The Impact of Student Model Updates on Contingent Scaffolding in a Natural-Language Tutoring System

Patricia Albacete[1](✉), Pamela Jordan[1], Sandra Katz[1],
Irene-Angelica Chounta[2], and Bruce M. McLaren[3]

[1] Learning Research and Development Center,
University of Pittsburgh, Pittsburgh, PA, USA
palbacet@pitt.edu
[2] Institute of Education, University of Tartu, Tartu, Estonia
[3] Human Computer Interaction Institute, Carnegie Mellon University,
Pittsburgh, PA, USA

Abstract. This paper describes an initial pilot study of Rimac, a natural-language tutoring system for physics. Rimac uses a student model to guide decisions about *what content to discuss next* during reflective dialogues that are initiated after students solve quantitative physics problems, and *how much support to provide* during these discussions—that is, domain contingent scaffolding and instructional contingent scaffolding, respectively. The pilot study compared an experimental and control version of Rimac. The experimental version uses students' responses to pretest items to initialize the student model and dynamically updates the model based on students' responses to tutor questions during reflective dialogues. It then decides what and how to discuss the next question based on the model predictions. The control version initializes its student model based on students' pretest performance but does not update the model further and assigns students to a fixed line of reasoning level based on the student model predictions. We hypothesized that students who used the experimental version of Rimac would achieve higher learning gains than students who used the control version. Although we did not find a significant difference in learning between conditions, the experimental group took significantly less time to complete the pilot study dialogues than did the control group. That is, the experimental condition led to more efficient learning, for both low and high prior knowledge level learners. We discuss this finding and describe future work to improve the tutor's potential to support student learning.

Keywords: Dialogue-based tutoring systems · Student modeling · Contingent scaffolding

1 Introduction

The key features of instructional scaffolding, as described by [12], include *contingency*, *fading* and, correspondingly, the gradual *transfer of responsibility* for learning and successful performance to the learner. "Contingency" refers to the adaptive nature of

© Springer Nature Switzerland AG 2019
S. Isotani et al. (Eds.): AIED 2019, LNAI 11625, pp. 37–47, 2019.
https://doi.org/10.1007/978-3-030-23204-7_4

scaffolding and is believed to be its core feature, from which the other two features stem. Instructors dynamically adjust their degree of control over the learning task according to their diagnosis of the student's current level of understanding or performance [14]. "Fading" refers to the gradual release of this support so that scaffolding can achieve its ultimate aim: to shift responsibility for successful performance to the student.

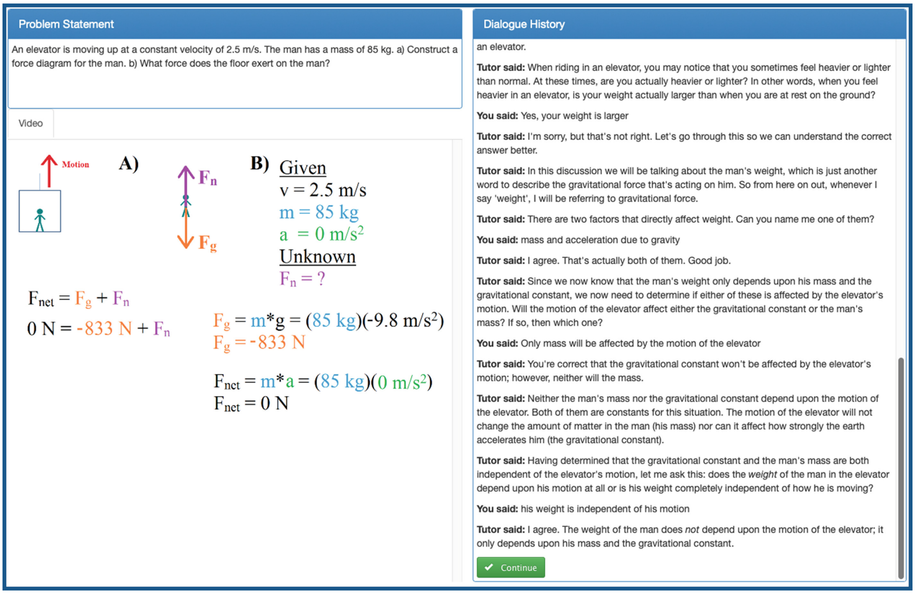
Wood and Wood [14] distinguished between three types of contingency during human tutoring sessions: *temporal*, *domain*, and *instructional contingency* (see also [13]). Temporal contingency is concerned with deciding *when* to intervene versus letting the learner struggle for a while or request help. Domain contingency is concerned with choosing appropriate content to address during an intervention, while instructional contingency is concerned with deciding how to address focal content—for example, in how much detail and through which pedagogical strategies (e.g., modeling, hinting, explaining, question asking)?

For the Rimac natural-language tutor [1, 2, 5, 9], we developed an Instructional Factors student model [4] that dynamically updates throughout the tutorial dialogue in order to represent the student's current level of understanding. The student model is used during decision-making about domain and instructional contingency. We compared this version of Rimac to a version that uses a static representation of the student's understanding based solely on the student's pretest performance, i.e., to a version that uses an array of knowledge components initialized with pretest scores as a student model, to make decisions about domain and instructional contingency. We predicted that classroom students who interacted with the version of Rimac that incorporates the adaptive student model would show greater learning gains than those who interacted with a version of Rimac that incorporates a simple static representation of a student's level of understanding. A student model that reflects students' progress should lead to more appropriate decisions regarding domain and instructional contingency. To our knowledge, this is the first real-time test of an Instructional Factors Model (IFM) being used by an ITS to tutor students in the classroom.

2 Rimac: An Adaptive Natural-Language Tutoring System

Rimac is a dialogue-based tutoring system that engages high school students in conceptual discussions after they solve quantitative physics problems (e.g., [1, 2, 10]). These dialogues are developed using an authoring framework called *Knowledge Construction Dialogues* (KCDs) (e.g., [6, 7, 11]). KCDs present a series of carefully ordered questions known as a *Directed Line of Reasoning* (DLR) [6], which guide students in responding to complex conceptual questions (reflection questions, or RQs). When the student makes an error at a particular step in the DLR, the tutor initiates a remedial sub-dialogue to address that error. Figure 1 shows the system's interface which presents, in the left pane, the problem statement along with a sample solution to a quantitative problem that students watch as a video and, in the right pane, an excerpt of a reflective dialogue between the system and the student which addresses conceptual knowledge associated with the quantitative problem.

Rimac adapts its instruction to students' ever evolving knowledge by incorporating a student model that is updated as the student engages in the dialogues and by implementing policies that, with the help of the student model predictions, allow it to choose the next question to ask at the appropriate level of granularity and with adequate support. The granularity level refers to domain contingency—that is, how much content is explicitly discussed with the student (e.g. discuss all the steps in the reasoning vs skip over some steps that the student can likely infer on her own). Adequate support refers to instructional contingency—that is, how much detail should be provided in questions and hints about the selected content.



An elevator is moving up at a constant velocity of 2.5 m/s. The man has a mass of 85 kg. a) Construct a force diagram for the man. b) What force does the floor exert on the man?

B) Given

$v = 2.5$ m/s

$m = 85$ kg

$a = 0$ m/s^2

Unknown

$F_n = ?$

$F_{net} = F_g + F_n$

$0\ N = -833\ N + F_n$

$F_g = m*g = (85\ kg)(-9.8\ m/s^2)$

$F_g = -833\ N$

$F_{net} = m*a = (85\ kg)(0\ m/s^2)$

$F_{net} = 0\ N$

an elevator.

Tutor said: When riding in an elevator, you may notice that you sometimes feel heavier or lighter than normal. At these times, are you actually heavier or lighter? In other words, when you feel heavier in an elevator, is your weight actually larger than when you are at rest on the ground?

You said: Yes, your weight is larger

Tutor said: I'm sorry, but that's not right. Let's go through this so we can understand the correct answer better.

Tutor said: In this discussion we will be talking about the man's weight, which is just another word to describe the gravitational force that's acting on him. So from here on out, whenever I say 'weight', I will be referring to gravitational force.

Tutor said: There are two factors that directly affect weight. Can you name me one of them?

You said: mass and acceleration due to gravity

Tutor said: I agree. That's actually both of them. Good job.

Tutor said: Since we now know that the man's weight only depends upon his mass and the gravitational constant, we now need to determine if either of these is affected by the elevator's motion. Will the motion of the elevator affect either the gravitational constant or the man's mass? If so, then which one?

You said: Only mass will be affected by the motion of the elevator

Tutor said: You're correct that the gravitational constant won't be affected by the elevator's motion; however, neither will the mass.

Tutor said: Neither the man's mass nor the gravitational constant depend upon the motion of the elevator. Both of them are constants for this situation. The motion of the elevator will not change the amount of matter in the man (his mass) nor can it affect how strongly the earth accelerates him (the gravitational constant).

Tutor said: Having determined that the gravitational constant and the man's mass are both independent of the elevator's motion, let me ask this: does the *weight* of the man in the elevator depend upon his motion at all or is his weight completely independent of how he is moving?

You said: his weight is independent of his motion

Tutor said: I agree. The weight of the man does *not* depend upon the motion of the elevator; it only depends upon his mass and the gravitational constant.

Fig. 1. Rimac interface. Problem statement shown in upper left pane, worked example video in lower left pane, and dialogue excerpt in right pane.

An individual learner's student model is built in two steps: first, using the results of the student's pretest, a clustering algorithm classifies the student as low, medium, or high. The purpose of this initial clustering is to increase the accuracy of the student model's predictions. Second, the student is assigned a cluster-specific regression equation that is then personalized with the results of the student's pretest. The regression equation assigned to the student represents an implementation of an Instructional Factor Analysis Model (IFM), as proposed by [4]. This student model uses logistic regression to predict the probability of a student answering a question correctly as a linear function of the student's proficiency in the relevant knowledge components (KCs). Additionally, as the student progresses through the dialogues, her student model is dynamically updated according to the correctness of her responses to the tutor's questions [5].

To be able to vary the level at which the tutorial discussions are conducted, for each reflection question (RQ), we developed dialogues at three different levels of granularity: an expert level (P—primary) which only includes the essential steps of the reasoning, a medium level (S—secondary), and a novice level (T—tertiary) which includes more basic knowledge such as definitions of concepts and laws. Figure 2 shows a graphic representation of an excerpt of a line of reasoning (if the net force on an object is zero then the object's velocity is constant) at three different levels of granularity.

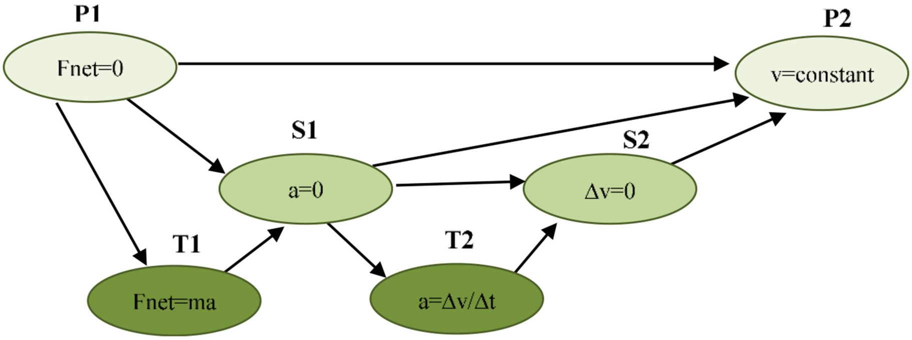


Fig. 2. Graphical representation of the line of reasoning Fnet = 0 → v = constant with different levels of granularity. Nodes represent questions the tutor could ask. Arcs represent the knowledge (KCs) required to make the inference from one node to the next.

After the tutor asks the student a reflection question, it first needs to decide if the student is knowledgeable enough to skip the discussion all together. To this end, if the student answers the reflection question correctly, the tutor consults the student model and if the student is predicted to know the relevant knowledge pertaining to the RQ with a probability of 80% or higher, she is considered to have mastered the target knowledge and is allowed to skip the RQ. On the other hand, if the student either does not answer the RQ correctly or has not mastered its relevant knowledge, the tutor engages in a reflective dialogue with the learner. At each step of this discussion, the tutor needs to decide at what level of granularity it will ask the next question in the line of reasoning (LOR) (or in a remedial sub-dialogue if the previous question was answered incorrectly) in order to proactively adapt to the student's changing knowledge level. It performs this adaptation by following policies aimed at driving the student to reason in an expert-like manner while providing adequate scaffolding. Hence, the tutor will choose a question in the highest possible granularity level that it deems the student will respond to correctly or that it perceives will be in the student's zone of proximal development (ZPD)—"a zone within which a child can accomplish with help what he can later accomplish alone" [3].

To make this choice, Rimac consults the student model, which predicts the likelihood that the student will answer a question correctly. The tutor interprets this probability in the following way: if the probability of the student responding correctly is higher than 60% then the student is likely to be able to respond correctly, and if it is lower than 40% the student is likely to respond incorrectly. However, as the prediction gets closer to 50%, there is greater uncertainty since there is a 50% chance that she will be able to answer correctly and a 50% chance that she will answer incorrectly. This uncertainty on the part of the tutor about the student's ability could indicate that the student is in her ZPD with regards to the relevant knowledge. Hence the tutor perceives the range of probabilities between 40% and 60% as a model of the student's ZPD [5]. Thus, the tutor will choose to ask the question in the highest possible level of the LOR that has a predicted probability of at least 40% of being answered correctly [2]. The exception to this policy is for questions belonging to the expert level LOR. For those questions, the tutor takes a more cautious approach and only asks them if it is quite certain that the student will answer them correctly, i.e., if the predicted probability of the student answering the expert level question is equal to or greater than 60%.

The expression of each question within the LOR is adapted to provide increased support as the certainty of a correct answer decreases [9]. For example, the tutor can ask a question directly with little support such as, "What is the value of the net force?" or with more support by expressing it as "Given that the man's acceleration is zero what is the value of the net force applied on the man?" In the latter case, the object is named concretely and a relevant hint ("Given that the man's acceleration is zero") is included, making this second version of the question less cognitively demanding.

3 Testing the System

3.1 Conditions

Two versions of the system were developed to use as control and experimental conditions. The control version used a "poor man's" student model that consisted of an array of KCs initialized with a score based on the student's pretest performance and that score did not vary throughout the study. Additionally, when students started a reflection question, they were assigned to a fixed LOR level (expert, medium, or novice) based on the correctness of their response to the RQ and on their KC scores according to the algorithm shown in Fig. 3.

The experimental condition used the adaptive version of the system described in previous sections, which embeds a student model that updates its estimates as the dialogue progresses and implements domain and instructional contingent scaffolding.

3.2 Participants

Students from a high school in Pittsburgh, Pennsylvania, in the U.S. were recruited to participate in the study. They were taking a college preparatory class (though not honors or Advanced Placement) that covered the topics discussed in the system. Students were randomly assigned to the control and experimental conditions and used the system as an in-class homework helper, hence the system was used after the material had been covered in class. A total of 73 students participated in the study; N = 42 were in the control condition and N = 31 in the experimental condition. The imbalance in the number of participants was due to students missing school and hence not completing the study (a t-test revealed no pretest difference between students who completed the study and those that did not, p = .471).

3.3 Materials

Using the experimental and control versions of the system, students solved 5 problems with 3–5 reflection questions per problem on the topic of dynamics. A pretest and isomorphic posttest (i.e., the pretest and corresponding posttest items only differed in their cover stories) were developed. The tests consisted of 35 multiple-choice test items that were presented online and automatically graded, though students did not receive feedback on the correctness of their answers. The test items were conceptual questions that tested the KCs associated with tutor's reflection questions but were not similar to the homework problems which required quantitative solutions as seen in the sample problem solution in Fig. 1. Students were given 30 min to complete the tests.

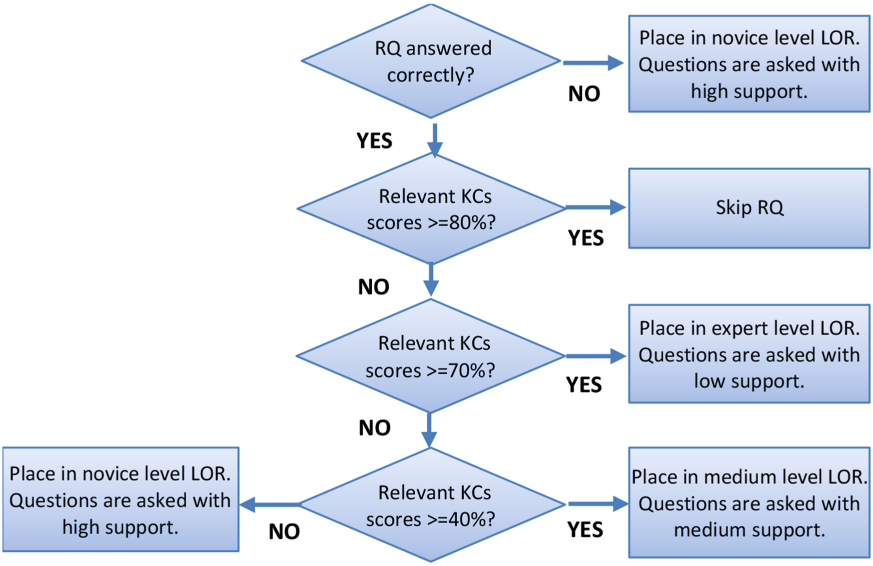

RQ answered correctly?
NO
Place in novice level LOR. Questions are asked with high support.
YES
Relevant KCs scores >=80%?
YES
Skip RQ
NO
Relevant KCs scores >=70%?
YES
Place in expert level LOR. Questions are asked with low support.
NO
Relevant KCs scores >=40%?
NO
Place in novice level LOR. Questions are asked with high support.
YES
Place in medium level LOR. Questions are asked with medium support.

Fig. 3. Flow chart showing behavior of control condition

3.4 Protocol

Students started by taking the online pretest. After the pretest, they interleaved solving homework problems on paper with using the system in the following way: First, students solved on paper the quantitative homework problem presented by the system; second, they viewed a video of a sample solution to that problem on the system as feedback (the video contained no discussion of conceptual material); third, students engaged in conceptual dialogues with the tutorial system which addressed the conceptual aspects of the quantitative problem they had just attempted to solve. After all problems were completed, students took the online posttest and a short satisfaction survey. The entirety of the study was performed in class over the course of 4 days. All students took the pretest on Day 1 and the posttest on Day 4 and worked on the homework problems at their own pace on Days 1–3.

3.5 Results

Our main hypothesis is that students in the experimental condition would learn more than those in the control condition due to the system's proactive adaptation of scaffolding to students' evolving needs. To test this hypothesis, we started by evaluating whether students in each condition learned from interacting with the system. Then we compared the mean learning gains between conditions and checked for an aptitude treatment interaction. Finally, we compared time on task between conditions.

Did students in each condition learn from interacting with the system? To answer this question a paired-samples t-test was performed comparing the mean scores of the pretest to those of the posttest in each condition. The tests revealed a statistically significant difference between mean pretest scores and mean posttest scores for students in both conditions suggesting that students learned from interacting with the system. Table 1 shows the results.

Table 1. Pretest vs. Posttest scores

Condition	Pretest mean SD	Posttest mean SD	*t*(n)	*p*	Cohen's *d*
Experimental	M = .505 SD = .093	M = .592 SD = .091	*t*(30) = 6.540	<.001	1.2
Control	M = .503 SD = .091	M = .615 SD = .089	*t*(41) = 7.565	<.001	1.2

Did students in one condition learn more than in the other? To investigate whether one version of the system fostered more learning than the other, we first performed an ANCOVA with Condition as fixed factor, prior knowledge (as measured by pretest) as covariate, and Posttest as the dependent variable. The results of this test suggest that condition had no statistically significant effect on posttest when controlling for the effects of prior knowledge, $F(1,70) = 1.770$, $p = .19$ Additionally, we performed an independent samples t-test comparing the mean gain from pretest to posttest between conditions. No statistically significant difference was found between the mean gain of the experimental condition (M = .087, SD = .074) and the mean gain of the control condition (M = .112, SD = .096), t(71) = 1.226, $p = .22$. The results of the t-test and ANCOVA suggest that students in both conditions learned equally. We also evaluated whether the incoming knowledge—as measured by pretest score—of students in each condition was comparable. An independents sample t-test revealed no statistically significant difference in students' prior knowledge between conditions $t(71) = .127$, $p = .90$.

Did the effectiveness of the treatment vary depending on students' prior knowledge? In other words, was there an aptitude-treatment interaction? To study this issue, we performed a regression analysis using Condition, Pretest, and Condition*Pretest (interaction term) as independent variables and gain as the dependent variable. The regression coefficient of the interaction term was not significant suggesting no aptitude-treatment interaction $F(1,69) = 1.456$, $p = .23$.

Was one version of the system more efficient than the other? To investigate this possibility, we compared the mean time that students spent working on the system[1] between conditions by performing an independent samples t-test. The test revealed that the mean time on task of the experimental condition (M = 51.26 min, SD = 12.44 min) was significantly shorter than the mean time on task of the control condition (M = 71.52 min, SD = 16.42 min), $t(71) = 5.754$, $p < .001$, Cohen's $d = 1.4$.

A closer look at time on task: Was the experimental system more efficient than the control system for students of *all* incoming knowledge levels? In a prior study where we compared a version of Rimac that used a "poor man's" student model (similar to the control condition of this study) to a version of Rimac that did not have a student model and had all students go through the novice LOR, we found the system with the student model was significantly more efficient than the system without the student model, but *only* for high prior knowledge students [8]. Hence, we decided to investigate if in the current study the experimental version was more efficient than the control for students of *all* levels of incoming knowledge. To this end, we partitioned the students in each condition into those with high incoming knowledge and those with low incoming knowledge using a median split. We then compared the time on task of high prior knowledge students in the control and experimental groups. To that end we performed an ANOVA which revealed that the mean time of task of high pretesters in the experimental group was 31% (20.8 min) shorter than in the control group, a statistically significant difference. Similarly, when comparing time on task for low prior knowledge students between conditions, an ANOVA revealed a 27% time on task difference in favor of the experimental condition which was statistically significant. See results in Table 2 and Fig. 4.

Table 2. Comparison of time on task (TOT) between conditions for high and low incoming knowledge students

Student prior kw	Condition	N	Mean TOT (min)	SD TOT (min)	F	*p*	Cohen's *d*
Low	Control	21	74.72	14.82	F(1,35) = 18.29	<.001	1.4
	Experimental	16	54.78	12.95			
High	Control	21	68.33	17.66	F(1,34) = 16.201	<.001	1.4
	Experimental	15	47.51	11.09			

[1] Time on task did not include the time students spent solving the problems on paper. Additionally, any inactivity longer than three minutes while a student worked on the system was not counted towards the time on task estimate since it could be indicative that the student had taken a break from the learning activity.

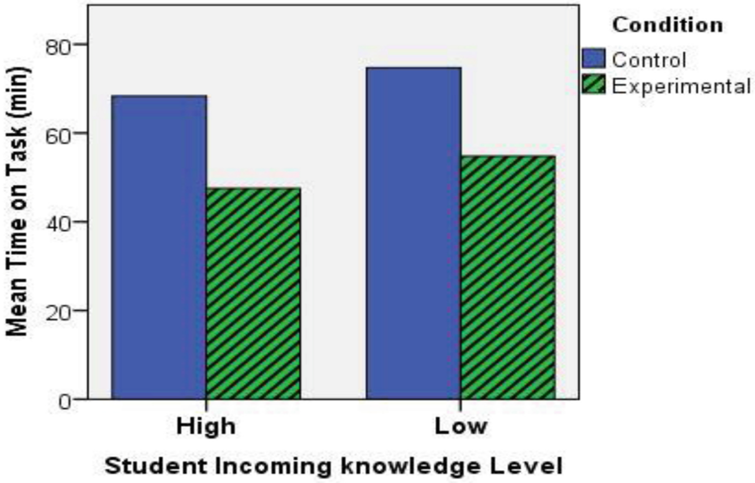


Fig. 4. Comparison of time on task between conditions for High and Low prior knowledge students.

4 Discussion and Future Work

In this paper we report on the comparison of two versions of Rimac to explore the effectiveness of incorporating a student model that is dynamically updated throughout the interaction to enable domain and instructional contingency during tutorial dialogues. One version of Rimac (experimental version) proactively adapts the content it discusses as well as the amount of support it provides during its interaction with the student by using the predictions of a student model that dynamically updates its assessment of students' understanding of particular KCs as the student progresses through the dialogues. The second version of Rimac (control version) sets the student on a fixed line of reasoning, rather than adapting to the students' evolving knowledge needs, based on the student's initial response to the reflection question under consideration and on the predictions of a static student model that only considers the student's pretest performance. We found that students in both conditions learned equally well. One possible reason this may have occurred is that regardless of the level of line of reasoning at which students are placed in the control system, if they lack the necessary knowledge to answer a question correctly, they are presented with a remedial sub dialogue that covers the knowledge subsumed in the lower level LORs. Hence, it is possible that the fixed LOR with its remediations were enough for students to have comparable knowledge gains as in the more adaptive, experimental condition.

The key finding of this work is that students who used the system with the dynamic student model (i.e., the experimental system) learned more efficiently, that is, in less time, than those who used the system with the static student model (i.e., control version). Of particular interest is the discovery that students with low incoming knowledge in the experimental condition were able to go through all the dialogues 27% faster (on average, experimental condition: 55 min, control condition: 75 min) than those in the control condition. This suggests that a dynamic student model is more effective than a static one in supporting domain and instructional contingency. The dynamic student model is able to effectively adjust to the students' evolving knowledge allowing them to traverse higher level lines of reasoning—which are shorter—as their knowledge improves, thereby saving them time. In contrast, a static student model will keep the granularity of the discussions with the students at the level defined by their incoming knowledge regardless of improvements in their knowledge that occur during the dialogues.

In future work, we plan to compare the adaptive system with two less adaptive versions of the system to try to separate on the one hand, the effect on learning of updating the student model during the dialogues and, on the other hand, the effects of providing domain and instructional contingency. In the first study, we will perform a more in-depth analysis of the impact that the student model's dynamic updates have on students' learning by isolating the evaluation of this feature. We will compare the current experimental version of the system with a control condition that would perform exactly the same way as the experimental version—i.e., deciding at what level to ask the next question and with how much support to express it rather than placing students in a fixed LOR—except that it would choose the next question based on the predictions of the static KC scores derived from the pretest rather than on the dynamically updated model. In the second study, we will evaluate more precisely the value of performing domain and instructional contingency (i.e., deciding what to ask and how to ask it on each step of the dialogue) by comparing the current version of the experimental condition with a control condition that improves on the flexibility of the one presented in this paper by placing students in fixed low, medium or high levels of lines of reasoning not just when the student answers the reflection question correctly (as in the current study) but also when the student answers it incorrectly. This may allow Rimac to place a student who may have slipped when answering the RQ in a more appropriate LOR level. The comparison of these versions of Rimac might provide additional evidence of the value of implementing scaffolding that contains domain and instructional contingency.

Acknowledgments. We thank Sarah Birmingham, Dennis Lusetich and Scott Silliman. This research was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A150155 to the University of Pittsburgh. The opinions are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

References

1. Albacete, P., Jordan, P., Katz, S.: Is a dialogue-based tutoring system that emulates helpful co-constructed relations during human tutoring effective? In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS (LNAI), vol. 9112, pp. 3–12. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19773-9_1
2. Albacete, P., Jordan, P., Lusetich, D., Chounta, I.A., Katz, S., McLaren, B.M.: Providing proactive scaffolding during tutorial dialogue using guidance from student model predictions. In: Penstein Rosé, C., et al. (eds.) AIED 2018. LNCS (LNAI), vol. 10948, pp. 20–25. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93846-2_4

3. Cazden, C.: Peekaboo as an Instructional Model: Discourse Development at Home and at School. Stanford University Department of Linguistics, Palo Alto (1979)
4. Chi, M., Koedinger, K.R., Gordon, G.J., Jordon, P., VanLehn, K.: Instructional factors analysis: a cognitive model for multiple instructional interventions. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., Stamper, J. (eds.) EDM 2011, pp. 61–70 (2011)
5. Chounta, I.-A., Albacete, P., Jordan, P., Katz, S., McLaren, B.M.: The "Grey Area": a computational approach to model the zone of proximal development. In: Lavoué, É., Drachsler, H., Verbert, K., Broisin, J., Pérez-Sanagustín, M. (eds.) EC-TEL 2017. LNCS, vol. 10474, pp. 3–16. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66610-5_1
6. Evens, M., Joel, M.: One-on-One Tutoring by Humans and Computers. Psychology Press, New York (2006)
7. Graesser, A.C., Lu, S., Jackson, G.T., et al.: AutoTutor: a tutor with dialogue in natural language. Behav. Res. Methods **36**, 180–192 (2004)
8. Jordan, P., Albacete, P., Katz, S.: Adapting step granularity in tutorial dialogue based on pretest scores. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) AIED 2017. LNCS (LNAI), vol. 10331, pp. 137–148. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_12
9. Katz, S., Albacete, P., Jordan, P., Lusetich, D., Chounta, I.A., McLaren, B.M.: Operationalizing Contingent Tutoring in a Natural-Language Dialogue System. Nova Science Publishers, New York (2018)
10. Katz, S., Albacete, P.: A tutoring system that simulates the highly interactive nature of human tutoring. J. Educ. Psychol. **105**(4), 1126–1141 (2013)
11. Rosé, C., Jordan P., Ringenberg, M., Siler, D., VanLehn, K., Weinstein, A.: Interactive conceptual tutoring in Atlas-Andes. In: AIED 2001, pp. 151–153(2001)
12. van de Pol, J., Volman, M., Beishuizen, J.: Scaffolding in teacher–student interaction: a decade of research. Educ. Psychol. Rev. **22**, 271–296 (2010)
13. Wood, D.: The why? what? when? and how? of tutoring: the development of helping and tutoring skills in children. Literacy Teach. Learn. **7**(1/2), 1–30 (2003)
14. Wood, D., Wood, H.: Vygotsky, tutoring and learning. Oxford Rev. Educ. **22**(1), 5–16 (1996)