes to be *known* and of application nodes to be *done*, indicating that the student currently has no example knowledge. Rule-nodes in the network (Fig. 3b) are set to estimates of student's long-term knowledge of Andes' rules. These estimates (left numbers in the triplets of Fig. 3b) indicate that the student knows the rule *r-apply-newton-law*, but probably cannot use the other two rules *(r-choose-body-for-newton* and *r-choose-single-body*).

Stage 1 - reading only. The student quickly reads the example goal and the first line of the solution and this clamps read-example nodes *re-goal* and *re-S1* to *Quick* (left values in the corresponding triplets in Fig 3a). The left numbers in the triples of Fig. 3c result from propagating these probabilities through in the network.

Even with a *Quick* viewing time, the student is likely know the goal and fact content *(g-force-*F_B and *f-tire-is-body*) of the viewed lines. The high probability of *g-force-*F_B, combined with the high probability of the rule node *r-apply-newton-law*, increases the probability that the student applies this rule and establishes the goal of applying Newton's second law to this example, as indicated by the increased probability of both the rule-application node *ra-apply-newton-law* and the resulting goal node *g-apply-newton-law*.

However, with *Quick* reading and poor knowledge of the rule *r-choose-body-for-newton*, the student is unlikely to apply this rule and to establish the goal to choose a body for Newton's law. Thus probabilities remain low for the rule-application node *ra-choose-body-for-newton* and for its output *g-choose-body-for-newton*.

Stage 2 - independent self-explanation. The student now views again line S1, ".…choose the tire as the body," clicks on the explain icon, selects the prompt,

A. This step is useful in solving the problem because …

and responds to it by selecting the goal stack [*top-goal*: apply-newton-law, *subgoal*: choose-body-for-newton] in the planning-rules index (Fig. 2a).

In Fig. 3, the center values now represent the state of the Bayesian network. The SE nodes *sel-apply-Newton-law* and *sel-choose-body-for-newton* are clamped to *done,* and the read-example node *read-S1* is clamped to *ok*. These probabilities propagate to increase the probability *of r-choose-body-for-newton*, a rule that already has a highly probable precondition, *g-apply-newton-law*. Thus the rule-application node *ra-choose-body-for-newton*, and the corresponding result (*g-choose-body-for-newton*) both acquire high probabilities, suggesting that the student now understands the function of choosing a body in the application of Newton's law to this example.

Stage 3 - coached self-explanation. When the student tries to move to another task, the SE Coach checks in the student model if there are rule-application nodes that have low probabilities, indicating self-explanation deficits for the current example. In the model including the central numbers in Figure 3 one rule-application has a low probability, *ra-choose-tire-as-body*. Therefore the SE Coach seeks to prompt self-explanation to remedy this deficit. Although prompt content is still being designed, it will probably start with a general suggestion such as: *Review the example solution more carefully. Make sure that you have understood how each line has been derived.* If necessary, the Se Coach can provide more specific prompts. Link 2 in Fig. 3, between *re-S1* and *ra-choose-tire-as-body*, tells the SE Coach to prompt self-explanation for line S1. The low probability of the rule *r-choose-single-body* suggests that the student is missing the knowledge necessary to generate the correct self-explanation. Thus the SE Coach may offer hints to improve knowledge of this rule.

Prompted by the coach, the student selects from the physics-rule index (Fig. 2b) the rule choose-single-body and also completes the corresponding template (Fig. 2d). The corresponding SE nodes (*sel-choose-single-body* and *inst-choose-single-body*) acquire the value *done*. Consequently, the probabilities of *r-choose-single-body* and *ra-choose-tire-as-body* increase, indicating that the student now has generated all relevant self-explanations, as indicated by the right numbers in the triplets for rule-application nodes in the Fig. 3c.

5. Conclusion

The goal of this work is to develop a tutoring framework, the SE Coach, that enhances learning from examples by guiding self-explanation. Although self-explanation has been successfully elicited by human tutors in the domains of Lisp programming [9] and the circulatory system [8], this work represents the first attempt to improve self-explanation by using a computer system and addressing a domain as complex and difficult to learn as Newtonian physics.

The SE Coach currently has the following modules: (a) A *Workbench* that supports correct and effective self-explanations through prompts to initiate self-explanation and tools to build self-explanations using the Andes' rule-set (a specification of the instructional domain). (b) A *Student Model* that uses a Bayesian network to combine in a principled way self-explanation actions in the Workbench with student's domain knowledge and knowledge of the example structure. This combined information generates a probabilistic assessment of the student understanding of the example. The SE Coach uses this assessment to decide whether to elicit further self-explanations.

The student model is currently under development for a small set of examples. A prototype interface (with attention-control panels, the explain icon and prototype prompts) was tested and performed well with individuals in the target audience (students at the U.S. Naval Academy).

The current central challenge is to implement and test a smoothly running coach, that uses the input from the student model to elicit effective self-explanation. In addition to the development of coaching strategies, this effort will require refining probabilities in the network, as we collect more data from pilot subjects.

The effectiveness of the SE Coach will be evaluated with students at the Naval Academy to assess whether: (a) actively guiding theory based self-explanation can increase students' problem solving performance in the challenging domain of Newtonian mechanics (compared to unguided study of examples); and (b) using a student model to prompt self-explanation produces better results than simple prompts with tools for building correct self-explanation (the Workbench alone).

7. References

1. Pirolli, P.L. and J.R. Anderson, The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology*, 1985. 39: p. 240-272.
2. VanLehn, K. Arithmetic procedures are induces from examples. In *Conceptual and Procedural Knowledge: The case of mathematics*, J. Hiebert, Editor. 1986, Lawrence Erlbaum Associates: Hillsdale, NJ.
3. Zhu, X. and H.A. Simon, Learning mathematics from examples and by doing. *Cognition and Instruction*, 1987. 4(3): p. 137-166.
4. 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: p. 59-89.
5. Catrambone, R., Aiding subgoal learning: effects on transfer. Journal of Educational Psychology, 1995. 87: p. 5-17.
6. Chi, M.T.H., M. Bassok, M.W. Lewis, P. Reimannet. al., Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 1989. 15: p. 145-182.
7. Needham, D.R. and I.M. Begg, Problem oriented training promotes spontaneous analogical transfer: memory-oriented training promotes memory for training. 1991. 19(6): p. 543-557.
8. Chi, M.T.H., N.d. Leeuw, M.-H. Chiu, and C. LaVancher, Eliciting Self-Explanations Improves Understanding. *Cognitive Science*, 1994. 18: p. 439-477.
9. Bielaczyc, K., P. Pirolli, and A.L. Brown, Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem-solving. *Cognition and Instruction*, 1995. 13(2): p. 221-252.
10. Renkl, A., Learning from worked-out examples. A study of individual differences. To appear in *Cognitive Science*, 1996.
11. Brown, A.L. and M.J. Kane, Preschool children can learn to transfer: Learning to learn and learning from example. *Cognitive Psychology*, 1988. 20: p. 493-523.
12. Chi, M.T.H., J.D. Slotta, and N.d. Leeuw, From things to processes: A theory of conceptual change for learning science concepts. *Learning and Instruction*, 1994. 4: p. 27-43.
13. Burrow, R. and G. Weber. Example Explanation in Learning Environments. In *Third International Conference on Intelligent Tutoring Systems*. 1996: Springer.
14. Conati, C. and K. VanLehn. Probabilistic Plan Recognition for Cognitive Apprenticeship. In *18th Annual Conference of the Cognitive Science Society, 1996, San Diego, CA.* New York: LEA.
15. Conati, C., A. Gertner, K. VanLehn, and M. Druzdzel. On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks. In *UM97, 6th International Conference on User Modeling*. 1997. Chia Laguna, Sardinia, Italy.
16. Pearl, J., *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. 1988, San Mateo, CA: Morgan-Kaufmann.