Providing Adaptive Support to the Understanding of Instructional Material

Cristina Conati
Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, BC, V6T1Z4, CANADA,
conati@cs.ubc.ca

Kurt VanLehn
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA, 15260, USA,
vanlehn@pitt.edu

ABSTRACT

We present an adaptive interface designed to provide tailored support for the understanding of written instructional material. The interface relies on a user model based on a Bayesian network, that assesses users' understanding as users read the instructional material and try to understand it by generating explanations to themselves. The user model's assessment is used by the interface to generate tailored scaffolding of further user's explanations that can improve the user's comprehension. After illustrating how the Bayesian user model assesses understanding from the user's explanations and from latency data on the user's attention, we discuss initial results on the effectiveness of the interface's adaptive interventions.

Keywords

Adaptive tutoring, user modeling, Bayesian networks, modeling attention.

1. INTRODUCTION

Being able to provide tutoring tailored to the needs of individual students has been identified as one on the main reasons for the effectiveness of Intelligent Tutoring Systems [16]. Much research has been devoted to provide adaptive guidance during problem solving [16], but other instructional activities may benefit from tailored support, because their effectiveness generally depends on the students' learning style and attitudes.

For instance, several cognitive science studies indicate that the amount of learning from instructional material depends on whether students spontaneously generate explanations to themselves (i.e. self-explain) as they read the material [2]. However, most students do not self-explain spontaneously, because they overestimate their understanding and/or do not use their knowledge to elaborate what they read [14]. To help these students, we have devised an educational environment, the SE-Coach, equipped with an interface providing the same adaptive guidance for self-explanation that has proven highly beneficial when administered by human tutors [2].

IUI'01, January 14-17, 2001, Sante Fe, New Mexico.

To provide this guidance, the interface must draw the students' attention to example parts that may be problematic for them and must be able to trigger self-explanations even from those students that do not have a tendency to self-explain. It is very important that the interface interventions be generated only when the student can benefit from the suggested self-explanations. Asking students to always make their self-explanations explicit to the system would burden the students who are natural self-explainers with unnecessary work, possibly compromising their motivation to deeply understand the instructional material. Thus, to determine when to intervene, the SE-Coach interface relies on a probabilistic student model (evolved from ideas proposed in [5]) that uses a Bayesian network [13] to assess how well students understand the instructional material by capturing both implicit self-explanation and self-explanations that the students generate via the interface.

Bayesian networks have been used in user modeling mostly to perform assessment during problem solving [4, 9, 11, 15]. We have extended their use to cover the sources of uncertainty involved in assessing learning from instructional material, in particular from example solutions. These sources of uncertainty include monitoring users' reading behavior through their focus of attention and assessing learning from users' explanations. Although there has been increasing interest in using data on users' attention to develop intelligent interfaces [7, 8], our model is one of the first to actively use these data in an adaptive system. Also, although other systems have been devised to improve learning by triggering users' explanations [1, 12], little research exists on how to monitor these explanations to assess learning.

In the rest of the paper, we briefly describe the SE-Coach's interface (see [6] for a more detailed description of the interface design and evaluation). We then illustrate how the student model Bayesian network is created automatically for each example. Next, we discuss how the SE-Coach's interface is dynamically modified by using the student model's assessment to elicit further self-explanation targeted to improve the user's example understanding. We finally discuss initial results on the effectiveness of the tailored support provided by the SE-Coach's interface.

2. INTERFACE FOR EXAMPLE STUDY

The SE-Coach interface allows students to read and self-explain example solutions (like the physics example in Figure 1) under the coach's supervision. Since eye-tracking and natural language understanding are still not powerful enough to reliably monitor these tasks, the SE-Coach interface includes two alternative mechanisms: a masking interface to track students attention and a set of menu based tools that allow students to constructively

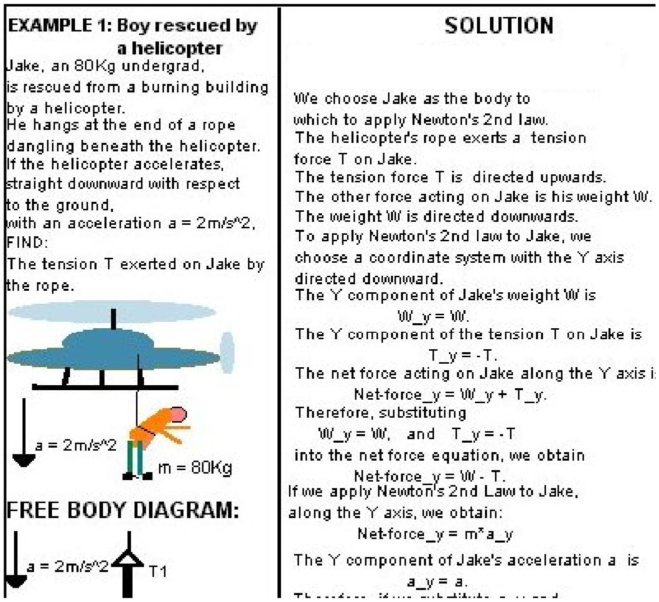


Figure 1: Sample physics example problem

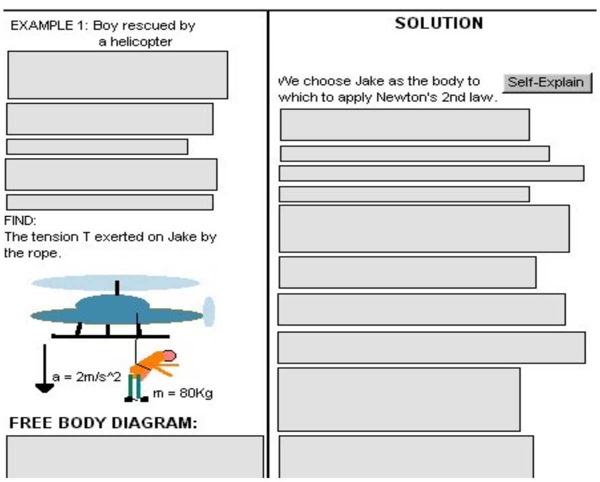


Figure 2: Example 1 shown with the masking interface

generate self-explanations. Figure 2 shows how the Newtonian physics example in Figure 1 is presented with the masking interface in the version of the SE-Coach implemented within the Andes tutoring system for Newtonian physics [17]. To view an example part, the student needs to move the mouse over the box that covers it. When the student uncovers an example part, a "self-explain" button appears, as a reminder to self-explain. Clicking on this button activates more specific prompts that suggest two kinds of self-explanations known to be highly effective for learning [14]. They are: (1) explain an example solution step in terms of domain principles (*step correctness*); (2) explain the role of a solution step in the underlying solution plan (*step utility*).

The interface provides tools to help students generate these two kinds of explanations. To explain *step correctness*, the student can activate a Rule Browser (see Figure 3a), containing a hierarchy of rules that represent principles in the instructional domain and reflect the content of the SE-Coach's knowledge base. The student can browse the rule hierarchy to find a rule that justifies the uncovered part and receives feedback on the correctness of her

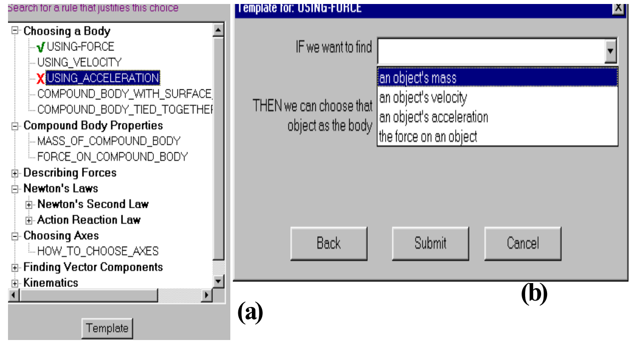


Figure 3: Rule Browser and a Template

selection. To explain more about a rule, the student can activate a Rule Template, a dialog box containing a partial definition of the rule that has blanks for the student to fill in (see Figure 3b). This definition includes preconditions for and consequences of the rule-application and reflects the rule definition in the SE-Coach knowledge base. Clicking on a blank in the Template brings up a menu of possible fillers. When the student submits a completed template, the SE-Coach will give feedback on the submission correctness. To explain *step utility*, the student can activate a Plan Browser that displays a hierarchical tree representing an example's solution plan. The student can search the hierarchy for the plan goal that most closely motivates the uncovered solution step and receives feedback on the correctness of the selected goal.

The student's reading and self-explanation actions are used to dynamically update the student model Bayesian network, that at any time during the interaction assesses the student's understanding of the different example parts. If a student tries to close the example when the student model indicates that there are still some parts that are problematic for him, the interface generates a warning and highlights the corresponding masking boxes (shown darker in Figure 4). It also changes the "self-explain" button for each highlighted line, to indicate what interface tools the student should use to better self-explain the line (see Figure 4). As the student performs new reading and self-explanation actions to follow the interface suggestions, the boxes' color and the related advice change dynamically to reflect the updates in the student model probabilities, as we will describe later.

3. THE SE-COACH'S STUDENT MODEL

3.1 Sources of Uncertainty

Several sources of uncertainty affect the SE-Coach's student model assessments and call for a probabilistic student model. The

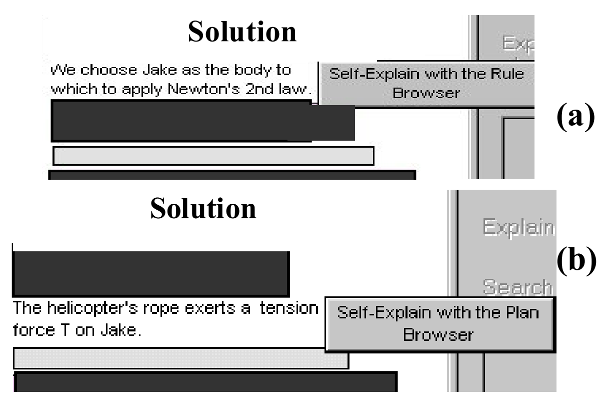


Figure 4: Adaptive prompts to self-explain

first source stems from the fact that we do not want to force spontaneous self-explainers to always make their self-explanations explicit through the SE-Coach interface tools. Thus, the student model must be able to detect self-explanations generated through implicit reasoning. The only evidence that the student model can use to do so is latency data on student's attention and estimates of student's domain knowledge. If a student spends enough time viewing a line of instruction and has sufficient knowledge to self-explain that line, we want the model to assess that likely the student self-explained the line correctly. But this adds uncertainty to the assessment, because both student's attention and student's knowledge cannot be unambiguously determined.

The second source of uncertainty exists because students have different studying styles. Some students prefer to generate solution steps by themselves before reading them in the example, while others rely completely on the example solution and start reasoning about a solution step only when they read it [14]. Hence, depending on the student's studying style, attention to a solution step can indicate not only self-explanation for that step but also derivation and self-explanation of subsequent steps.

The third source of uncertainty is due to the fact that sometime examples leave out some solution steps (i.e., they have *solution gaps*). A student needs to self-explain these steps to understand the solution thoroughly. This adds an additional level of implicit reasoning that the student model needs to assess from latency data and knowledge estimates.

Finally, little research exists on how people learn from menu selections and template filling. Thus, even when students generate correct self-explanations through the interface tools described in the previous section, there is uncertainty about how these self-explanations reflect learning and understanding.

The SE-Coach student model evolved from the Andes' student model for problem solving [4], to formalize and handle in a principled way these sources of uncertainty, as we describe in the next sections.

3.2 Generation of the Bayesian Network

The student model Bayesian network is built automatically for each example. This is crucial to allow the extension of the SE-Coach's set of examples with relatively limited effort, given that the resulting networks are quite large and complex (in the order of hundred of nodes). To add a new example, a problem solving module builds a model of correct self-explanation (SE model) for the example, starting from (i) a knowledge base of rules describing domain principles and abstract planning steps; (ii) a formal definition of the example givens. The resulting SE model (see Figure 5b) is a dependency network that encodes how the intermediate facts and goals (F- and G- nodes in Figure 5b) derive from domain rules (R- nodes in Figure 5b) and from facts and goals matching the rules' preconditions. These derivations are explicitly represented in the SE model by rule-application nodes (RA- nodes in Figure 5b) and correspond exactly to the self-explanations for *step correctness* and *step utility* that the SE-Coach targets. For instance, the node *RA-body-by-force* in Figure 5b encodes the explanation that Jake is chosen as the body because a physics rule says that if we want to find a force on an object, that object should be selected as the body to which to apply Newton's 2nd law. The node *RA-goal-choose-body* encodes the explanation that choosing Jake as the body fulfils the first subgoal of applying Newton's 2nd law, i.e. selecting a body to which to apply the law.

When a student opens an example, the student model Bayesian network for the current session is automatically created from the SE model and from a student's long-term model. The structure of the Bayesian network derives directly from the SE model. All nodes in the initial network have binary values representing the probability that the student knows rules, goals and facts in the example solution and that she has explained the related derivations. Rule node priors derive from estimates of a student's current domain knowledge stored in the long-term student model. This model also contains a probability η representing a student's tendency to reason ahead, namely to derive the example solution autonomously rather than reading it in the example. The probability η is used to specify the conditional probability table (CPT) for rule-application nodes, defined as a Noisy-AND. The Noisy-AND models the following assumptions: (1) a student

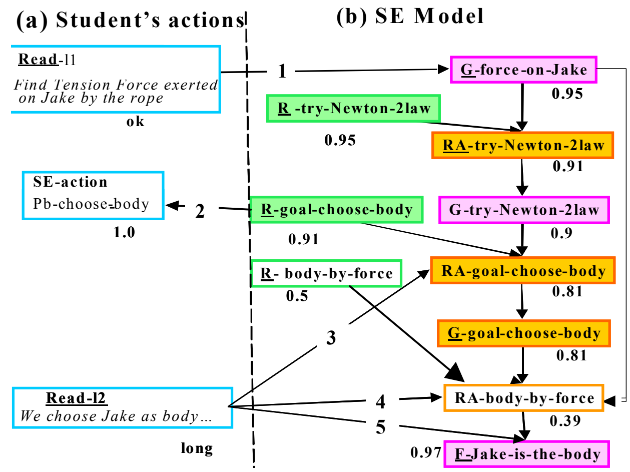


Figure 5: Segment of student model for example 1

cannot perform a derivation if she does not know the corresponding rule and its preconditions; (2) there is a probability (noise) that the student does not apply the rule even when she has all the relevant knowledge. This is exactly the probability that a student does not reason ahead, 1-η. Although η generally depends on the student only, it can sometimes depend on the rule as well. For instance, most students taking introductory physics understand when they need to apply Newton's 2nd law as soon as they read a problem's statement. Thus, the rule *R-try-Newton-2law* in Figure 5b is generally associated with a large η in the long-term student model. Fact and goal nodes (collectively called proposition nodes) deterministically depend on their parent rule-application. Priors of root proposition nodes, representing the example givens, are set to 0 until the student starts reading the example text.

As a student performs reading and self-explanation (SE) actions, the initial Bayesian network is dynamically updated with nodes representing these actions (see Figure 5a). In the next sections, we describe the semantics of the nodes representing student's actions and how they influence the probabilities of nodes in the SE model.

3.2.1 Read nodes.

Read nodes (nodes with prefix Read in Figure 5a) represent viewing items in the masking interface. The values of read nodes reflect the duration of viewing time and are set by comparing the total time a student spent viewing an item (TVT) with the minimum time necessary to read it (MRT). We currently compute the MRT for each item by assuming the reading speed of an average-speed reader (3.4 words/sec.,[10]). Depending upon the result of the comparison, the value of a read node can be LOW (TVT < MRT, time insufficient for reading), OK (TVT≈MRT, time sufficient for reading only) or LONG (TVT >> MRT, time sufficient for self-explanation).

Each read node connects to the SE model node reflecting the semantic content of the viewed item (see link 1 and 5 in Fig. 5). These links indicate that viewing time influences the probability of knowing the related content. When a proposition node has input from *both* a read node and a rule-application node (e.g., *F-jake-is-the-body* in Figure 5), then a student can acquire the corresponding proposition *either* by reading it in the example *or* by deriving it from rules and previous propositions. This relationship is represented in the CPT for proposition nodes whose content a student has viewed in the example solution (see Table 1). If a rule-application node is TRUE (i.e., the student explained the corresponding derivation), by definition the proposition node is TRUE (i.e., known by the student). Otherwise, the probability of knowing the proposition node is small if reading time is LOW (probability p_1 in Table 1) and high when the time is OK or LONG (probabilities p_2 and p_3 respectively in Table 1) .

Table 1: CPT for a proposition node the student viewed

READ		*LOW*	*OK*	*LONG*
RULE-APPL	*T*	1.0	1.0	1.0
	F	$p_1 < 0.5$	$p_2 > 0.9$	$p_3 > p_2$.

Some proposition nodes in the Bayesian network may not be connected to a read node, even after the student has viewed all the elements in the example solution. This happens when the solution omits some of the steps, as it often happens in instructional material. In Figure 5, for instance, the nodes *G-try-Newton-2law* and *G-goal-choose-body* cannot have any read node pointing to them, because these goals are not explicitly mentioned in the example. A student can know unmentioned propositions only by deriving them from rules and other example propositions.

The fact that the student viewed an example item does not necessarily mean that the student self-explained it. In particular, it does not mean that the student self-explained the inferences describing how the item derives from a domain principle (*step correctness)* and what goal it achieves in the solution plan (*step utility*). However, the longer the student viewed an example item, the higher the probability that he self-explained it. This relationship between viewing time and self-explanation is encoded in the student model by linking a read node that represents viewing an example item with the rule-application nodes that represent self-explanation for correctness and utility for that item (see, for instance, link 3 and 4 in Figure 5). The CPT for these rule-application nodes is modified to take reading time into account, as shown in Table 2 (For simplicity, this table shows a rule with only one precondition.) A student cannot self-explain a derivation correctly if he does not have the necessary knowledge (i.e., the rule and its preconditions), no matter for how long the student attended to the derivation (these cases are grouped under the *otherwise* entry in Table 2).

Table 2: CPT for rule-application node after a student viewed the related proposition node

Knows RULE	Knows Goal/Fact	READ		
		LOW	*OK*	*LONG*
T	*T*	$p_1 = \eta$	$p_2 = \eta$	$p_3 > \max\{p_2, 0.9\}$
otherwise		0	0	0

If the student has all the necessary knowledge, the probability that proper self-explanation occurred increases with viewing time. If viewing time is LOW (i.e., insufficient for reading) or OK (i.e. sufficient for reading only), self-explanation for this line could only have occurred if a student reasoned forward from previous lines. The probability that this happened (represented by probabilities p_1 and p_2 in Table 2) is the probability η in the long term student model, modeling the student's tendency to derive solution lines autonomously, before reading them in the example. Thus,

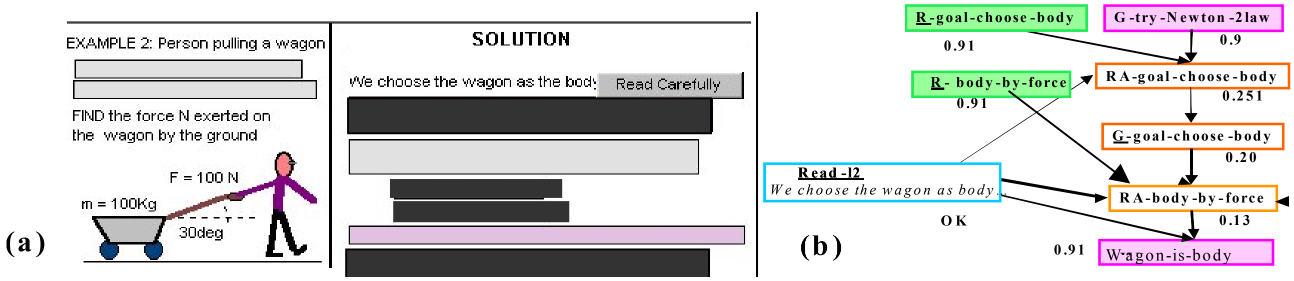
$$P(RA = T|Rule = T, Goal = T, Read \in \{LOW, OK\}) = \eta$$

as shown in Table 2. If viewing time is LONG, the probability that self-explanation occurred (p_3 in Table 2) is set to be at least 0.9.

3.2.1 SE Nodes

Nodes representing the student's self-explanation actions (SE nodes) are dynamically added to the Bayesian network to model the occurrence of Rule Template filling and selection of goals in the Plan Browser. Since Rule Template filling provides evidence of the student's understanding of the corresponding rule, the SE node for a Template filling action is linked to the corresponding rule node in the Bayesian network. Similarly, the SE nodes encoding goal selection in the Plan Browser are linked to the rule nodes establishing the corresponding goals in the SE model with (see link 2 in Figure 5).

Since SE actions such as menu selections and template filling involve building self-explanations by composing given material, they do not provide as strong evidence of rule understanding as self-explanations generated verbally would. Even if a student has little knowledge of a rule, he may still be able to generate a self-explanation involving that rule by using the SE-Coach tools, because of the scaffolding that these tools provide. However, whether the student learns from the self-explanation action depends upon how much constructive reasoning she performs during the process. Given the lack of established results on how people learn from menu selections and template filling, in the SE-Coach student model we prefer to be conservative. We assume that each correct SE action increases the probability that a student learns the corresponding rule, but if the student starts with low rule knowledge we want evidence from more than one correct SE action before assessing that the rule is mastered.



(a) (b)

Figure 6: Transfer of rule probabilities to a new example

To achieve this behavior without having to change the CPT of a rule node every time a new SE action involving that rule occurs, we direct the link from rule nodes to SE nodes (see Figure 5a). The entry P(SE=T| Rule=F) in a SE node's CPT represents the probability that a student can complete a correct SE action without knowing the corresponding rule. This probability can be adjusted to vary the amount of evidence that a correct SE action provides toward rule learning. For instance, the higher the number of wrong attempts a student makes before generating a correct SE action, the higher we set P(SE node=T| Rule=F) when we add the corresponding SE node, because it becomes more likely that the student achieved the correct action through random selection in the interface tools, rather than through reasoning.

4. USING THE STUDENT MODEL TO GENERATE TAILORED SCAFFOLDING

At any time during the student's interaction with the SE-Coach, the probabilities in the Bayesian network assess how the student's domain knowledge and example understanding change through the student's interface actions. In particular, the probabilities associated with rule-application nodes represent the probability that the student self-explained the corresponding derivations. Rule-application nodes with probability below a given threshold become the target of the SE-Coach interventions.

Figure 5 shows the probabilities in the student model after a student viewed the line "We want to find the tension force on Jake" long enough for reading it, viewed the line "We choose Jake as the body" quite longer and self-explained the latter with the Plan Browser. The high probability of the node *G-try-Newton-2law*, not mentioned in the example solution, and of its parent rule-application node, show how the Bayesian network can model forward reasoning and filling solution gaps. The high probabilities of the nodes *R-try-Newton-2law* and *G-force-on-Jake* propagate downward because of the large η assigned to the rule *R-try-Newton-2law* (as we described in the previous section), which translates into a small noise (1-η) for the Noisy-AND CPT of the rule-application node *RA-try-Newton-2law.*

In Figure 5, there is only one rule-application node that still has a low probability, the non-shaded node *RA-body-by-force*. From this node's descendant, *F-Jake-is-the-body*, the SE-Coach infers that the missing explanation relates to the first line in the example solution. From the fact that the only input node with low probability for *RA-body-by-force* is the rule *R-body-by-force*, the SE Coach detects that the missing explanation relates to this rule. Hence, it adds the first solution line among the lines to highlight in the masking interface and modifies its self-explain button to suggest self-explanation with the Rule Browser (see Figure 4a). Similarly, when the SE-Coach detects that the low probability of a rule-application node is due to low probability of the closest planning rule ancestor, it modifies the interface to suggests self-explanation through the Plan Browser (see Figure 4b). When low probability of a rule-application node is caused only by too short reading time, the "self-explain" button for the related solution line is turned into a hint suggesting to read more carefully.

The student is not obligated to follow the interface suggestions. When the student decides to close an example, the student model Bayesian network is discarded, but the new rule nodes' probabilities are used to update the long term student model. These probabilities will become the new priors in the student model for the next example study task and will influence the system's interventions accordingly. Let's suppose, for instance, that the student generates both the Plan Browser and Rule Browser explanations for the segment of example in Figure 5 and then opens a new example, shown in Figure 6a. The first line of this example's solution ("We choose the wagon as the body") is similar to the one for the previous example, as similar is the corresponding segment of Bayesian network (shown in Figure 6b). However, the priors of the rule nodes involved in this part of the model reflect the student's previous interaction with the system. If the student views the first example line for long enough, the student model will predict that the student has performed all the inferences required to self-explain this line, because there is high probability that she has the knowledge necessary to do so. Thus, the SE-Coach will never ask this student to explicitly self-explain the selection of the wagon as the body by using the interface tools. The only prompting that the interface may generate on this example line would be caused by viewing time too short for self-explanation. In this case, the Read node modeling attention to the body-selection line would be set, for instance, to OK, preventing the high probability of the rule nodes *R-goal-choose-body* and *R-body-by-force* to propagate to the corresponding rule-application nodes (see Figure 6b) and resulting in the generation of a *read-carefully* prompt for the body-selection line (see Figure 6a). If the student then reviews the line for sufficient time, the student model will be dynamically updated and the line will be turned back to an "explained" state (light gray) when the student moves away from it.

In summary, the SE-Coach interface prompts students to explicitly generate self-explanations using its tools only when the student model assesses that this can improve the student's understanding. In particular, the interface helps those students who do not self-explain because they do not monitor their understanding, by drawing the students' attention to example parts that may be problematic for them and by providing specific scaffolding on what knowledge these explanations should tap. Asking students to always make their explanations explicit through the interface tools would of course enable more accurate assessment of their understanding, but would also burden the students who are natural self-explainers with unnecessary work, possibly compromising the effectivevess of their self-explanations and their motivation to use the system.

5. EVALUATION OF THE INTERFACE ADAPTIVE INTERVENTIONS

To evaluate the effectiveness of the SE-Coach, we conducted a study in which 29 subjects studied Newton's 2nd Law examples with the complete SE-Coach (experimental condition), while 27 subjects studied the same examples with the masking interface only and no coaching (control condition). All subjects took a pretest and a posttest consisting of Newton's 2nd Law problems. At the time of the experiment, all subjects were taking introductory physics in different colleges and had started studying Newton's laws in class. In this section, we present a more detailed analysis of preliminary data on the usability of the SE-Coach interface that we obtained from this study and reported in [6]. The analysis focuses on the appropriateness of the interface adaptive interventions by showing how they influenced the performance of students in the experimental group (the only ones that had these interventions). More general data on the difference between the performance of the experimental and the control group can be found in [3].

5.1 Student Model Set Up

During the evaluation, we only used examples with limited gaps in the presented solution. That is, the example solution mentioned all the steps derived from physics rules (fact nodes in the student model), and omitted only the most immediate goal that each step fulfills. Thus, this study does not give us information on how well the student model presented here handles the additional level of uncertainty introduced by the higher degree of implicit reasoning involved in studying examples with larger solution gaps. Also, constraints on the study duration prevented us from initializing the student model with data on the students' initial knowledge and studying style. Hence, we assigned to most rules a prior of 0.5, and we assumed that students are very unlikely to reason forward (η = 0.02) because other studies [14] show that this is how most students usually behave. The threshold for considering rule nodes as known was set to 0.9 and the conditional probabilities for SE nodes were set in such a way that a correct self-explanation action achieved at the first attempt would make the corresponding rule node reach the threshold if the rule probability was 0.5 or higher. Given the lack of formal theories on how people learn from menu selection and template filling actions, this choice is not informed but rather dictated by the rationale that when we do not have good estimates of what initial knowledge students have (as the priors of 0.5 indicate), we prefer to be less conservative in assessing student learning from correct SE actions, to compensate for possible inaccuracies in the model set up.

5.2 Results

We computed from log data how often students followed the interface adaptive prompts to further self-explain. The results are summarized in Table 3. For each type of prompt, the table reports: (i) the maximum number of prompts that could appear in the interface for the three examples in the study. These are the prompts the interface would generate if there was no student model. (ii) The average number of prompts adaptively generated by relying on the student model. (iii) The average percentage of these prompts the students followed. After reporting these results in [6], we computed the correlation between the percentages of followed prompts and students' post-test scores, controlling for pretest scores. All three measures significantly (or nearly significantly) correlate with post-test performance ($p = 0.056$ for Rule Browser/Template prompts; $p = 0.024$ for Plan Browser prompts; $p = 0.016$ for "Read more carefully" prompts). These data provide an initial indication that the adaptive prompts based on the student model effectively elicited self-explanations that improve students' learning, although further data should be gathered to control for other variables that might have caused the correlation, such as general academic ability or conscientiousness.

Table 3: SE-Coach prompts that students followed

Prompt Type	Max.	Generated	Followed
Use RuleBrowser/Templ.	43	22.6	38.6%
Use PlanBrowser	34	22.4	42.0%
Read mode carefully	43	7	34.0%

A second result that we obtained from further analysis of the results presented in [6], indicates how accurately the student model assesses knowledge changes from SE actions. We found an interaction between the accuracy of this assessment and when subjects had started studying Newton's Laws in their classes. We computed the correlation between posttest scores and the number of rules that reached high probability in the student model. The correlation is very low ($r = -0.03$) for subjects from classes that had started the example topic more than a week before the study (early-start subjects) and it is higher (= 0.33) for subjects from classes that had started just a few days before (late-start subjects). This indicates that the evidence that correct SE actions provide toward rule knowledge in the student model reflects more accurately how late-start subjects learned from these actions. Since our data showed no significant differences in the two groups' initial knowledge or in how they used the interface tools, we hypothesize that the difference in the correlation exists because the SE-Coach examples were more challenging for late-start subjects and therefore they put more effort than the early-start ones in reasoning and learning from the same SE actions. These results suggest that the students' learning stage should be taken into account when modeling learning from SE actions. For instance, in the CPT for SE nodes, the probability $P(SE=T|Rule=F)$ (e.g. the probability of generating a correct SE action without knowing the corresponding rule), should be set to an higher value if a student has been working on the example topic for some time (like our early-start subjects), to account for the possibility that a lower level of motivation may result in less learning from SE actions.

6. CONCLUSIONS AND FUTURE WORK

Providing interactive, tailored support to the understanding of instructional material will become increasingly important as more instruction will be delivered through computers and distant learning on the Web. We have described a computational framework that provides tailored support to the understanding of instructional material presenting example solutions. The framework relies on a probabilistic student model that takes into account the different sources of uncertainty embedded in assessing understanding from latency data on attention, and from explanations generated through menu selections and template filling. Although there has been extensive research on monitoring user's attention, the work described in this paper is one of the first attempts to actively use latency data to adapt the interaction to the user's behavior. To our knowledge, it is also the first attempt to assess learning from students' explanations.

We have discussed initial results on the effectiveness of the adaptive support to example studying that the SE-Coach interface provides by using the student model's assessments. The results indicate that, despite the lack of accurate initial parameters in the student model, this support improved student's learning. The results also suggest how to improve the accuracy of the student model by taking into account the students' learning stage.

Although the probabilistic student model currently captures attention through a masking interface and explanations through menu based tools, its structure and assessment are independent from the input modality. For example, as research on eye tracking and natural language progresses, the model can be modified to monitor latency data and explanations through these modalities. This can be done by changing the conditional probabilities in the model, to reflect the more accurate evidence that eye-tracking and verbal explanations may provide on user's attention and understanding. Since eye tracking and natural language provide less scaffolding for reading and self-explanation, we plan to explore what types of learners benefit more from an interface that relies on them and what learners benefit more from the current, more constrained interface. This will allow us to explore how to dynamically adapt the interaction mode to the users' learning style. We also plan to work on the automatic generation of examples that tailor the number of solution gaps to the student's current knowledge, to provide adaptive support to the transition from example studying to problem solving.

7. REFERENCES

[1] Aleven, V., K.R. Koedinger, and K. Cross. Tutoring answer-explanation fosters learning with understanding. in AIED '99, 9th World Conf. of Artificial Intelligence and Education. 1999. Le Mans, France.

[2] Chi, M.T.H., Self-Explaining: The dual process of generating inferences and repairing mental models. Advances in Instructional Psychology, in press.

[3] Conati, C. and Vanlehn, K., Further Results from the Evaluation of an Intelligent Computer Tutor to Coach Self-Explanation in Proc of the Ninth Int. Conf. on Intelligent Tutoring Systems, ITS 2000, Montreal.

[4] Conati, C., Gertner, A., VanLehn, K., and Druzdzel, M. On-line student modeling for coached problem solving using Bayesian networks, in User Modeling: Proc. of the Sixth Int. Conf., A. Jameson, C. Paris, and C. Tasso, Editors. 1997, Spring Wien: New York.

[5] Conati, C., J. Larkin, and K. VanLehn, A computer framework to support self-explanation, in Proceedings of the Eighth World Conf. of Artificial Intelligence in Education. 1997.

[6] Conati, C. and K. VanLehn. Teaching meta-cognitive skills: implementation and evaluation of an tutoring system to guide self-explanation while learning from examples. in AIED'99, 9th World Conf. of Artificial Intelligence and Education. 1999. Le Mans, France.

[7] Gluck, K.A., J.R. Anderson, and S.A. Douglass. Broader Bandwidth in Student Modeling: What if ITS were "Eye"TS? in ITS 2000, 5th Int. Conf. on Intelligent Tutoring Systems. 2000. Montreal, CA.

[8] Horvitz, E. Principles of Mixed-Initiative User Interfaces. in CHI '99, ACM SIGCHI Conf. on Human Factors in Computing Systems. 1999. Pittsburgh, PA.

[9] Jameson, A., Numerical uncertainty management in user and student modeling: An overview of systems and issues. User Modeling and User-Adapted Interaction, 1995. **5**.

[10] Just, M. and P. Carpenter, The psychology of reading and language comprehension, ed. A.a. Bacon. 1986, Boston.

[11] Luckin, R. and B. du Boulay. Capability, potential and collaborative assistance. in UM '99: User Modeling. 1999. Banff, Canada: SpringerWienNewYork.

[12] Okamoto, T. and T. Kasai. The collaborative system with situated agents for activating observation learning. in ITS2000, 5th Int. Conf. on Intelligent Tutoring Systems. 2000. Montreal, Canada.

[13] Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. 1988, San Mateo, CA: Morgan-Kaufmann.

[14] Renkl, A., Learning from worked-examples: A study on individual differences. Cognitive Science, 1997. **21**(1): p. 1-30.

[15] Reye, J. Two-phase updating of student models based on dynamic belief networks. in 4th Int. Conf. on Intelligent Tutoring Systems (ITS '98). 1998. S. Antonio, Texas.

[16] Shute, V.J. and J. Psotka, Intelligent tutoring systems: Past, Present and Future, in Handbook of Research on Educational Communications and Technology, D. Jonassen, Editor. 1996, Scholastic Publications.

[17] VanLehn, K., Conceptual and meta learning during coached problem solving, in ITS96: Proceeding of the Third Int. Conf. on Intelligent Tutoring Systems., C. Frasson, G. Gauthier, and A. Lesgold, Editors. 1996, Springer-Verlag: New York.