Linking Dialogue with Student Modelling to Create an Adaptive Tutoring System for Conceptual Physics

Sandra Katz
University of Pittsburgh
Learning Research and Development Center
Pittsburgh, PA, USA
Corresponding author email: katz@pitt.edu

Patricia Albacete
University of Pittsburgh
Learning Research and Development Center
Pittsburgh, PA, USA

Irene-Angelica Chounta
University of Tartu
Center for Educational Technology
Tartu, Estonia

Pamela Jordan
University of Pittsburgh
Learning Research and Development Center
Pittsburgh, PA, USA

Bruce M. McLaren
Carnegie Mellon University
Human-Computer Interaction Institute
Pittsburgh, PA, USA

Diego Zapata-Rivera
Educational Testing Service
Princeton, NJ, USA

Acknowledgements. We thank Sarah Birmingham, Dennis Lusetich, and Scott Silliman for many valuable contributions to this project. Insightful comments from the reviewers and editors on a previous draft broadened our thinking about the state of TDSs and important "next steps" that need to be taken. 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 expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education.

Linking Dialogue with Student Modelling to Create an Adaptive Tutoring System for Conceptual Physics

Abstract

Jim Greer and his colleagues argued that student modelling is essential to provide adaptive instruction in tutoring systems and showed that effective modelling is possible, despite being enormously challenging. Student modelling plays a prominent role in many intelligent tutoring systems (ITSs) that address problem-solving domains. However, considerably less attention has been paid to using a student model to personalize instruction in tutorial dialogue systems (TDSs)—ITSs that engage students in natural-language, conceptual discussions. This paper describes Rimac, a TDS that tightly couples student modelling with tutorial dialogues about conceptual physics. Rimac is distinct from other TDSs insofar as it dynamically builds a persistent student model that guides reactive and proactive decision making in order to provide adaptive instruction. An initial pilot study set in high school physics classrooms compared a control version of Rimac without a student model with an experimental version that implemented a "poor man's student model"; that is, the model was initialized using students' pretest scores but not updated further. Both low and high prior knowledge students showed significant pretest to posttest learning gains. However, high prior knowledge students who used the experimental version of Rimac learned more efficiently than high prior knowledge students who used the control version. Specifically, high prior knowledge students who used the student model driven tutor took less time to complete the intervention but learned a similar amount as students who used the control version. A subsequent study found that both high and low prior knowledge students learned more efficiently from a version of the tutor that dynamically updates its student model during dialogues than from a control version that included the static "poor man's student model." We discuss future work needed to improve the performance of Rimac's student model and to integrate TDSs in the classroom.

Keywords: tutorial dialogue systems, adaptive instruction, scaffolding, physics education

1. Introduction

> …*so difficult is the task, that it is tempting to conclude that maybe it would be better to try to avoid student modelling all together, to search for some magical "end run" around the need to understand the learner at all. Unfortunately, whether the tutoring system is an "old fashioned" present-and-test frame-based tutor, a deeply knowledgeable AI-based expert advisor, or a scaffolding environment situated in the learner's world, it must adapt to the learner or be forever condemned to rigidity, inflexibility, and unresponsiveness.* (Greer and McCalla 1991, p. viii)

This statement appears in the preface to a collection of papers on student modelling edited by Jim Greer and Gordon McCalla entitled, "Student Modelling: The Key to Individualized Knowledge-Based Instruction," which stems from a 1991 NATO workshop by the same name. It captures the editors' resolute stance that some form of student modelling is an essential ingredient of an adaptive tutoring system. Much of Jim's work with his colleagues showed developers of intelligent tutoring systems (ITSs) that "the problem of student modelling" (p. v) is manageable, despite being enormously difficult. Promising results from user studies of the ITSs that they developed, most of which were driven by Bayesian student models, proved this challenge to be worth tackling (Zapata-Rivera and Greer 2004b). These results inspired the next generation of ITS developers, including the authors of this paper, to develop new approaches to student modelling, and/or to implement student modelling in tutors for students of various ages, across subject domains. [See Pavlik et al. (2013) for a review.]

Most ITSs address problem-solving domains. These tutors rely on student modelling to achieve macro- and micro-adaptation, according to the two-loop model of tutoring behavior that VanLehn (2006) proposed—that is, to choose appropriate tasks for a student (*macro*-adaptation) and to provide feedback and support, as needed, at each task step (*micro*-adaptation). Many student model guided ITSs have shown promising learning outcomes (Aleven et al. 2016a; Conati and Kardan 2013; Desmarais and d. Baker 2012; Mitrovic 2012; Pavlik et al. 2013; Shute 1995). Some ITSs, such as the Cognitive Tutors, have been providing effective instruction to thousands of students (e.g., Blessing et al. 2009; Koedinger and Corbett 2006). However, there is one genre of ITSs that has lagged behind problem-solving tutors with respect to its use of student modelling to drive adaptive instruction: tutorial dialogue systems (TDSs). TDSs engage students in natural-language conceptual discussions with an automated tutor. (For example, see Table 1 and Fig. 1.)

Student modelling's low profile in tutorial dialogues systems, relative to problem-solving ITSs, is not surprising, given that TDSs add the challenge of natural-language understanding to two well-established student modelling problems: the inherent uncertainty of the model's assessments and the fact that students' understanding evolves as they interact with a tutoring system. The student model must constantly be updated to keep pace with this "moving target" (Greer and McCalla 1991). In addition, it is not straightforward to apply approaches to student modelling used in problem-solving ITSs to TDSs, due to characteristics intrinsic to tutorial dialogue. First, pairs of tutor-student dialogue turns do not always represent a step during problem solving. Instead, some dialogue turns may contribute relevant background knowledge and thus may require a finer-grained representation in order to track students' knowledge. Second, conceptual discussions do not neatly map to the structured steps typical of problem solving, as illustrated in Tables 1 and 2. (See also Fig. 1.)

Despite these challenges, student modelling is just as important for TDSs as it is for any other tutoring system, for the reasons that Greer and McCalla (1991) stated: to thwart "rigidity, inflexibility, and unresponsiveness" (p. viii). This paper describes work that we have been doing to more tightly couple student modelling with automated tutorial dialogue than has been done in most TDSs, with the goal of making these tutors more adaptive and efficient. Our motivation stems largely from classroom-based studies that we conducted while developing Rimac, a prototype tutorial dialogue system for conceptual physics. Rimac aims to enhance high school students' understanding of concepts associated with quantitative physics problems that they solve on paper.[1] (See Table 1 and Fig. 1.) Several studies examined the relation between particular tutoring strategies, student characteristics and learning, in order to derive adaptive tutoring policies (decision rules) to implement in Rimac (e.g., Jordan et al. 2015a, 2015b, 2018; Jordan et al. 2012; Katz and Albacete 2013; Katz et al. 2016; Katz et al. 2018). We consistently found significant pretest to posttest learning gains, regardless of which condition students were assigned to—that is, which version of a tutoring strategy or policy students experienced (e.g., a high vs. low frequency of restatement, different types of summarization, different ways of structuring remediation). In addition, several studies revealed interesting aptitude-treatment interactions (Jordan et al. 2015a, 2018; Katz et al. 2016). However, many students complained that Rimac's dialogues are "too long," during informal interviews and on user surveys. More concerning was feedback indicating that the tutor was insufficiently adaptive and inefficient—specifically, that Rimac's dialogues often spend too much time on concepts that the student understands and too little time on concepts that they have been struggling with. Other TDS developers have reported similar user feedback (e.g., Kopp et al. 2012).

We realized that the only way to address the problems of inefficiency and inadequate personalization in TDSs would be to model students' understanding of the content addressed during discussions with the tutor and to use these models to guide the system in making adaptive decisions. In other words, we decided to take the torch that developers of ITSs without natural-language interaction had been carrying for a long time by assigning student modelling a more prominent role in Rimac than has been done in most TDSs.

---

[1] See https://sites.google.com/site/rimacsite/ for a project overview and to download publications.

Table 1: A sample tutorial dialogue in Rimac

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

Reflection Question: 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?

S0: yes

T1 *(P1)*: Does the weight of the man in the elevator depend upon his motion at all or is weight completely independent of motion?

S1: independent of motion

T2 (*P2)*: That's correct. The weight of the man does *not* depend upon the motion of the elevator; it only depends upon his mass and the gravitational constant. This is evident from the equation that we use to calculate weight: $F_g = mg$. Neither the mass nor the gravitational constant depends upon motion; therefore, the weight will not.

It is well known that when riding in an elevator, most people will experience a mild shift in how heavy they *feel* at different times during the trip; sometimes you'll feel lighter and sometimes you'll feel heavier. So let me ask you this…What force are you actually perceiving whenever you feel heavier or lighter at different times during an elevator's motion?

S2: F_g

> **T3 *(S3)*:** I'm sorry, but that's not right. As strange as it may sound, it's actually the normal force that you are feeling. Let's see why this is true…If you step into an elevator in order to go up to a higher floor, at what point in the elevator`s motion would you expect to feel *heavier*?
>
> **S3**: Right after you begin moving.
>
> **T4 *(S4)***: That`s right. If you step into an elevator in order to go up to a higher floor, then at what point in the elevator`s motion would you expect to feel *lighter*?
>
> **S4**: Sometime in the middle of the trip.
>
> **T5 *(S4)***: That`s incorrect. We just established at what point in an elevator`s motion you will tend to feel heavier; now let`s again consider when you will feel *lighter.* At what point in the elevator's motion would you expect to feel lighter?
>
> **S5**: As the elevator is slowing to a stop.
>
> **T6 *(S5)***: I agree. Most people experience feeling *lighter at the end* when the elevator is coming to a stop. At the very beginning of the upward trip, how does the normal force compare to the weight?
>
> **S6:** The normal force and weight are the same size.
>
> **T7 *(S6)***: I disagree. When the elevator first starts moving, it is speeding upward. This means that the man has an *upward acceleration*. When the acceleration is upwards, it means the net force is upwards. Thus, **the normal force has to be bigger than the weight** in order for the net force to point upwards. At the very end of the upward trip, how does the normal force compare to the weight?
>
> **S7:** The normal force is smaller than the weight.
>
> **T8:** I agree. When the elevator's trip comes to an end the normal force on the man is smaller than his weight, because his acceleration is downwards. When the acceleration is downwards, it means the net force is downwards.

Recap: And so, it is the normal force from the ground that you are actually experiencing when you are aware of how heavy or light you are. For this reason, in situations like this, the normal force is often referred to as your *apparent weight.* This is why you are said to be experiencing "weightlessness" when in free fall. If the elevator`s cable snapped and you found yourself in free fall, you would feel "weightless" because there is no upward normal force acting on you. Therefore, your *apparent* weight would be zero despite the fact that your *actual* weight would still be the same as it`s always been.

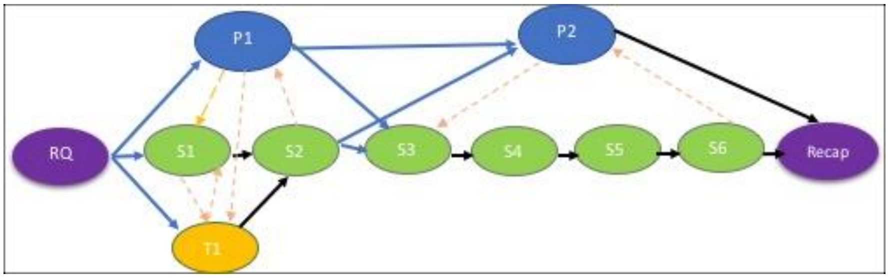
T=tutor; **S**=student; indentation shows shifts in granularity level (e.g., a shift from the primary line of reasoning to the secondary or tertiary level, as illustrated in the directed graph that generates alternate paths through the dialogue script, shown in Table 2). Italicized labels after tutor turn numbers (eg., *P1*, *S2*) correspond to nodes in the directed graph shown in Table 2.

Table 2: Directed Line of Reasoning that underlies the reflective dialogue shown in Table 1

Reflection Question: 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?

- **P1**: Does weight depend upon motion at all?
    - **S1:** What two factors directly affect weight?
        - **T1**: What is the equation that is used to determine the weight of an object?
    - **S2**: Is either the acceleration due to gravity or the mass affected by the elevator's motion?
- **P2:** What force are you perceiving whenever you are aware of how heavy you feel (such as in an elevator)?
    - **S3**: If you step into an elevator in order to go up to a higher floor, than at what point in the elevator's motion would you expect to feel *heavier*: the very beginning of the trip, the very end of the trip, or sometime in the middle?
    - **S4**: If you step into an elevator in order to go up to a higher floor, than at what point in the elevator's motion would you expect to feel *lighter*: the very beginning of the trip, the very end of the trip, or sometime in the middle?
    - **S5**: At the very beginning of the upward trip, how does the normal force compare to the weight?
    - **S6**: At the very end of the upward trip how does the normal force compare to the weight?
- **Recap**

Directed Graph: Possible paths through the dialogue



P = primary path; S = secondary path; T = tertiary path through the directed graph that corresponds to the dialogue's line of reasoning
Black arrows = required next step (e.g., after S4 go to S5)
Blue arrows = decision points (e.g., after initial RQ, go to P1 or S1 or T1)
Orange (dashed) arrows = remedial steps (e.g., P1 to S1 to T1 back to S1)

The remainder of this paper will proceed as follows. Section 2 illustrates the need to implement student modelling within tutorial dialogue systems. Section 3 describes Rimac and explains how the tutor uses a student model to drive adaptive scaffolding during dialogue. Section 4 summarizes classroom-based studies that addressed the following questions: (1) To what extent is the substantial effort required to incorporate student modelling in TDSs worthwhile, with respect to learning gains and efficiency? and (2) How important it is to dynamically update a student model to drive adaptive tutoring, as opposed to using a static student model that is initialized based on students' pretest performance, but not updated further (Albacete et al. 2017a; Jordan et al. 2017)? Section 5 discusses limitations of Rimac's student model and outlines future work to address these limitations. Section 6 describes related work on adaptive instruction in tutorial dialogue systems, and Section 7 focuses on one limitation of TDSs that needs to be addressed in order to promote sustained use of these systems in the classroom. We conclude by tying our work on incorporating student modelling in Rimac to Jim's vision for ITSs.

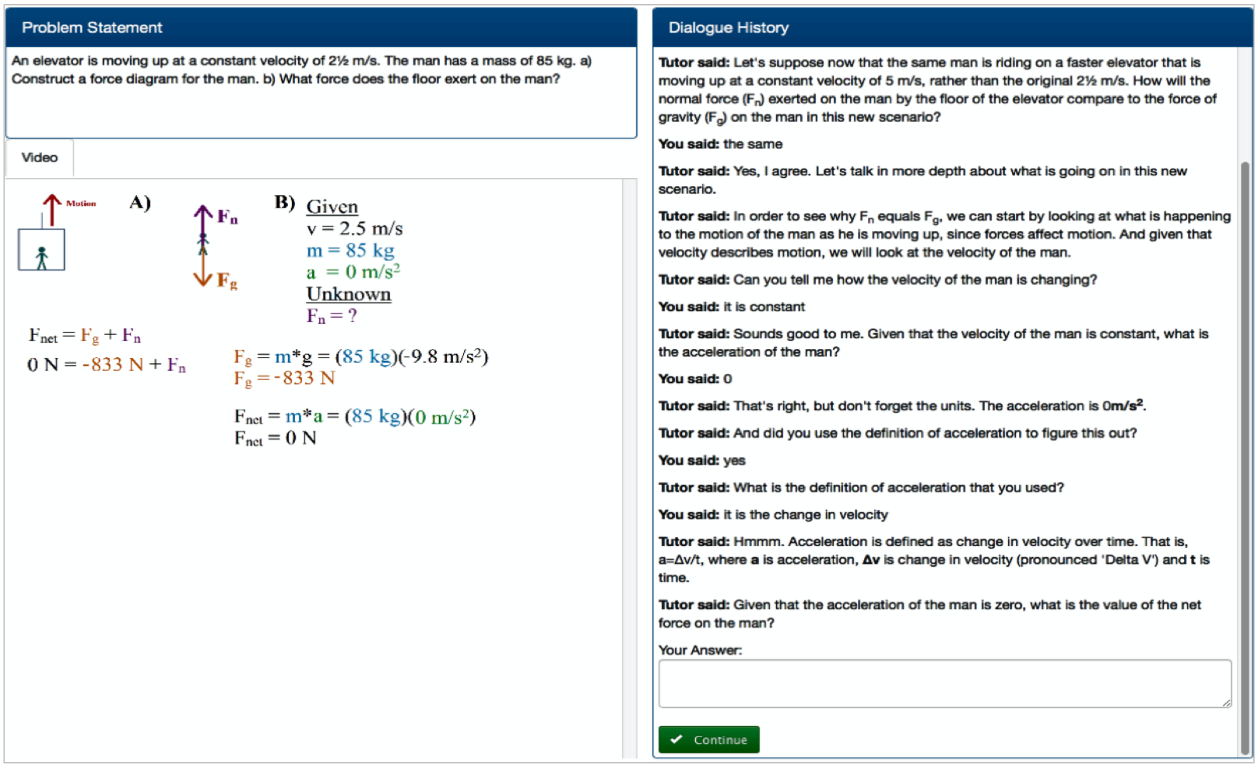
Table 3: The shortest path through the graph shown in Table 2
Reflection Question: When riding in an elevator, you may notice that you sometimes feel heavier or lighter than normal. At these times, are you actually heavier or lighter? In other words, when you feel heavier in an elevator, is your weight actually larger than when you are at rest on the ground?
S1: no, weight is the same
T1 (*P1*): That's correct. Does the weight of the man in the elevator depend upon his motion at all, or is weight completely independent of motion?
S2: weight completely independent of motion
T2 (*P2)*: That's correct. Your weight does *not* depend upon the motion of the elevator; it only depends upon your mass and the gravitational constant. This is evident from the equation that we use to calculate weight: $F_g = mg$. However, when riding in an elevator, most people will experience a mild shift in how heavy they *feel* at different times during the trip; sometimes you'll feel lighter and sometimes you'll feel heavier. What force are you actually perceiving whenever you feel heavier or lighter during an elevator's motion?
S3: the normal force
T3 *(Recap)*: I agree. And so, it is the normal force from the ground that you are actually experiencing when you are aware of how heavy or light you are. For this reason, in situations like this, the normal force is often referred to as your *apparent weight*. This is why you are said to be experiencing "weightlessness" when in free fall. If the elevator`s cable snapped and you found yourself in free fall, you would feel "weightless" because there is no upward normal force acting on you. Therefore, your *apparent* weight would be zero despite the fact that your *actual* weight would still be the same as it`s always been.
T=tutor; **S**=student; indentation shows shifts in granularity level (e.g., a shift from the primary line of reasoning to the secondary or tertiary level, as illustrated in the directed graph that generates alternate paths through the dialogue script, shown in Table 2). Italicized labels after tutor turn numbers (eg., *P1*, *S2*) correspond to nodes in the directed graph shown in Table 2.

2. Illustrating the Problem: The Limited Adaptivity of Tutorial Dialogue Systems

The amount of temporary support or "scaffolding" provided during human tutoring is contingent upon the learner's level of understanding or skill in carrying out a task (Belland 2014; van de Pol et al. 2010; Wood and Middleton 1975). For example, studies of parent-child interactions during problem-solving tasks (e.g., Pino-Pasternak et al. 2010; Pratt et al. 1992; Pratt et al. 1988; Pratt and Savoy-Levine 1998; van de Pol et al. 2010; van de Pol et al. 2015; Wood and Middleton 1975) have found that parents dynamically adjust the support that they provide to align with the child's zone of proximal development (ZPD), defined as "the conceptual space or zone between what a child is capable of doing on his or her own and what he or she can achieve with assistance from an adult or more capable peer" (Reber et al. 2009; Vygotsky 1978). The hallmark of effective scaffolding in these studies is a high frequency of adherence to the *Contingent Shift Principle* (Wood and Middleton 1975): "If the child succeeds, when next intervening offer less help. If the child fails, when next intervening take over more control" (p. 133). These studies, and related research on scaffolding in other contexts—for example, teachers' guidance of peer group interactions in the classroom (e.g., van de Pol et al. 2015)—indicate that tutorial dialogue systems should strive to emulate the contingent scaffolding of human tutoring.

Unlike human tutors, most tutorial dialogue systems tailor instruction to the student only to a limited extent. Many TDSs implement a popular framework called *Knowledge Construction Dialogues* (KCDs), which step all students through the same pre-scripted "directed line of reasoning" (DLR) (Hume et al. 1996), regardless of the student's ability in the targeted content. (Table 2 presents an example of a DLR.) Individualized instruction is limited to the tutor's deviations from the main path in the DLR, when the student answers incorrectly and the tutor launches a remedial sub-dialogue. Tutoring returns to the DLR's main path when a remediation has completed (e.g., Ai and Litman 2011; Chi et al. 2014; Evens and Michael 2006; Lane and VanLehn 2005; Litman and Forbes-Riley 2006; Rosé et al. 2006; Ward and Litman 2011). Similarly, dialogues in the AutoTutor family of TDSs follow an approach that is based on observations of human tutoring called *Expectation and Misconception-Tailored (EMT) dialogue* (e.g., Graesser et al. 2014b; Graesser et al. 2017a). Each dialogue sets an agenda of *expectations* (anticipated, correct responses) that must be covered at some point during the dialogue. The tutor remediates when a student's response reveals a *misconception* (flawed or missing knowledge).

Both of these approaches can cause frustration when students feel that the tutor is forcing them to engage in lengthy discussions about material that they already know and not addressing content that they need help with (e.g., Kopp et al. 2012; Jordan et al. 2018). Negative affective responses are worth heeding, in light of evidence that boredom and frustration with a tutoring system predict poor learning gains (Baker et al. 2010).



Problem Statement

An elevator is moving up at a constant velocity of 2½ 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?

Given
$v = 2.5$ m/s
$m = 85$ kg
$a = 0$ m/s^2
Unknown
$F_n = ?$

$F_{net} = F_g + F_n$
$0\ \text{N} = -833\ \text{N} + F_n$

$F_g = m*g = (85\ \text{kg})(-9.8\ \text{m/s}^2)$
$F_g = -833\ \text{N}$

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

Dialogue History

Tutor said: Let's suppose now that the same man is riding on a faster elevator that is moving up at a constant velocity of 5 m/s, rather than the original 2½ m/s. How will the normal force (F_n) exerted on the man by the floor of the elevator compare to the force of gravity (F_g) on the man in this new scenario?

You said: the same

Tutor said: Yes, I agree. Let's talk in more depth about what is going on in this new scenario.

Tutor said: In order to see why F_n equals F_g, we can start by looking at what is happening to the motion of the man as he is moving up, since forces affect motion. And given that velocity describes motion, we will look at the velocity of the man.

Tutor said: Can you tell me how the velocity of the man is changing?

You said: it is constant

Tutor said: Sounds good to me. Given that the velocity of the man is constant, what is the acceleration of the man?

You said: 0

Tutor said: That's right, but don't forget the units. The acceleration is 0**m/s^2**.

Tutor said: And did you use the definition of acceleration to figure this out?

You said: yes

Tutor said: What is the definition of acceleration that you used?

You said: it is the change in velocity

Tutor said: Hmmm. Acceleration is defined as change in velocity over time. That is, a=Δv/t, where **a** is acceleration, **Δv** is change in velocity (pronounced 'Delta V') and **t** is time.

Tutor said: Given that the acceleration of the man is zero, what is the value of the net force on the man?

Your Answer:

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

To illustrate the limited adaptivity of most TDSs, let's consider two cases that apply the KCD approach. The first case is illustrated in the dialogue shown in Table 1[2]. The dialogue script (DLR) shown at the top of Table 2 generated this dialogue. The DLR is also represented in the lower part of Table 2 as a directed graph that can produce alternate paths through the dialogue. Each node includes a question that the tutor asks the student as he or she progresses through the DLR. The graph shows that tutoring during this dialogue can take place at three levels of granularity. The primary (P) level requires the most amount of inferencing on the student's part (low granularity); the intermediate or secondary (S) level breaks down the steps between P-level nodes (moderate granularity); and the tertiary (T) level provides background knowledge needed to answer S and P level questions correctly (high granularity). The simulated student whose dialogue is included in Table 1 clearly needs scaffolding to understand the reasoning that leads to a correct response to the reflection question. This student answered the reflection question and the question in T2 (node P2) incorrectly, indicating that he or she does not understand that changes in the normal force govern a person's perception that their weight changes as they ride in an elevator—not their

---

[2] The dialogue in Table 1 was produced by one of the authors, to illustrate points discussed in this section. The tutor's and student's turns are shown verbatim, as they appear in the dialogue log.

actual weight, which is nearly constant. The remedial sub-dialogue provided by the DLR's secondary (S) path (turns T3-T8) therefore seems a fitting response to this student's apparently poor understanding.

Now consider the case of a more knowledgeable student who has a correct perception of the physical phenomenon addressed in this problem (i.e., the perception that one's weight changes with motion), understands that weight is independent of motion, and understands that a change in the *normal force* (not weight) is what one perceives while riding an elevator. This student should be allowed to move onto a more challenging problem, provided that he or she answers the main RQ correctly. However, in TDSs that implement the KCD approach—including Rimac before we incorporated student modelling—the shortest path that this student could take through this dialogue would be RQ→P1→P2→Recap, as shown in Table 3. Even a relatively short dialogue such as this would likely be a waste of this student's time.

Adherence to the standard KCD approach can also be insufficiently adaptive for less knowledgeable students. KCDs tacitly assume that traversing the primary path through a DLR, temporarily shifting to secondary and lower paths only when necessary for remediation, is appropriately challenging for all students. However, this is often not the case. The primary path through a dialogue script typically requires more background knowledge and reasoning ability than many students bring to a conceptual problem. A tutoring system with a student modelling component could use available data (e.g., the student's performance on a pretest, course exams, homework assignments) to predict that the student is not yet ready to answer particular P-level questions. The tutor could then immediately conduct the dialogue at a finer granularity level—at the S level and, as necessary, the T level, etc. In other words, adaptation could begin even before the tutor asks the first question in a dialogue script, based on the information represented in the student's student model. This more adaptive tutoring behavior would emulate what experienced human tutors do: ask questions that are within students' capability to answer correctly, look for cues that students need some support (e.g., a delayed response), and provide that support in order to avert an incorrect answer (Fox 1991).

The central hypothesis motivating our work is that tutorial dialogue systems would be more effective and efficient if they could consult a student model to provide individualized, knowledge-based instruction, as most effective problem-solving ITSs do. "Knowledge-based" means that information about student characteristics is represented in a student model, such as the student's understanding of curriculum elements (knowledge components), demographic information, affective traits such as interest in the subject matter, engagement, self-efficacy, etc. (Chi et al. 2011). The absence of such information about students forces designers of tutorial dialogue systems to make a "best guess" about how to structure a KCD—that is, what the main line of reasoning should be, what remedial or supplemental sub-dialogues to issue, and when—and then to hard code these guesses into dialogue scripts. The consequence is that students are often underexposed to material they don't understand and overexposed to material they firmly grasp. The first problem renders these systems ineffective while the second makes them inefficient, as shown in the examples discussed in this section.

Research by Kopp et al. (2012) indicates that a high dosage of interactive tutoring is not always necessary. The authors compared two versions of AutoTutor. The standard (control) version presented six conceptual questions to students in a research methods course. The dialogues associated with these questions addressed all expectations (targeted KCs), remediating when students responded incorrectly. An alternative, experimental version engaged students in dialogues about three conceptual questions and then presented three additional questions but did not engage students in dialogues. Instead, each conceptual question was followed by a canned response and explanation. The authors found that students in the experimental condition learned as much as students in the control condition, but they did so more efficiently (i.e., in less time). A follow-up study showed that it was just as effective to present the three highly interactive dialogues before the three static ("question + canned response") exercises as the reverse ordering.

These findings prompted us to design and develop a student modelling engine for Rimac that could increase learning efficiency in a more adaptive manner than the approach taken in Kopp et al.'s (2012) studies, arbitrarily alternating between intensive dialogue and no dialogue. Specifically, the student model would enable the tutor to choose which reflection questions to discuss further after a student responds to a reflection question correctly, and at what level of granularity.

3. Student Modelling in Rimac

3.1 Overview of Rimac

Students' failure to grasp basic scientific concepts and apply this knowledge to problem solving has been a persistent challenge, especially in physics education. Even students who are adept at solving quantitative problems often perform poorly on qualitative problems, and misconceptions can linger throughout introductory college-level physics courses (Halloun and Hestenes 1985; Mestre et al. 2009). Rimac provides a research platform to examine how to develop an adaptive tutoring system that can enhance students' conceptual knowledge about physics.

Rimac's conversations with the student are implemented as Knowledge Construction Dialogues (KCDs), as described in Section 2 and illustrated in Tables 1-3 and Fig. 1. Dialogues are authored using the TuTalk dialogue development toolkit, which allows domain experts to construct natural-language dialogues without programming (Jordan et al. 2006). Authors can focus instead on defining the tutorial content and its structure. This rule-based approach to dialogue, coupled with encouraging short answers at each student turn, increases input understanding and affords greater experimental control than do alternative approaches (e.g., Dzikovska et al. 2014).

Field trials conducted to test and refine Rimac typically involve having students complete an assignment during class or as homework. Since Rimac is not a problem-solving tutor, students solve quantitative physics problems on paper. They enter their answer to each problem in a box in the tutor's interface and then have the option to watch a video that presents a brief (~4-5 minute), narrated worked example of a correct solution. (See Fig. 1.) Because worked examples have consistently been shown to support learning (e.g., Atkinson et al. 2000; Cooper and Sweller 1987; McLaren et al. 2016; Sweller and Cooper 1985; van Gog et al. 2006), we use worked examples to provide students with feedback on problem solving. Rimac presents a series of conceptually focused reflection questions (RQs) about each just-solved problem, such as the RQs shown in Table 1 and Fig. 1. Students engage in a conversation about each RQ by typing responses to the tutor's questions. Although Rimac's dialogues supplement quantitative problems, the tutor's dialogues could alternatively be presented independent of problem-solving exercises, as a tool to enhance students' conceptual understanding, scientific reasoning and explanation skills.

Each step in Rimac's dialogues is associated with a set of learning objectives or knowledge components (KCs). For example, referring to the dialogue and DLR shown in Tables 1 and 2, respectively, the tutor's question in T1 (node P1) addresses the KC, *The weight (or gravitational force) of an object is the same regardless of whether or not the object is accelerating*; the tutor's question in T6 (node S5) addresses the KC, *For an object accelerating upward at a constant rate, the upward normal force must be larger than the downward gravitational force.* Rimac's student modelling component initializes its assessment of each KC that its dialogues address based on students' responses to pretest questions that target these KCs. It then uses students' responses to dialogue questions to dynamically update the student model's assessment of the KCs associated with each question, as described in the next section and in Chounta et al. (2017a). [See also Albacete et al. (2019).]

Rimac currently does not support student initiative through question asking. The main reason is that a repair mechanism would be necessary if student initiative were encouraged, to enable the system to develop a shared understanding of the student's initiative, which may include topic shifts. In our opinion, the technology for handling conversational repair and understanding needs to improve dramatically before student initiative can be supported. We consider this an important goal for future research, in light of abundant research that demonstrates the instructional benefit of question asking (e.g., Gavelek and Raphael 1985; King 1990, 1994; Palincsar 1998) and the substantially higher frequency of student questions asked during one-on-one human tutoring than in the classroom (Graesser and Person 1994).

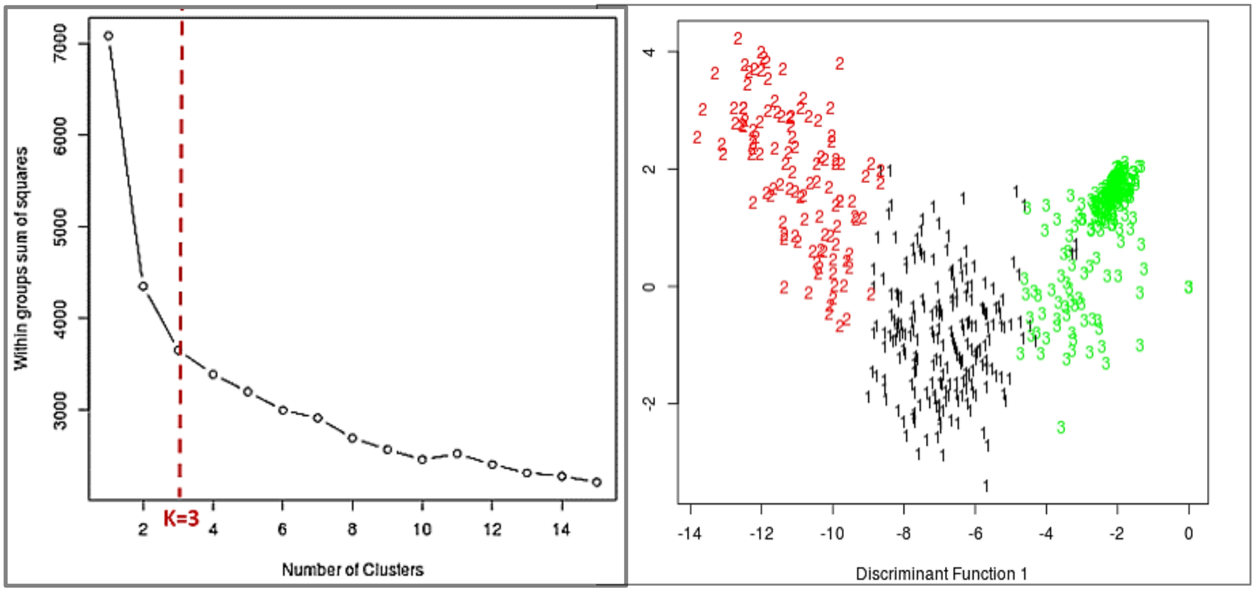


Fig. 2 The within-groups sum of squares by number of clusters, used to determine the optimal number of clusters (left), and the results of the K-means clustering process (right)

3.2 How Rimac Produces a Student Model and Uses it to Guide Adaptive Tutoring

3.2.1 Overview

Rimac's student model enables the tutor to emulate the contingent (adaptive) scaffolding provided by human tutors. Specifically, Rimac implements *domain contingency* and *instructional contingency* (Katz et al. 2018; van de Pol et al. 2010; Wood 2001, 2003). Domain contingency entails selecting appropriate content to focus on while a student performs a task, while instructional contingency entails addressing this content with an appropriate amount of support.

To achieve domain contingency in Rimac, we developed different versions of each dialogue, each version corresponding to a line of reasoning (LOR) at a different level of granularity. When embedded in dialogues, these alternative LORs can be represented as a directed graph, as shown in Table 2 (Albacete et al. 2019). To achieve instructional contingency*,* we developed different versions of the questions, hints, feedback, and explanations that are associated with each step of a DLR. Each version provides a different level of support (i.e., low, medium, or high support), chosen based on the student model's assessment of how likely the student is to correctly answer the question asked at a dialogue step. For example, consider these two versions of the same core question: "What is the man's acceleration?" versus "Given that the net force applied on the man is 2N, and thinking about Newton's Second Law, what is the man's acceleration?" The latter question provides more support than the former because it reminds the student what the value of the net force is and that net force and acceleration are mathematically related (Katz et al. 2018).

Rimac always aims for mastery by selecting the question at the lowest possible level of granularity that the student is likely to answer correctly—that is, the question bearing the least discussion about relevant background knowledge. The tutor consults the student model to make this choice. Rimac represents a student model as a regression equation, implemented using Instructional Factor Analysis Model (IFM), as proposed by Chi et al. (2011). The system starts with a default stereotypical model for low, medium, and high prior knowledge students, classifies the student into one of these groups based on the student's overall pretest score, and then adjusts the default model based on the student's responses to pretest items and dialogue questions. This section describes the process of initializing and updating a student model in Rimac and using the model to guide individualized instruction. [For more detail, see also (Albacete et al. 2018), Chounta et al. (2017a), and Katz et al. (2018).]

3.2.2 Initializing the student model and customizing it for a particular student

We used stereotyping (Tsiriga and Virvou 2002) to initialize Rimac's student model. Training data enabled us to specify three student "personas" that the system could use to classify students based on their pretest performance: low, medium, and high prior knowledge students. The training data consisted of 560 students' pretests and dialogues over a four-year period (2011-2015). We used K-Means to cluster students in the training data set based on their pretest performance because we had previously found that students' average pretest scores correlated positively with the student model's predictions (Chounta et al. 2017a). Also, pretest data (and/or experts' assessment data) is typically used to initialize student models because it enables initialization of KC-specific parameters, such as difficulty level and prior knowledge (Gong 2014).

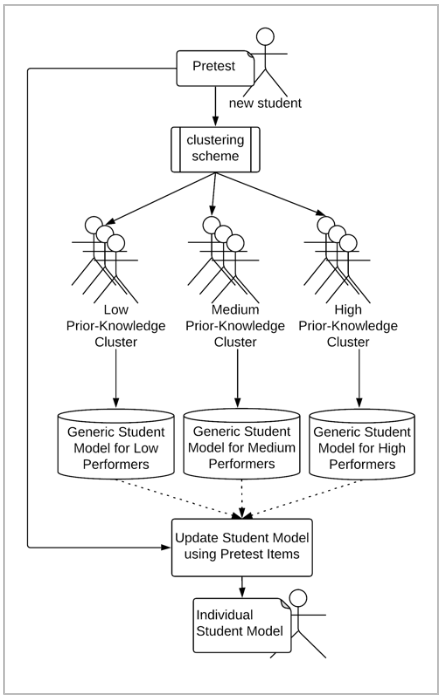

Pretest
new student
clustering scheme
Low Prior-Knowledge Cluster
Medium Prior-Knowledge Cluster
High Prior-Knowledge Cluster
Generic Student Model for Low Performers
Generic Student Model for Medium Performers
Generic Student Model for High Performers
Update Student Model using Pretest Items
Individual Student Model

Fig. 3 The student model initialization process for a new student

The appropriate number of clusters in K-means is typically determined by plotting the within-groups sum of squares by number of clusters and choosing the point where the plot bends (Hothorn and Everitt 2014). The sum of squares plot shown at the left side of Fig. 2 indicated that three was the appropriate number of clusters for the training data. We validated that these clusters represent groups of "high," "medium," and "low" prior knowledge students based on students' average pretest scores (Fig. 2, right). We then trained an instance of the student model for each persona (cluster). This yielded a better prediction accuracy score than did using the whole training data set, so we decided to use the three generic student models that resulted from this training step, as illustrated in Fig. 3: the Low Prior Knowledge persona, the Medium Prior Knowledge persona, and the High Prior Knowledge persona.

Each student who uses the tutor for the first time takes an online pretest, which enables the system to classify the student according to one of these personas. Rimac initializes a student's model by using the generic student model that coincides with this persona. It then personalizes the student model by analyzing the student's responses to test items that target particular knowledge components. We developed a program called the *Multiple-choice Knowledge Assessment Tool* (McKnowAT) to automatically assess students' understanding of the KCs associated with a given multiple choice test item, based on the student's selection (or non-selection) of the item's options (Albacete et al. 2017b).

3.2.3 Updating the student model

As the student progresses through a dialogue, the student model is dynamically updated based on the student's responses to the tutor's questions. Each dialogue exchange (pair of tutor-student dialogue turns) is treated as a training instance, represented by the KCs that the exchange addresses and the status of the student's response to the tutor's question (i.e., correct or incorrect). For example, in the dialogue excerpt shown in Table 1, the student answers the question asked in turn T2 incorrectly. This turn maps to primary path node P2 ("What force are you actually perceiving whenever you feel heavier or lighter during an elevator's motion?"). The main KC associated with this node is: *When a person is aware of how heavy they feel, they are perceiving the magnitude of the normal force acting on them, not their actual weight.* Consequently, the student model will downgrade its assessment of this KC, expressed as the probability that the student understands it.

The student modelling system updates the student model after every exchange. In this way, the model maintains the most current image of the student's knowledge state. In addition to being kept up to date, each student model is persistent—that is, carried over from one problem and dialogue to the next, one assignment to the next. In contrast, most TDSs that implement some form of student modelling create the student model anew during each dialogue.

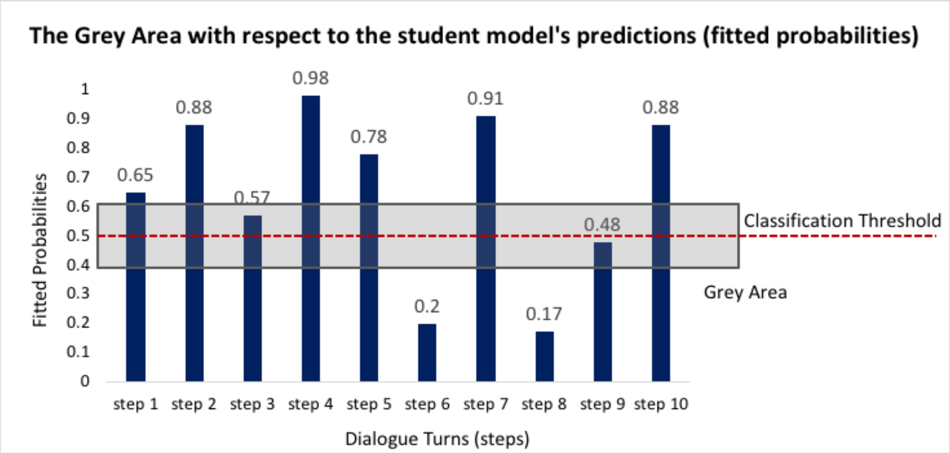


Fig. 4 The Grey Area construct with respect to fitted probabilities (i.e., the probability that a student will answer a particular step correctly), as predicted by the student model for a random student and for the steps in a conceptual problem-solving task (RQ)

3.2.4 Choosing the next question to ask during a dialogue

Few students follow exactly the same path through one of Rimac's dialogue scripts, such as the DLR shown in Table 2. Some transitions from one question to the next are hard coded into the dialogue script; that is, no decision needs to be made about which question to ask next. (These predefined transitions are indicated by black arrows in Table 2.) Predefined transitions help to keep dialogue length manageable and to reduce cognitive load. They often correspond to knowledge that was discussed in a previous RQ and summarize those KCs so that the student can focus on the goal(s) of the current RQ. To illustrate, this is the case with nodes S5 and S6 shown in Table 2. These nodes address KCs that were discussed in detail in a previous RQ (i.e., the relationship between normal force and weight in an accelerating or decelerating elevator). Hence, when a student answers the question asked at S5 ("At the very beginning of the upward trip, how does the normal force compare to the weight?") incorrectly, the tutor remediates by summarizing these previously discussed KCs before proceeding to the next question asked at S6 ("At the very end of the trip, how does the normal force compare to the weight?"), as shown in tutor turn T7 of Table 1. Predefined transitions also support dialogue coherency, because some questions need to be asked in sequence, as is the case with nodes S3 and S4, and S5 and S6. In addition, some predefined transitions reflect “bottom out” remediations; the tutor provides the correct answer and there is no lower level line of reasoning to discuss.

Other nodes in a dialogue’s DLR raise a decision to be made about which node to traverse to next. These decisions are based on the student model’s predictions about the likelihood that the student will answer the question asked at candidate “next step” nodes correctly. For example, if a student answers the RQ correctly, the tutor must first decide if the student needs to discuss this problem further, based on the student model’s assessment of the learner’s understanding of the most relevant KCs associated with this RQ, as discussed in Jordan et al. (2016) and Albacete et al. (2019). If so, the tutor’s next decision is to choose an appropriate level at which to conduct this discussion: Should the discussion start at the primary, secondary, or tertiary level? (Transition choice points are indicated by blue arrows in Table 2.)

When making these decisions, the tutor strives to balance challenge with potential success. Rimac uses logistic regression to predict the probability of a student answering the question asked at each candidate node correctly as a linear function of the student’s proficiency on the associated KCs. The tutor then uses the classification threshold to interpret the meaning of this probability. The “classification threshold” is the probability that determines which class will be chosen, such as correct vs. incorrect. It is shown as a dotted line in Fig. 4. In Rimac, this threshold was determined to be 0.5 (Chounta et al. 2017a). Hence, if the probability of a student answering a question is >= 0.5, it is interpreted as “the student is likely to answer the question correctly;” otherwise, the probability is interpreted as “the student is likely to answer the question incorrectly.”

As a prediction gets closer to 0.5, there is increasing uncertainty: ultimately, a 50% chance that the student will answer the question correctly and a 50% chance that the student will answer it incorrectly. Consequently, this probability cannot be interpreted reliably. We refer to the region of high uncertainty between 0.4 and 0.6 as the “Grey Area” (Chounta et al. 2017a, 2017b), as shown in Fig. 4. The student model’s high degree of uncertainty in its predictions within the Grey Area might simply be due to insufficient evidence; that is, the system has not yet accrued enough data to assess the student’s understanding of the material that a question addresses. Alternatively, high uncertainty might reflect the student’s unsteady command over this material. Perhaps the student is on the brink of understanding but has not yet sufficiently mastered the KCs associated with a question—as evident, for example, in the student’s inconsistent responses to items on the pretest and in previous questions that address these KCs. With these considerations in mind, we proposed that predictions in the Grey Area might indicate that the student is in the ZPD for the targeted material. Correspondingly, the Grey Area might afford a computational model of the ZPD (Chounta et al. 2017a, 2017b).

Once the system has interpreted the student model’s output (i.e., its predictions of success at each candidate “next node”), the tutor chooses the question at the lowest level of granularity whose predicted probability of a correct response is at least 0.4—in other words, within or above the Grey Area. The student is expected to either be able to answer this question correctly (i.e., the probability of a correct response is >= 0.6) or to be within his or her ZPD for that question (i.e., the probability of a correct response is 0.4-0.6). The student model examines each possible next question, starting with the P-level and moving down. It checks the probability of each of these questions in sequence (P, then S, then T level, and further down if possible) until it finds a question that it can ask, according to the student model’s selection policies. For example, again referring to the DLR shown in Table 2, if the student answers the top-level RQ correctly and the tutor’s prediction of a correct response is 0.6 for nodes P1 and S1, the tutor will ask the question at P1 because this question is more challenging than the one at S1 (i.e., the question at P1 requires more inferencing). If the last question asked was at P1, the probability of a correct response at P2 is 0.4, and the probability of a correct response at S3 is 0.75, the tutor will traverse from P1 to S3—again with the aim of balancing challenge and potential success. P2, the expert-level question, will not be asked because its probability of being answered correctly is below 0.6. S3 will be asked instead, since its probability of a correct response is at least 0.4.

By taking this approach to choosing the next question, Rimac treats the uncertainty of student modelling as a feature, not as a bug. It uses the model's estimates of how likely a student will succeed at the next step in a DLR to choose an appropriate level at which to address this step. Few predictions will approximate 1.0—that is, complete confidence that the student will answer the question asked at a candidate "next step" node correctly, as illustrated in Fig. 4. For example, slips sometimes happen, even when students have mastered the KCs associated with a question. The student model is nonetheless robust enough to adaptively choose the next step and to ask the question at an appropriate level of support, as described next.

3.2.5 Adapting the level of support provided at each dialogue step

The Contingent Shift Principle that underlies effective scaffolding during human tutoring (e.g., Pratt and Savoy-Levine 1998; Wood 2001; Wood et al. 1976) is partly realized in Rimac when the tutor selects the next step in a dialogue script. The tutor's predictions about the student's probability of answering each candidate "next step" correctly allow it to determine whether to ask a question at the P level, S level, T level, etc., which often results in shifts across levels—for example, asking a question that requires a high amount of inferencing on the student's part at one step (e.g., a P-level question), then shifting to a question that requires a moderate amount of inferencing at the next step (e.g., an S-level question).

Rimac's job is not done when it adaptively chooses the next step in a line of reasoning, simulating the domain contingency observed in effective human tutoring and teaching (Brownfield 2016; Brownfield and Wilkinson 2018; Rodgers et al. 2016; van de Pol et al. 2010; Wood 2001). The tutor must also simulate instructional contingency by deciding how to ask the question at this step and how to implement other tutoring strategies (e.g., hints, explanations, feedback). As is the case with choosing the next step, adjusting the support provided in a question, hint, etc. is tantamount to emulating the Contingent Shift Principle, because the level of support that the tutor provides at one step might shift up or down at the next step, based on the tutor's assessment of the student's need for support in order to answer the question at these two steps correctly. For example, if the tutor decides to ask a question at the P-level, the student will always receive low support, because the student model is fairly certain that the student will answer this question correctly (i.e., probability >= 0.6). Conversely, if the probability of a correct response to a question is < 0.4—that is, below the Grey Area—the tutor is fairly certain that the student will answer incorrectly and will therefore provide ample support. Since there is less certainty within the Grey Area (0.4-0.6) regarding how much support the student will need to answer the question correctly, we divide this region into three segments: lower third (0.40-0.45), middle third (0.46-0.52), and upper third (>= 0.53) corresponding to high, medium, and low support, respectively. This policy for choosing how much support to provide when the probability of success lies within the Grey Area pertains only to S-level and T-level questions. P-level questions are always asked with low support.

Rimac's dialogue authors produce several variations of questions, feedback on students' responses, hints, explanations, and other tutoring strategies for each dialogue step. These variations allow the tutor to choose an appropriate level of support (low, medium, or high), given the predicted probability that the student will answer the question at a given step correctly. We specified guidelines for dialogue authors to use to generate alternative forms of questions, hints, etc. and decision rules to guide the tutor in choosing among these alternatives. These authoring guidelines and decision rules operationalize what it means to provide a "high" level of support, versus a "moderate" level, versus a "low" level when implementing tutoring strategies (Katz et al. 2018).

Fortunately, several scaffolding researchers and developers of teacher professional development programs have also faced the challenging task of defining different levels of support (LOS)—for example, in order to measure the frequency of contingent shifts observed in various instructional contexts (e.g., Pratt et al. 1992; Pratt and Savoy-Levine 1998; van de Pol et al. 2014; Wood et al. 1978). This work has yielded many frameworks that specify different levels of teacher support, or different levels of cognitive complexity, depending on whether a framework differentiates levels from the teacher's or student's perspective, respectively. Table 4 provides an example of a framework that distinguishes levels of support according to how much control the teacher exerts during small group problem-solving tasks in the classroom (van de Pol et al. 2014). This framework has been used or adapted in several teacher training programs and scaffolding studies (Brownfield 2016; Brownfield and Wilkinson 2018; Rodgers 2017; Rodgers et al. 2016; San Martin 2018; van de Pol et al. 2019; van de Pol et al. 2014, 2015). However, the descriptions of each level of support (control) provided in this and most frameworks are not specific enough to develop authoring guidelines to generate alternative questions, hints, etc., and decision rules that an automated tutor can consult to choose among these alternatives. For example, referring to the sample LOS framework in Table 4, it is unclear how to produce a "broad and open question" at Level 1, versus a "more detailed but still open" question at Level 2, versus "a hint or suggestive question" at Level 4.

Table 4: Sample Level of Support framework
Adapted mostly verbatim from van de Pol et al. (2014, p. 649)

TDc0 *No control/no support.* Teacher absent

TDc1 *Lowest control—The teacher:*
- Provides no new content
- Elicits an elaborate response
- Asks a broad and open question

TDc2 *Low control—The teacher:*
- Provides no new content
- Elicits an elaborate response, mostly for an elaboration or explanation of something ("why" questions)
- Asks a more detailed but still open question

TDc3 *Medium control—The teacher:*
- Provides new content
- Elicits a short response (yes/no or choice)

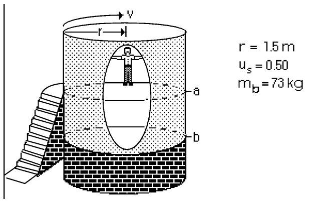
TDc4 *High control—The teacher:*
- Provides new content
- Elicits a response
- Gives a hint or suggestive question

TDc5 *Highest control—The teacher:*
- Provides new content
- Elicits *no* response
- Gives an explanation or the answer to a question

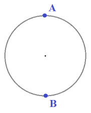
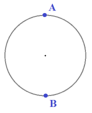
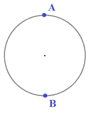
TDc = Teacher's degree of control. Support increases from lowest to highest level (TDc1 – TDc5).

Table 5: Rules for generating and selecting questions at different levels of support

Problem statement:
A popular amusement park ride (shown in the image on your paper) operates as follows: riders enter the cylindrical structure when it is stationary with the floor at the point 'a'. They then stand against the wall as the cylinder begins to rotate. When it is up to speed, the floor is lowered to the position marked 'b', leaving the riders 'suspended' against the wall high above the floor. Draw a force diagram for the rider after the floor has dropped out. Include ALL forces acting on the rider.



Reflection question:
In the example of the amusement park ride, we were told that the rider is travelling along a circular path at a *constant speed*. Since the rider is not speeding up or slowing down, can he still possibly be experiencing an acceleration?

High level of support (low probability of correct response)	***Medium level of support*** (medium probability of correct response)	***Low level of support*** (high probability of correct response)
If the probability that the student will answer the question correctly is low, choose a question with a high level of support, generated according to the following factors: ● *Factor A*: All physical quantities refer to the context of the problem. ● *Factor B:* Include a hint/support ● *Factor C:* Explicitly state results of previous step	If the probability that the student will answer the question correctly is low, choose a question with a medium level of support, generated according to the following factors: ● *Factor A*: Mention the context of the problem only once. ● *Factor B:* Include a hint/support ● *Factor C:* Refer to the results of previous step in a general way	If the probability that the student will answer the question correctly is high, choose a question with a low level of support, generated according to the following factors: ● *Factor A*: Refer to physical quantities in an abstract way. ● *Factor B:* Do not include a hint/support ● *Factor C:* Not applicable.
 **Tutor-1:** *Knowing that **at location A the rider's velocity vector points directly to the left**,* let me ask about the "bottom" of the circle. If the rider is traveling counterclockwise on a circular path, which way does his velocity vector point at the 'bottom' of the circle (at location B in the figure at right)?	 **Tutor-1:** *Knowing the **direction of the velocity vector at location A**,* let me ask about the "bottom" of the circle. If the rider is traveling counterclockwise on a circular path, which way does his velocity vector point at the "bottom" of the circle (at location B in the figure at right)?	 **Tutor-1:** If the rider is traveling counter-clockwise on a circular path, which way does the velocity vector point at the "bottom" of the circle (at location B in the figure at right)?

Factor A: **whether to refer to physical quantities as stated in the problem** (boldface)
Factor B: whether to provide a hint/other type of support (underlined)
Factor C: whether to refer to the results of a previous step, and how explicitly (italics)

In order to more precisely operationalize "different levels of support," we examined level definitions across several LOS frameworks, and transcripts from human physics tutoring sessions, with the aim of identifying factors that dialogue authors could use to adjust the level of support afforded by alternative forms of questions, hints, etc. (Katz et al. 2018). For example, some factors that render a question more (or less) abstract include:

- Does the question refer to objects included in the problem statement (e.g., "What is the velocity of the bicycle?" vs. "What is the velocity?")?
- Does the question provide a hint and, if so, what type of hint—one that states a piece of information needed to answer the question correctly, or one that prompts the student to recall this information on his or her own—that is, a *convey information hint* versus a *point to information hint*, respectively (Hume et al. 1996)?

- How much and what type of information should be included in the question—for example, a reference to the answer to the previous question; a list of possible answers to choose from [e.g., "how is the velocity varying (increasing, decreasing, constant, etc.)?" versus "how is the velocity varying?"]?
- In a quantitative domain such as physics, does the question refer to a law or principle in equation form or sentence form—for example, "$a = \Delta v/\Delta t$" vs. "acceleration is defined as the change in velocity over the time interval"?

These factors were incorporated in authoring guidelines and decision rules for questions, as illustrated in Table 5.

4 Pilot Studies to Assess the Student Model Driven Version of Rimac

We conducted two classroom-based studies to gain an initial sense of whether incorporating student modelling within tutorial dialogue systems is worth the effort, as measured by learning gains, more efficient learning, or both. In both studies, we found an advantage for efficiency, measured by time on task, but not for effectiveness, measured by pretest to posttest gain scores. We summarize these studies in this section. [See Jordan et al. (2017) for more detail on Study 1 and Albacete et al. (2019), for more detail on Study 2.]

4.1 Study 1

Does a TDS that uses a "poor man's student model" to decide when to decompose steps into lower-level sub-steps promote more efficient learning than a TDS that always decomposes steps into sub-steps?

We predicted that the answer to this question would be yes, because micro-level tutoring at the sub-step level can be time-consuming. To test our hypothesis, we compared two versions of Rimac. Dialogues in the control version behave as dialogues developed using the KCD framework typically do; they take a cautious approach to instruction. That is, even when a student answers the tutor's question at a given step during the dialogue correctly, the tutor will address the step's sub-steps, in case the student's correct response stemmed from a lucky guess or from incomplete reasoning that nonetheless was "good enough." For example, referring to the DLR shown in Table 2, a student in the control group would not be allowed to skip the dialogue associated with this RQ if he or she were to answer this RQ correctly. Similarly, if the student were to answer a P-level question correctly, the student would still be required to traverse this step's associated S- and T-level nodes. For example, after the student answers the question at P1 correctly, the tutor would choose the following path leading to P2: S1→T1→S2→P2.

In contrast, dialogues in the experimental version of Rimac use students' pretest performance to decide whether to decompose a step into its corresponding sub-steps, provided that the step is decomposable. The student model bases this decision on the predicted probability that the student already understands the KCs associated with these sub-steps, measured as 0.8 or above for the top-level RQ and 0.5 or above for decomposable steps in the DLR. For example, if the student model does not allow the student to skip the DLR in Table 2 because the predicted probability that the student understands the main concepts associated with this RQ correctly is less than 0.8, the student could still proceed through a relatively short path (e.g., RQ→P1→P2→Recap), as long as the predicted probabilities of success at P1 and P2 are above 0.5.

Data from 72 students enrolled in physics classes at 3 high schools were included in the study (N=35 Control; N=37 Experimental). Students in both conditions took an online pretest in class. They then solved two quantitative problems on paper. For each problem, students had the option to watch a worked-example video in the tutor that provided feedback on problem solving but no conceptual information.

After each problem, they engaged in dialogues that addressed the concepts associated with the just-solved problem (five RQs total, across the two problems). Finally, students took an online post-test in class and completed a user satisfaction survey.

Data analysis revealed significant pretest to posttest gain scores in both conditions. However, neither condition was more beneficial for learning than the other and no aptitude-treatment interactions were observed. Both high and low prior knowledge students learned a comparable amount, as measured by pretest to post-test gain score, regardless of which condition they were assigned to. The only significant difference between conditions was an efficiency advantage for high prior knowledge students in the experimental condition. The mean time for this group of students to complete the dialogue intervention was 34 minutes in the experimental condition versus 46 minutes in the control condition. More importantly, this gain in efficiency did not come at a cost to learning. High prior knowledge students across conditions showed comparable gain scores.

4.2 Study 2

Does a TDS whose student model is dynamically updated at each dialogue step promote higher learning and efficiency gains than a TDS that uses a static student model?

This follow-up to Study 1 compared two student model guided versions of Rimac. The control version was similar to the experimental version in the first study. The student model was initialized using students' pretest scores and not updated further. As in Study 1, students could skip the discussion and move on to the next RQ if they answered the current RQ correctly and their pretest scores on all of the most relevant KCs were 0.8 or above. Otherwise, if the student could not skip the RQ, he or she would be assigned to a fixed path through the line of reasoning at the expert level (P) if the student's scores on all relevant KC's were greater than 0.7; a fixed path at a medium level (S) if the student's scores on all relevant KCs were greater than 0.4; and a fixed path at a novice level (T or lower) otherwise. Hence, the lowest scoring KC in the set drove decision making. We refer to this condition informally as the "poor man's student model" because it lacks the complex mechanisms needed to dynamically update the student model and use the model to select steps, as described in Section 3.2.

In the experimental version of the tutor, the student model's assessment for each KC was updated dynamically after each dialogue step. As described in Section 3.2, the tutor referred to the predicted probability of a correct response at each candidate "next step" to select a dialogue step. It aimed for mastery by selecting the step with the highest probability of a correct response, within the path through the DLR that requires the most amount of inferencing. We predicted that the experimental version of the tutor would outperform the control version on learning and efficiency gains. As with the first study, only the latter prediction was realized. However, this time the experimental version showed an efficiency advantage for both high and low prior knowledge students, classified according to a medium split of students' pretest scores.

Data from 73 students enrolled in physics classes at one high school were included in the study (N=42 Control; N=31 Experimental). The study protocol was similar to that followed in the first study, except that students solved a few more problems and engaged in more dialogues with the tutor: 5 quantitative problems with 3-5 reflection questions per problem.

As with the first study, we found significant pretest to posttest gains across conditions, but neither condition was more beneficial with respect to learning gains than the other. This might be due to the thoroughness of the control condition. If a student were to answer a question incorrectly, he or she would go through a remedial sub-dialogue that explicitly addresses all of the material that the path the student had been assigned to (e.g., P or S) expects the student to infer.

The lack of higher learning gains for the experimental condition might also be due to the shortness of the intervention. Perhaps the instructional benefit of dynamic updating only manifests in sufficiently long interventions. Consequently, when we add more content (problems and dialogues) to the tutor, we might observe greater learning gains. This content will include challenging problems that allow high incoming knowledge students to learn new material and better understand concepts that they have not fully mastered, as well as problems that give low-performing students more practice in acquiring and applying basic concepts.

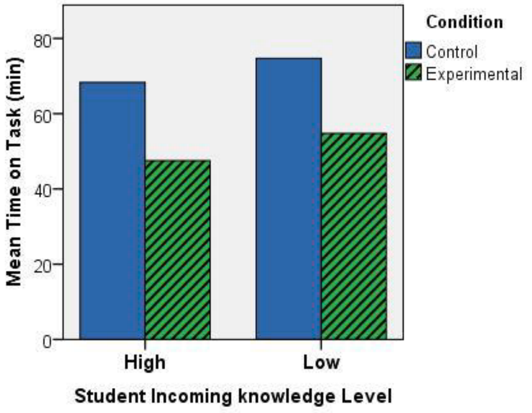


Fig. 5 Comparing time on task between conditions for low and high prior knowledge students

As in Study 1, no aptitude-treatment interactions were observed. However, in Study 2 both high and low prior knowledge students learned more efficiently (i.e., took less time on task) in the experimental condition than in the control condition. Specifically, students in the dynamic student modelling group went through the dialogues about 27% faster than students in the static student modelling group, as illustrated in Fig. 5. Increased efficiency is an important outcome because it indicates that the tutor focuses on material that students need help with and doesn't spend too much time on material that students have sufficiently mastered. This leaves time for more challenging tasks and, perhaps, helps to sustain student interest.

4.3 Assessing the Student Model's Performance

The experimental condition from Pilot Study 2 provided data that we could use to assess the accuracy of the student model's predictions. The 31 participants in the experimental condition answered a total of 2,603 questions. Approximately 94% of students' responses to these questions (2,436) were predicted to lie outside of the Grey Area; that is, the probability of a student answering a question correctly was either below 0.4 or above 0.6. The remaining 6% of responses (167) were predicted to lie within the Grey Area; that is, the probability of a student answering a question correctly was between 0.4 and 0.6, thereby with dubious interpretability. The student model's accuracy outside of the Grey Area was 0.61; in other words, 61% of the model's predictions were correct. As expected, the student model's accuracy within the Grey Area, 55%, was lower than it was outside the Grey Area.

The confusion matrix of the student model's predictions when used as a binary classifier (correct vs. incorrect response), with a classification threshold set to 0.5, is shown in Table 6, where the positive class of "1" signifies correctness. The model's precision (specificity) and recall (sensitivity) were 0.94 and 0.62, respectively. Hence, with respect to precision, 94% of the model's predictions that a student's response would be correct were in reality correct. With respect to recall, 64% of correct responses were predicted as such, although the model mistakenly classified many responses as correct that were in fact incorrect (97%).

This high precision and low recall indicate that the model's classifier is extremely "picky" when it comes to detecting incorrect answers. In other words, our model is able to predict correct answers with better accuracy than incorrect answers. Indeed, only 3% of incorrect responses were predicted as such. This skew towards predicting correct responses may indicate overfitting. This is a plausible explanation because the dataset was imbalanced: 63% of students' responses (1,543) were correct.

Another possible explanation for the student model's poor performance in predicting incorrect responses is that the classification threshold (0.5; see Fig. 4) might not be appropriate for this dataset. This threshold was set using a training dataset. However, the Receiver Operating Characteristic (ROC) curve for the binary classifier shown in Fig. 6 indicates a higher optimal classification threshold for the pilot study dataset, 0.8l; that is, 0.8 is the point where the classifier can achieve the maximum precision (specificity) and recall (sensitivity). A possible explanation for this discrepancy in classification thresholds is that students who participated in the pilot study were more knowledgeable than students whose data comprise the training dataset. This would also account for the imbalanced pilot study dataset, which favors correct responses.

Table 6: Confusion matrix of the student model when used as a binary classifier

n = 2436	**Actual Correct (questions that in fact student answered correctly)**	**Actual Incorrect (questions that in fact student answered incorrectly)**
Predicted Correct	1453 (94%)	866 (97%)
Predicted Incorrect	90 (6%)	27 (3%)
Total	1543	893

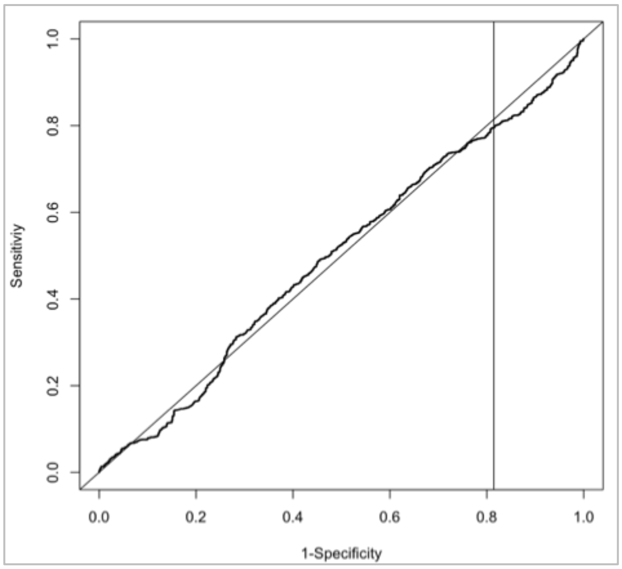


Fig. 6 ROC curve of the binary classifier as used in the pilot study

5. Future Work to Improve the Student Model's Performance

The accuracy of the student model's predictions outside of the Grey Area (61%) indicates that the model's performance is heading in the right direction but there is considerable room for improvement. We have two strands of modifications planned: changes that directly address how the student model is implemented and changes that are external to the student model but could impact its performance—for example, possible improvements in knowledge representation and in the tutor's natural-language processing capability. This section provides an overview of planned work in each strand.

5.1 Planned Enhancements to Rimac's Student Modelling Component

In Section 4.3, we noted that differences between the training dataset and the actual dataset may have led to inaccurate specification of the classification threshold used in the pilot study. The classification threshold also impacts the model's Grey Area boundaries (i.e., its upper and lower limits). We plan to further examine the relationship between the classification threshold and student ability, as measured by pretest performance. If we find that students' prior knowledge has as strong an impact on where the optimal classification threshold lies as we suspect, we will routinely use students' pretest scores to customize the classification threshold and Grey Area boundaries for each student, thereby potentially increasing the student model's ability to accommodate students' needs.

We also plan to augment the student model with a "slip" parameter in order to handle situations where a student may provide a wrong answer despite their having the knowledge needed to answer correctly. This feature is fundamental to other student modelling approaches, such as Bayesian Knowledge Tracing, but it is not commonly implemented in logistic regression student models, such as the IFM that we use. Nonetheless, related research has shown that "slips" can be sufficiently modelled, and that doing so improved the predictive accuracy of the student model in problem-solving tutoring systems (MacLellan et al. 2015). We expect that a slip parameter will be especially important to have in place when Rimac has enough content to support longer interventions, with greater risk that students will forget material that they previously knew.

Additionally, there are different ways to represent the data used to train and update the model that could influence student model accuracy. In Rimac, the data used to train students' initial student models consist of information about the question-response pairs students experience during their interactions with Rimac. The types of questions students answer vary as to how much support they provide the student. However, the biggest consistent differences in the amount of support provided are between the initial reflection questions, which are asked in the same way to all students, and the questions within the dialogues that follow up on the reflection question, which provide adaptive levels of support, as described in Section 3.2.5. We plan to test whether distinguishing between reflection questions and dialogue questions during model training and updating improves student model accuracy. Similarly, we plan to test whether including pretest questions in the training data could influence the accuracy of the student model. Some pretest questions are similar to reflection questions, so we will investigate whether accuracy is better when test questions are categorized separately or grouped with reflection questions during training and updating of student models.

We also plan to test whether filtering out turns with responses that are unrecognized by the system is better for prediction accuracy than leaving them in and treating them as incorrect responses. Currently the system classifies unrecognized student input as incorrect. The rationale for this policy is that it would be more harmful to skip a dialogue that could be potentially helpful to a student than to repeat information that the student already knows. However, this policy negatively affects the accuracy of the student model since it causes it to update using false data (i.e., unrecognized correct responses that are classified as incorrect). Filtering unrecognized input would be done both when a model is initially trained and when it is updated in real time. That is, if we filter out responses that the system could not recognize during training of the model, then at run time we will not update the student model when a response cannot be recognized. The refined training data will be used to fine-tune other parameters of the modelling algorithm, such as the learning rate—how fast the model adapts or "learns" from new data—and to improve the responsiveness of the model with respect to dynamic updates. Related ITS research has shown that using different modelling parameters—which consequently result in different learning curves for different subpopulations of the student population (i.e., fast learners and slow learners)—can provide more accurate metrics for student learning (Chang et al. 2006; Doroudi and Brunskill 2019) and, overall, more accurate predictions (Chounta and Carvalho 2019).

Finally, we plan to explore whether building separate student models for different phases of interaction improves prediction accuracy. For example, we could build one model to predict the correctness of answers to reflection questions and a separate one to predict the correctness of dialogue questions.

5.2 Planned Enhancements to System Features that Impact the Student Model's Performance

Two aspects of a tutorial dialogue system that are external to the student model but nonetheless impact its performance include knowledge representation—how the knowledge components that the tutor tracks are structured—and natural-language recognition capability. We describe some of our plans to improve these capabilities within Rimac.

5.2.1. Knowledge representation.

The complexity of a student model—namely, the number of predictive factors relative to the number of observations in the training data—may lead to overfitting. Complexity is proportional to the number of KCs that the dialogues address; the more KCs, the greater the complexity. Since Rimac tracks a large number of KCs (~260), complexity likely contributed to overfitting.

One way to reduce student model complexity is therefore to reduce the number of KCs. Representing KCs as a knowledge hierarchy instead of as a flat list would support conceptual grouping of KCs. For example, all KCs that address Newton's Second Law could be represented by one abstract "super KC". We can reduce the number of KCs included as predictor variables in the model if we infer the higher-level "super KCs" from their child KCs instead of including them in the model as separate predictor variables. Alternatively, the system could maintain separate models for different groupings of KCs, as noted previously with respect to reflection questions and dialogue questions. Other possible groupings could be by topic (mechanics, electricity, thermodynamics, etc.) or by type of knowledge (procedural, conceptual, metacognitive, etc.).

5.2.2. Natural-language recognition

Although the accuracy of the NLU approach for short dialogue responses that is used in Rimac is high [i.e., Rosé et al. (2001) measured an accuracy of 96.3%], this approach depends on creating both semantic and syntactic representations of typical student responses based on the current dialogue context. However, sometimes students respond in atypical or unexpected ways that often indicate their affective state—for example, profanity that stems from frustration (Zapata-Rivera et al. 2018). We expect that if a student expresses himself in an atypical way on occasion, it would not be so noticeable or negatively consequential. But if he frequently expresses himself in an atypical way, then NLU will not recognize these responses, which could result in a less positive experience for that student (Dzikovska et al. 2010). The consequence for a student when he answers correctly but is not understood is that the TDS lengthens the dialogue and covers material the student may already know.

We plan to test improvements to our approach to dealing with unrecognized responses. Rimac already incorporates an algorithm that uses restatement with or without verification from the student when the best recognition score for a student response is below threshold but still well above zero (Jordan et al. 2012). Currently a match between a student response and the best candidate concept is categorized as high, medium, low or unanticipated. If the highest score for a student input matches a candidate concept at a score of 0.8 or greater, the match quality is set to high. If the input-to-concept matches with a score between 0.8 and 0.7 the match quality is set to medium. If the input-to-concept matches with a score between 0.7 and 0.6 the quality is set to low. A score below 0.6 between the input and the best candidate concept means it was not understood. When the match quality for a student's input is high, that input is treated as understood. When the match quality is medium the system says, "I understood you to say <phrase that represents the concept matched>" (i.e., the system revoices what it heard). When the match quality is low the system asks, "Are you saying <phrase that represents the concept matched>?" If the student says yes, then it goes with that match. If the student says no with no attempt to restate, then the system marks it as unanticipated. If the student says no and answers again then the system attempts to understand this new response instead. We plan to test the effect of lowering the values that define high, medium and low after adding a nonsense detector and an off-topic detector. During field trials, we will check students' perception of the increased use of this strategy (e.g., Is it confusing when it happens? Is it happening too frequently?).

We also plan to test the performance of deep learning approaches for dialogue as a possible means of improving recognition. Although deep learning models for dialogue are showing promise (e.g., Gunasekara et al. 2019), to our knowledge, none so far have been trained using computer-human interaction data; they have been trained only using human-human interaction data. Thus, the effect of the noisiness of our computer-human training data (i.e., misrecognitions of student responses) on performance is unknown. Although we could correct the misrecognitions manually, the follow-up dialogues would then no longer be appropriate and there would be a new type of data noisiness introduced. We plan to test the accuracy of these NLU changes on held-out data to determine if any of these approaches delivers significant improvements in recognition accuracy.

6. Related Work

The sample of tutorial dialogue systems described in this section illustrates various approaches to providing adaptive instruction. We chose six frequently cited TDSs that influenced subsequent TDS development (over 30 dialogue-based ITSs have been developed), especially of approaches that attempt to track information about a student for more than a single turn in order to adapt the dialogue. As VanLehn (1988) noted, it is important to bear in mind that any intelligent tutoring system's student model can best be thought of as representing the tutor's *perception* of the student's cognitive state, not the student's *actual* cognitive state. We focus on modelling student cognition instead of affect because the former has been our focus in developing Rimac. To highlight Rimac's contributions, we point out differences relative to Rimac's approach to adaptivity during dialogue. Except as relevant, we do not describe the various natural-language understanding and generation techniques nor the full pedagogical approaches that TDSs have used. Additional information about these and other topics related to student modelling in conversational tutors can be found in several review articles—for example, Pavlik et al. (2013), Brawner and Graesser (2014), Bimba et al. (2017) and Alkhatlan and Kalita (2018). (See also papers on the tutorial dialogue systems cited and discussed in this section.)

To inform dialogue adaptations, some TDSs include a dedicated student model as part of their system architecture, while others rely on localized diagnosis and classification of the representations they derive during language understanding to model what the student currently knows (i.e., they don't track changes in student knowledge). The particular approach(es) to student modelling and adaptivity that TDSs implement depend on various factors including the developers' research goals and constraints, underlying theories of learning and instruction, and research on human tutoring. Most of the TDSs discussed in this section focus more on understanding the student's recent natural-language contributions in order to drive adaptivity, whereas Rimac has not yet done so because it encourages short-answer responses to allow for better input recognition. However, encouraging short answers is not feasible in some learning contexts and a mixture of response types is ultimately preferred.

CIRCSIM-Tutor (Evens and Michael 2006) is the first fully developed tutorial dialogue system, and the first to implement a student modelling module that tracks student performance during the dialogue in order to make micro-adaptive decisions. We focus on CIRCSIM-Tutor version 3 (Zhou and Evens 1999) because it best illustrates how the multi-level structure of its student model would support dynamic planning of adaptive dialogue. Unfortunately, this version of the student model was not included in any CIRCSIM-Tutor evaluations so its potential to support student learning remains untested.

CIRCSIM-Tutor adopts an "overlay and buggy modelling" approach. Students enter their predictions about the relationship between several parameters during three domain phases in a Prediction Table. The tutor compares students' predictions with an expert's predictions in order to identify correct and incorrect (buggy) predictions. Similarly, during dialogue, the tutor compares students' responses with expert responses in order to diagnose and classify each response according to its degree of correctness (e.g., correct, near hit, near miss). Each tutorial dialogue remediates one incorrect (buggy) prediction in the Predictions Table by dynamically building a hierarchical dialogue plan.

The student modelling component constructs a four-tiered performance model for each student: a "local" assessment for domain concepts, an assessment for the three system phases that the student makes predictions about in each problem, an assessment for each problem, and a global assessment across problems. Scores at each level are propagated upwards to compute scores at higher levels. Each level of this student model can provide different types of information to planning rules that implement macro- and micro-adaptive decisions. For example, information at the local (concept) level can contribute to decisions about whether to ask a follow-up question about a particular concept. Information at the phase level can help select tutoring method(s) to achieve particular tutoring goals—for example, determine what type of hint to provide. Information at the problem level can be used to design a lesson plan, while information at the global assessment level can support the tutor in choosing the next problem. However, CIRCSIM-Tutor's implementation, like Rimac's, focuses on micro-adaptive decision rules. Rimac also uses overall performance on concepts (KCs) to guide dynamic decision making—in particular, to decide at each dialogue step which node in a KCD's finite state network to traverse to. However, in order to provide the high degree of experimental control necessary to address our research questions, Rimac does not currently dynamically alter its dialogue scripts.

Unlike CIRCSIM-Tutor's student model, Rimac's base model is learned from prior student dialogues and updates are weighted based on prior student data rather than by using the same weightings for every concept. Concept unique weightings have the potential to adjust for concepts of varying degrees of difficulty but evaluating the benefits of doing so remains for future work.

EER-Tutor (Weerasinghe et al. 2009) consults a hierarchy of errors that students make in the Entity-Relationship domain in order to diagnose students' solutions to database design tasks. EER-Tutor provides adaptive feedback in the form of scripted dialogues that are linked to each error category. It builds two constraint-based student models: a short-term model of satisfied and violated constraints in the current solution and a long-term student model that records constraint histories. These models represent students' problem-solving performance, not their performance during dialogue. Since the system does not use dialogue-based evidence of students' domain knowledge and reasoning skills to update the student models (e.g., the types of errors made and the level of prompting the student needed in order to correct these errors), the tutor cannot use this information to guide adaptive decision making.

A small set of "adaptation rules" customize the scripted dialogues that address errors (constraint violations) in the student's solution. These rules represent all three aspects of scaffolding contingency, although the system developers do not claim to have specified them with this intention. For example, one rule targets temporal contingency; it tells the tutor how long to wait before intervening when a student seems to be idling. Another rule targets domain contingency by controlling how to choose an error to address when there is more than one error in a student's submitted solution. EER-Tutor makes this decision both reactively, based on its error diagnosis, and proactively. It chooses the error type that the student has the highest probability of making in future solution attempts. Most of the remaining rules support instructional contingency. For example, one rule determines whether to issue a problem-independent question before asking a contextualized, problem-specific question.

Although EER-Tutor's adaptation rules are domain-neutral, the dialogues that instantiate these rules are scripted. Similarly, the authoring guidelines that Rimac's dialogue developers consult to alter the level of support provided by questions, hints, etc. apply across quantitative problem-solving domains. However, Rimac's rules differ from EER-Tutor's rules in two main ways: they are more extensive and are grounded in scaffolding theory. The EER-Tutor, but not the adaptation rules per se, is based on constraint-based learning theory. A study that compared EER-Tutor with adaptive dialogues and EER-Tutor with non-adaptive dialogues indicated that a limited set of rules to guide adaptive tutoring is better than none (Weerasinghe et al. 2010).

Weerasinghe et al. (2009) discuss the potential benefits of updating the student models during dialogue in future versions of EER-Tutor. For example, recording which types of errors a student makes and how much prompting the student needed in order to correct an error could be used to assess the error's associated constraint(s) and to determine, more broadly, if the student's reasoning skills have improved. Rimac's dynamic updating of the student model during its dialogues is one of the critical features that supports micro-adaptive tutoring. It also supports macro-adaptive tutoring. Dynamic assessment of knowledge components informs Rimac's decisions about which reflective dialogues a student could profitably skip.

The **Geometry Explanation Tutor** (Aleven et al. 2001) also searches a hierarchical ontology that contains complete and buggy explanations against which it classifies a student's contributions. The classification serves as a proxy model of the student's ability to produce complete and accurate explanations. The tutor responds adaptively by issuing a scripted feedback message that is associated with its corresponding response category.

As the developers acknowledge, the tutor does not customize its feedback because it lacks the data and functionality necessary to do so, such as a dialogue history and dynamic planning. In terms of scaffolding theory, it implements domain contingency by providing feedback that addresses students' errors but does not implement instructional contingency as do most TDSs, including Rimac. Nonetheless, a classroom study indicated that domain contingency alone can be beneficial. Students who used the dialogue version of the Geometry Explanation Tutor produced higher quality explanations than students who used the menu-driven version, although neither group outperformed the other with respect to problem-solving performance (Aleven et al. 2004).

Like all cognitive tutors, the Geometry Explanation Tutor and its predecessor [the PACT Geometry Tutor; Aleven et al. (1999)] choose appropriate problems for students to work on. However, since information about students' explanation performance during dialogue is not used to update the system's student model, macro-adaptation is limited to selecting problems that develop students' geometry skills, not their explanation skills.

Beetle-2 (Dzikovska et al. 2014) lacks a model of students' task performance to refer to, unlike the above TDSs but similar to Rimac. However, like EER-Tutor, it responds to errors that students make during problem-solving tasks. It dynamically analyzes the dialogue state across multiple turns by comparing the student's explanations, and other input, with benchmark responses. This analysis produces a "diagnosis structure" that represents correctly mentioned objects and relations in students' responses and missing, irrelevant, and contradictory parts. This "diagnosis structure" serves as a model of students' understanding of domain concepts and relations. Beetle-2's tutorial planner then uses this model to select a generic tutoring tactic and works with its natural-language dialogue generator to instantiate this tactic in a contextualized manner—for example, by mentioning objects from the student's problem-solving work.

Instructional contingency is achieved by dynamically choosing more directive tactics, as necessary, to address the selected error in the diagnosis structure during a remedial dialogue.

As Beetle-2's developers acknowledge, one limitation is its lack of a persistent student model that could guide adaptive task selection. The order of problems in the curriculum and the order of questions within each dialogue are pre-specified. Although macro-adaptivity in Rimac is currently limited to deciding whether to skip a reflective dialogue, its persistent student model could be used to personalize problem and dialogue selection in a future version of the tutor. Also, Rimac adapts its tactics both reactively and proactively according to a student's cumulative performance, which also requires a persistent student model.

An evaluation of Beetle-2 that compared it with a no-training control suggested that its curriculum is effective. However, no significant differences were found when comparing a dialogue version to one that simply gave the correct answer when the student was not correct (Dzikovska et al. 2014). The Guru (Olney et al. 2012) and iStart (e.g., Allen et al. 2015; McCarthy et al. 2020; McNamara et al. 2007) tutoring systems are similar to Beetle-2 in that they focus on adapting student-tutor interactions according to a precise understanding of the student's recent contributions, not on building a persistent student model as in Rimac.

AutoTutor (e.g., Graesser 2016; Graesser et al. 2017a) supports Expectation and Misconception-Tailored (EMT) dialogues and aims to model novice tutors (e.g., a peer tutor). As with Beetle-2 and Rimac, there is no student model of problem solving to consider. In this case it is because there is no separate problem solving that precedes a dialogue. A separate EMT frame is pre-built for each problem that the system covers (a labor-intensive process). Its slots include anticipated expectations and their components and anticipated misconceptions. Misconceptions are addressed didactically through scripted explanations whereas discussions about an expectation in an EMT frame can take place across one or several turns until each expectation is adequately covered. Students' turns are analyzed using a speech act classifier (Olney et al. 2003) and latent semantic analysis (LSA) by comparing the student's contributions to the expectations and misconceptions encoded in the EMT frame.

AutoTutor achieves domain contingency by consulting the EMT frame to choose an expectation to discuss next that is incomplete or missing in the student's response. Graesser et al. (2004) state that a "processing module" manages this decision when there is more than one unfulfilled expectation (or expectation component) but they do not specify this module's decision rules or criteria. The tutor's default protocol is to elicit an expectation or one of its components by issuing a series of increasingly directive tutoring tactics, stopping when the expectation has been satisfied: first pump (e.g., "tell me more"), then hint, then prompt; if all else fails, assert (state the expectation). Instructional contingency can take place by varying the entry point into this default sequence. For example, an early version of AutoTutor (Graesser et al. 2003) used fuzzy production rules that considered factors such as dialogue history and student ability (e.g., based on pretest performance) to decide where to start. With a high ability student, the tutor might skip the pump and start with a hint. In contrast, Rimac updates information on student ability throughout the dialogue and this information persists across discussions with the student. Rimac achieves domain and instructional contingency by consulting its student model to decide which node to traverse to next and how to address the selected node's associated content (i.e., how much support to provide), respectively. For example, a hint will provide more or less support depending on the student's ability.

Students using AutoTutor have shown greater learning gains than students using a variety of simpler alternatives to a TDS (e.g. reading a textbook) and have shown similar learning gains to those tutored by human experts, when domain coverage is similar (VanLehn et al. 2007). However, there have been no comparisons that directly test the effectiveness of its form of adaptivity to that implemented in other TDSs.

Graesser et al. (2017a) describes several possible improvements. One is that assessment information could be attached to expectations and misconceptions in order to assess the student's performance on particular expectations (and misconceptions) or on the problem as a whole. For example, an expectation could be scored based on the amount of support that the student needed in order to meet that expectation—that is, how far along the assistance protocol (pump, hint, etc.) did scaffolding get? In addition, by mapping expectations to theoretically grounded knowledge components, students' performance on these KCs could be tracked across problems, thereby producing a persistent student model.

DeepTutor (Rus et al. 2013a; Rus et al. 2015) explores macro-adaptivity in tutorial dialogue systems and takes a similar approach to AutoTutor for providing micro-adaptive feedback. DeepTutor uses a framework called a *Learning Progression Matrix* to model students' level of proficiency in each course topic and across a sequence of increasingly difficult topics that the course covers. A learning progression matrix is a hierarchically structured construct. A course consists of topics, which are addressed through a series of lessons. Each lesson includes a series of tasks (problems, dialogues, and other activities). A task is accomplished through a series of solution steps and each step can be facilitated through a series of tutoring tactics (hints, pumps, prompts, etc.). Adaptivity can be applied by implementing alternative instructional strategies at each level. To date DeepTutor has focused on implementing macro-adaptivity at the course level and micro-adaptivity within each dialogue task.

Learning progressions model students' journey towards mastery in a particular domain: "learners go through a pathway of knowledge states, where each state has distinct patterns of understanding and misconceptions" (Rus et al. 2013b, p. 4). The goal of macro-adaptive tutoring is to design a customized *learning trajectory* for a particular student—a set of tasks that will advance the student to higher states within a particular course topic and across the topics included in the course's curriculum. An initial learning trajectory can be defined based on a student's pretest scores and then dynamically adjusted based on a student's performance on problems, short multiple-choice tests, and dialogue contributions.

DeepTutor selects dialogue tasks to accommodate the student's current state for a topic, as recorded in the student's learning progression matrix. Consequently, DeepTutor's dialogues focus on the limited set of expectations and misconceptions that are associated with the student's current state, not on the full suite of expectations that an AutoTutor dialogue typically addresses. This departure from AutoTutor's approach has the potential to support more targeted, customized learning conversations. Micro-adaptive scaffolding within each dialogue takes place by choosing which expectation, expectation component, or misconception to address in the next tutor step (i.e., domain contingency) and the "best" tactic to use to address this step (i.e., instructional contingency). The first decision is made based on the tutor's evaluation of the student's responses during dialogue, while the second decision is guided by a complex set of tactical decision rules, such as the "complex mechanism" that controls hinting (Rus et al. 2014a, p. 316).

DeepTutor potentially offers more support for macro-level adaptivity than Rimac currently does in that it encodes a curriculum and tracks progress through the curriculum (reminiscent of CIRCSIM-Tutor's levels). Macro-adaptivity in Rimac is currently limited to allowing a student to skip dialogues that address content that the student model indicates the student has sufficiently mastered. However, it remains for future work to determine how learning progressions could also support the type of proactive and reactive adaptations that Rimac implements. For example, would it be necessary to recognize that a student may be between the defined cell levels of the learning progression matrix?

A small-scale evaluation of an early version of DeepTutor (N=30) that compared students who used a micro-adaptive version with students who used a macro- and micro-adaptive version found significantly higher pretest to posttest learning gains in the fully adaptive condition (Rus et al. 2014b). In this version, students' proficiency levels were estimated based on overall pretest scores and no updates were made to proficiency levels during dialogue. In contrast, Rimac dynamically updates its student model during each dialogue.

Summary of Contributions. Few TDSs besides Rimac maintain a persistent student model—one that is dynamically maintained across tutoring tasks and sessions. A persistent student model is necessary to support the outer, macro-adaptive loop of VanLehn's two-loop framework (VanLehn 2006). Otherwise, system developers must specify the order of tasks (problems or dialogues). In addition, most TDSs make micro-adaptive decisions reactively, in response to students' performance. However, few TDSs besides Rimac also carry out proactive decision-making in order to challenge the student without overwhelming him, gently nudging the student towards mastery. Towards that end, Rimac performs micro-adaptive tutoring by predicting students' success in answering a set of candidate "next step" questions, chooses one (domain contingency), and then determines how much help to provide, through which tutoring strategies, etc. (instructional contingency). Hence, Rimac stands apart from other TDSs by dynamically building and maintaining a persistent student model that supports reactive and proactive decision making, in order to emulate the contingent scaffolding of human tutoring (Katz et al. 2018). The classroom studies described previously (Section 4) indicate that this combined approach to decision making supports more efficient tutoring than does reactive decision making alone. They also indicate that IFM effectively implements this approach.

7. Towards Integrating Tutorial Dialogue Systems in the Classroom

Despite strong evidence that adaptive tutorial dialogue systems support learning (e.g., Albacete et al. 2017a; Albacete et al. 2019; Dzikovska et al. 2014; Evens and Michael 2006; Forbes-Riley and Litman 2011; Graesser 2016; Graesser et al. 2017b; Weerasinghe et al. 2009; Weerasinghe et al. 2011), they have not yet become an integral part of classroom instruction. Addressing this disconnect between research and practice is especially important for blended learning classrooms, which combine online learning with traditional methods of instruction (lectures, textbook readings, problem-solving assignments, etc.). Many challenges remain to be met to achieve sustained use of TDSs including improved language understanding and student modelling performance, and identification of effective ways to sustain student engagement with TDSs and ITSs in general (e.g., Graesser et al. 2014; Jackson et al. 2012; Kopp et al. 2012). This section focuses on one limitation of TDSs that prevents their large-scale adoption: their lack of a companion open learner model (OLM). OLMs display students' progress and have the potential to maintain student model accuracy and student engagement (Bull 2020; Bull and Kay 2016). Our choice of this focal point is motivated by Jim's interest in using OLMs to perform similar roles in ITSs (e.g., Zapata-Rivera and Greer 2004a, 2004b). We show that dialogue data adds some wrinkles to the fabric of OLM design and discuss why incorporating some types of OLMs in TDSs may not be feasible long term, such as during an entire course.

7.1. Using OLMs to Display Student Progress

Considering several ITSs and adaptive hypermedia systems, Bull (2020) remarked that "there are open learner models [OLMs] about!"[3] As mentioned, an open learner model allows students, teachers, and other education stakeholders (e.g., parents, peers, school administrators) to inspect the contents of a tutor's student model in order to track students' progress. An OLM also allows users to interact with and

---

[3] "About" (British English) is roughly equivalent to "around" or "all over."

perhaps edit the student model in order to improve its accuracy. However, when it comes to TDSs, there are *no* "open learner models about". To our knowledge, all TDSs lack a companion OLM, even though the reverse holds true: several OLMs include natural-language interaction with a chatbot—for example, STyLE-OLM (Dimitrova 2003; Dimitrova and Brna 2016), NLDtutor (Suleman et al. 2016) and CALMsystem (Kerly and Bull 2008; Kerly et al. 2008; Kerly et al. 2007). This section focuses on the use of OLMs to track student progress through *inspectable OLMs*. Inspectable OLMs allow users to examine the student model and receive explanations for its assessment values but not change them. The next section addresses student model maintenance, automatically or manually through OLMs that give users varying degrees of control over the model's content.

Several studies have found benefits from allowing students to inspect and/or interact with their student model through an OLM. Germane to this festschrift, Jim Greer's work with Diego Zapata-Rivera showed that providing various forms of guidance to students as they interact with an OLM can increase the frequency of students' reflection on their work (e.g., Zapata-Rivera and Greer 2002, 2003; Zapata-Rivera and Greer 2004a). Reflection is an important metacognitive activity that can promote learning (e.g., Long and Aleven 2017). Other observed benefits of student interaction with OLMs include learning gains (Brusilovsky et al. 2015; Chen and Chan 2008; Duan et al. 2010; Girard 2011; Hsiao and Brusilovsky 2017; Long and Aleven 2017; Shahrour and Bull 2009), improved self-assessment accuracy among students who do not already demonstrate this skill (Kerly et al. 2008; Long and Aleven 2017; Suleman et al. 2016), confidence gains (Al-Shanfari et al. 2017), and heightened motivation to use the tutoring system (e.g., Bull and Pain 1995; Long and Aleven 2017; Thomson and Mitrovic 2010).

OLMs have also proven to be useful for teachers. In particular, "teacher dashboards" combine the behavioral data of learning analytics with the system-inferred assessment data of OLMs. They support teachers in planning adaptive instruction for individual students, small groups, or an entire class (e.g., Bull and McKay 2004; Girard and Johnson 2008; Grigoriadou et al. 2010; Pérez-Marín and Pascual-Nieto 2010; Riofrío-Luzcando et al. 2019; Xhakaj et al. 2017a, 2017b; Yacef 2005; Zapata-Rivera and Greer 2001; Zapata-Rivera et al. 2007). Teacher dashboards have also been developed to facilitate real-time classroom orchestration, a term used to describe how a teacher monitors and guides her classroom (e.g., Holstein et al. 2018a; Holstein et al. 2018b; Holstein et al. 2018c; Holstein et al. 2019). Recent studies indicate that teachers' use of learning analytics dashboards to support classroom orchestration can promote learning gains (Holstein et al. 2017b; Holstein et al. 2018c). [See Bull (2020) for a thorough review of key findings on the benefits and limitations of OLMs for teachers, students, and other users.]

Several questions need to be addressed in order to develop OLMs for TDSs that could result in similar benefits. As with the design of any score report, the first critical step is to conduct an audience analysis, in order to find out what questions various users would bring to an OLM or teacher dashboard for a TDS (Zapata-Rivera and Katz 2014). We focus on teachers and expect considerable overlap between teachers' questions and students' questions (Zapata-Rivera 2019). Teachers will likely want to be able to inspect visualizations of a student's performance on particular concepts, such as skill meters and bar graphs, as a student would when inspecting their personalized OLM. However, teachers will likely also want to view aggregate assessments for their class as a whole. For example, in order to plan a lesson for an upcoming class, a teacher might want to see average scores on selected knowledge components, a summary list of KCs with the lowest average scores, common misconceptions, etc. (Aleven et al. 2016b; Xhakaj et al. 2016). Teachers will also likely want to inspect behavioral data, such as: How many problems and/or dialogues did a student complete for a particular assignment? What was the average completion rate for a given class? How many reflection questions or questions asked during a particular dialogue did a student leave blank or respond to with gibberish, possibly indicating disengagement?

Other, less predictable questions that teachers might bring to a dashboard for a TDS could be identified through user-centered design sessions (e.g., Aleven et al. 2016b; Epp et al. 2019; Holstein et al. 2018a; Holstein et al. 2017a, 2019; Holstein et al. 2010; Xhakaj et al. 2016). For example, to what extent do teachers want to "drill down" in order to receive explanations about the reasoning behind the student model's competency assessments? These explanations would present the evidence that the student model used to infer that a student has a good or poor understanding of a particular concept and how much weight each piece of evidence contributed to this inference. Explanations such as these are available in tutoring systems that incorporate an inspectable or interactive OLM, and in *independent OLMs* (i.e., OLMs that are not associated with a particular tutoring system) (e.g., Bull 2016; Bull and Kay 2016; Bull and Pain 1995; Dimitrova 2003; Dimitrova and Brna 2016; Ginon et al. 2016; Kerly et al. 2008; Suleman et al. 2016; Tchetagni et al. 2007).

Providing similar explanations in TDSs raises interesting design challenges. The main evidence that a TDS uses to infer students' competency on knowledge components is dialogue data. Displaying tutor-student exchanges during dialogue (e.g., a tutor's question followed by the student's response) is considerably more complex than, say, showing a list of quiz scores or test items that a student missed. For example, while the tutor explains a low score on a concept, how many incorrectly answered questions that target that concept can the OLM display without causing clutter and information overload? How can a dialogue exchange be sufficiently contextualized so that it makes sense? Would it suffice to show one exchange back? Or should contextualization of dialogue-based evidence be personalized—for example, by allowing teachers (and other users) to ask to "see more" or "see less", as necessary? Some OLMs visualize student progress over time (e.g., Ferreira et al. 2017; Rueda et al. 2007; Rueda et al. 2003). Accommodating teachers' potential desire to track student progress in understanding key concepts across a series of dialogues will likely raise even more challenging design issues.

As with any ITS, the student model in a TDS will have more confidence in some of its assessments than others. In addition to presenting dialogue-based evidence and explaining how competency ratings were calculated, an OLM for a TDS could represent model (un)certainty by adapting its display in various ways (Epp and Bull 2015). However, it is important to test carefully to ensure that teachers can interpret uncertainty representations correctly. For example, Zapata-Rivera et al. (2016) found that teachers tend to have difficulty interpreting error bars in score reports, although a brief video tutorial sufficed to resolve this problem.

7.2. Using OLMs to Maintain Student Model Accuracy

Students' cognitive state is a moving target, constantly changing in response to their interactions with a tutoring system and various external learning resources—for example, other online materials, teachers, peers, family, friends, and textbooks. As with any ITS, a TDS's student model will quickly become outdated without some means in place to update it. Student model maintenance can take place automatically and/or manually through user interaction with an OLM (Bull 2016, 2020). For example, the Next-TELL OLM automatically integrates assessment data that is transferred from multiple sources through its application program interface (API) (e.g., Bull et al. 2013; Bull and Wasson 2016). If a teacher assigns weights to various data resources, as in EI-OSM (Zapata-Rivera et al. 2007), this can help students understand discrepancies between the OLM's representations and their expectations. However, a drawback of integrating multiple data sources in a student model (and OLM) is that this can increase student model inaccuracy, due to varying formats, degrees of reliability, specificity, etc. As Bull (2016) noted, "With the variety of learning data now available, we have to accept that there is greater room for imprecision in the learner model" (p. 19).

A student model can become outdated and inaccurate for several other reasons, including the system's inability to automatically import data from online and standalone learning systems; learning that takes place offline, through instructional activities such as reading a textbook and interacting with teachers, peers, etc.; "noise" from student guesses and memory lapses (slips). In the context of TDSs, inaccuracy also stems from the system's inability to understand some of the student's dialogue input. Consequently, there is a strong need for teachers and students to be able to interact with a (future) TDS's OLM in order to manually maintain its student model.

As described in the SMILI☺ framework (Bull 2020; Bull and Kay 2016), the various types of OLMs referred to in the literature differ mainly according to how much control they allow students to exert over the student model's content. At one end of the spectrum stands *inspectable only OLMs*, which give the system full control over model updating; students can only examine representations of the model's competency ratings and perhaps explanations about the reasoning that underlies these ratings. At the other end of the spectrum stands *editable OLMs,* which allow students to directly change the model until new evidence that may override the student's self-assessment comes along. Editable OLMs risk introducing more inaccuracy due to students' erroneous ratings, while inspectable-only OLMs prevent obvious errors in the model from being corrected—for example, a momentary slip or, in the context of a (future) OLM for a TDS, unrecognized student dialogue input.

For these and other reasons, various alternative approaches to interactive student model maintenance stand between these two goal posts (inspectable-only and editable OLMs) (Bull 2020), for example: OLMs that allow students to add information to their model but not replace its content (*learner adds information OLMs*); OLMs that allow users to challenge the model and prove that they know more (or less) than the system thinks they do, typically by answering a challenge question (*persuadable OLMs*); and OLMs that encourage negotiation, allowing the student and the system to justify their position until they either reach agreement or must defer to a policy, typically set by teachers, about what to do if agreement can't be reached (*negotiable OLMs*). In VisMod, Zapata-Rivera and Greer (2004b) gave teachers similar status as the system. Teachers could veto students' proposed changes to the student model (Bull 2020).

Several exploratory studies suggest that interactive model maintenance through an OLM does indeed promote improved consistency and accuracy, when the system maintains some degree of control (e.g., Bull and Pain 1995; Dimitrova 2003; Dimitrova et al. 2001; Suleman et al. 2016). Other potential benefits include increased student agency over learning (Bull and Kay 2007, 2016), metacognitive behaviors such as planning and self-monitoring (Bull and Kay 2013), learning gains (Kerly and Bull 2008), and increased motivation when students perceive their interaction with an OLM as a break from the system's main task (Thomson and Mitrovic 2010). However, as Bull (2020) stated, an important goal for future research is to determine which types of OLMs are best suited for different types of students, at different stages of learning, and with different types of tasks (p. 19). Research to address this issue indicates that we can't always trust our intuitions about which mappings are best—for example, that mature students will be able to edit their student model appropriately. Britland (2010) found that university students were unable to detect and correct errors that were planted in an editable OLM.

The importance of considering the nature of a tutoring system's tasks and activities is especially relevant to OLM design for tutorial dialogue systems. The last thing that a student who has just completed a lengthy automated dialogue probably wants to do is convince the tutor to alter their student model, especially if this would entail a detailed, Toulmin style negotiation activity (e.g., Van Labeke et al. 2007; Zapata-Rivera et al. 2007) or yet another dialogue (e.g., Kerly et al. 2008; Suleman et al. 2016). Consequently, another challenge for future research in OLM design for TDSs is to discover how to reap the benefits of interactive model maintenance—in particular, negotiable OLMs—without compromising student motivation.

Table 7: An initial research agenda to inform the development of OLMs for TDSs

- What information do various users (teachers, students, parents, etc.) want to receive from a TDS's OLM?
- What are effective ways to display and contextualize dialogue-based evidence for the system's assessments, in order to explain these assessments?
- Which type(s) of OLMs are best suited for tutorial dialogue systems and for different types of students who use these systems?
- How can we reap the benefits of interactive student model maintenance through an OLM without jeopardizing student motivation?
- What factors constrain the feasibility of sustained classroom use of TDSs, and ITSs in general?
- How can we reduce the teacher workload required to develop student models and OLMs for long term use of TDSs in the classroom?
- To what extent does the inclusion of OLMs in TDSs result in similar benefits to those observed from including OLMs in other types of tutoring systems, such as learning gains, increased motivation, and improved self-assessment accuracy?

Another challenge that has received little attention to date is to find ways to reduce the teacher workload required for student model and OLM development. This challenge pertains to all types of OLMs (independent, inspectable-only, or interactive), and to automated and manual student model maintenance. Teachers who have participated in OLM research and development have been "assigned" a multitude of tasks such as defining conditional probabilities (Zapata-Rivera and Greer 2004b); specifying weights for evidence based on credibility, relevance, and other factors (Zapata-Rivera et al. 2007); specifying reasons why a value should be changed so that students can choose these reasons from a menu, and setting thresholds to determine when evidence is sufficient to change assessment values (Ginon et al. 2016); engaging in negotiation discussions with individual students about their student model (Zapata-Rivera and Greer 2004b); defining performance expectations at various points during a course so that students can determine if they are on track (Bull and Mabbott 2006); including supplemental feedback (Bull et al. 2015); associating tasks and test items with knowledge components, etc. It may be feasible for teachers to perform these roles short term, as teachers have done during brief usability studies such as those cited in the preceding paragraph. However, we are skeptical that teachers could sustain these roles throughout a course—not only in TDSs, but in ITSs in general. For example, Zapata-Rivera et al. (2007) reported that teachers found EI-OSM useful, but teachers also cautioned that they have limited time to calibrate the system's evidence parameters. As the authors eloquently stated, "Teachers wanted control over a system that can perform most tasks autonomously" (p. 298).

Feasibility of teacher involvement with student model and OLM development will likely depend on several contextual factors such as class size and course load. It is hard to envision teachers in large schools, with large classes, interacting with individual students about their OLM, ok'ing (or vetoing) students' requested changes to the system's assessment values, mapping test items to KC's for every quiz and assignment included in a learner management system so that this data can be integrated in a tutor's student model and OLM, etc. Hence, important tasks to add to the research agenda include specifying factors that constrain the feasibility of sustained use of ITSs and finding ways to automate or semi-automate as many teacher roles as possible. For example, perhaps a tutoring system could use large datasets to automatically learn or recalibrate evidence weightings.

Last but certainly not least, an important goal for future research is to determine if integrating OLMs in TDSs results in similar benefits to those found by combining these technologies in other types of adaptive tutoring systems such as learning gains, improved self-assessment ability, and increased motivation to use the tutor. Table 7 summarizes the research and design questions raised in this section, to set a roadmap for future work.

8. Conclusion

Long before computers that could help people learn put a glint in educators' eyes, a wise man observed that "Necessity knows no law except to conquer."[4] Whether or not Jim was familiar with this maxim, he argued for it, in effect, when he urged tutoring system developers to cast aside commonplace rules like "favor simplicity over complexity" and to instead take on the complex, difficult task of student modelling in order to meet a greater necessity: making these systems *adaptive* to the learner. Many effective problem-solving ITSs that incorporate a student model support his position.

In this paper, we argued that it is time for more developers of tutorial dialogue systems to follow suit and incorporate a student modelling component that can guide the tutor in providing adaptive support during complex conceptual discussions. We described how we do this in Rimac and summarized studies that found that: (1) an experimental version of Rimac that links a "poor man's student model" (i.e., a static, pretest-initialized student model) with automated dialogue promotes more efficient learning than a control version that lacks a student model, among high prior knowledge students, and (2) a dynamically updated student model promotes more efficient learning than the static, "poor man's student model", among both high and low prior knowledge students. Although we have not yet observed an advantage of student model driven dialogue with respect to learning gains, we expect that this will follow from improving the performance of the student model, including more content to support long-term use of the tutor, and incorporating OLMs that provide tools to allow students and teachers to interact with and maintain the student model. Future generations of students who learn more effectively and efficiently from student model endowed dialogue tutors will unwittingly have Jim to thank.

A personal note from the first author (Sandy Katz)

I arrived at the ITS 2018 venue in Montreal unfashionably late for the morning's keynote address. Trying to slip in undetected, while crossing the hotel lobby I heard a familiar voice call my name. It was Jim, walking slowly towards me with a cane that I couldn't recall seeing him use before, but it had been several years since our conference paths last crossed. We caught up easily, as we had over the years at ITS-related conferences near and far, from Montreal to Crimea. Neither of us made it to the keynote, but I had no regrets, especially after learning that Jim passed away just a few days later.

If successful, this paper pays tribute to Jim and the lasting impression that his keen insights about how to wield technology to provide students with adaptive tutoring made on my work. But this tribute would not be complete without noting that Jim was himself a model of kindness, fairness, and generosity. We still have much to learn from him, morally and intellectually.

Acknowledgements. We thank Sarah Birmingham, Dennis Lusetich, and Scott Silliman for many valuable contributions to this project. Insightful comments from the reviewers and editors on a previous

---

[4] The author of this maxim, which is translated from Latin, is Publilius Syrus (85-43 BC).

draft broadened our thinking about the state of TDSs and important "next steps" that need to be taken. 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 expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education.

References

Ai, H., & Litman, D. (2011). Assessing user simulation for dialog systems using human judges and automatic evaluation measures. *Natural Language Engineering, 17*(4), 511-540, doi:https://doi.org/10.1017/S1351324910000318.

Al-Shanfari, L., Epp, C. D., & Baber, C. (2017). Evaluating the effect of uncertainty visualisation in open learner models on students' metacognitive skills. In E. André, R. S. Baker, H. Xiangen, M. M. T. Rodrigo, & B. du Boulay (Eds.), *Proceedings of the 18th International Conference on Artificial Intelligence in Education, AIED 2017, Wuhan, China, June 28 - July 2 2017* (Vol. 10331, pp. 15-27, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-61425-0_2.

Albacete, P., Jordan, P., & Katz, S. (2017a). Is a dialogue-based tutoring system that emulates helpful co-constructed relations during human tutoring effective? In C. Conati, N. Heffernan, A. Mitrovic, & M. Verdejo (Eds.), *Proceedings of the 17th International Conference on Artificial Intelligence in Education, AIED 2015, Madrid, Spain, June 22-26 2015* (Vol. 9112, pp. 3-12, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-19773-9_1.

Albacete, P., Jordan, P., Katz, S., Chounta, I. A., & McLaren, B. M. (2019). The impact of student model updates on contingent scaffolding in a natural-language tutoring system. In S. Isotani, E. Millán, A. Ogan, P. Hastings, B. M. McLaren, & R. Luckin (Eds.), *Proceedings of the 20th International Conference on Artificial Intelligence in Education, AIED 2019, Chicago, Illinois, June 25-29 2019* (Vol. 11625, pp. 37-47, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-030-23204-7_4.

Albacete, P., Jordan, P., Lusetich, D., Chounta, I. A., Katz, S., & McLaren, B. M. (2018). Providing proactive scaffolding during tutorial dialogue using guidance from student model predictions. In C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, et al. (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence in Education, AIED 2018, London, UK, June 27-30 2018* (Vol. 10948, pp. 20-25, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-93846-2_4.

Albacete, P., Silliman, S., & Jordan, P. (2017b). A tool to assess fine-grained knowledge from correct and incorrect answers in online multiple-choice tests: An application to student modeling. In J. Johnston (Ed.), *Proceedings of the World Conference on Educational Media and Technology, EdMedia 2017, Washington, D.C., USA, June 21-23 2017* (pp. 988-996): Association for the Advancement of Computing in Education (AACE). https://www.learntechlib.org/primary/p/178413/. Accessed 25 Sept. 2020.

Aleven, V., Koedinger, K. R., & Cross, K. (1999). Tutoring answer explanation fosters learning with understanding. In S. P. Lajoie, & M. Vivet (Eds.), *Proceedings of the 9th International Conference on Artificial Intelligence in Education, AIED 1999, Le Mans, France, July 19-23 1999* (pp. 199-206). Amsterdam: IOS Press.

Aleven, V., McLaughlin, E. A., Glenn, R. A., & Koedinger, K. R. (2016a). Instruction based on adaptive learning technologies. In R. E. Mayer, & P. A. Alexander (Eds.), *Handbook of Research on Learning and Instruction* (2nd ed., pp. 522-560). New York, NY, USA: Routledge.

Aleven, V., Ogan, A., Popescu, O., Torrey, C., & Koedinger, K. R. (2004). Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In J. C. Lester, R. M. Vicari, & F. Paraguaçu (Eds.), *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004, Maceió, Alagoas, Brazil, Aug. 30 - Sept. 3 2004* (Vol. 3220, pp. 443-454, Lecture Notes in

Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-540-30139-4_42.

Aleven, V., Popescu, O., & Koedinger, K. R. (2001). Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), *Proceedings of the 10th International Conference on Artificial Intelligence in Education, AIED 2001, San Antonio, Texas, USA, May 19-23 2001* (pp. 246-255). Amsterdam: IOS Press. doi:http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.9561.

Aleven, V., Xhakaj, F., Holstein, K., & McLaren, B. M. (2016b). Developing a teacher dashboard for use with intelligent tutoring systems. In R. Vatrapu, M. Kickmeier-Rust, B. Ginon, & S. Bull (Eds.), *Proceedings of the 4th International Workshop on Teaching Analytics in conjunction with EC-TEL 2016, Lyon, France, Sept. 13-16 2016* (Vol. 1738, pp. 15-23, CEUR Workshop Proceedings).

Alkhatlan, A., & Kalita, J. (2018). Intelligent tutoring systems: A comprehensive historical survey with recent developments. *International Journal of Computer Applications, 18*(43), 1-20, doi:10.5120/ijca2019918451.

Allen, L. K., Snow, E. L., & McNamara, D. S. (2015). Are you reading my mind? Modeling students' reading comprehension skills with natural language processing techniques. In P. Blikstein, A. Merceron, & G. Siemens (Eds.), *Proceedings of the 5th International Conference on Learning Analytics and Knowledge, LAK '15, Poughkeepsie, New York, USA, 2015* (pp. 246-254). New York, NY, USA: Association for Computing Machinery. doi:https://dl.acm.org/doi/10.1145/2723576.2723617.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research, 70*(2), 181-214, doi:https://doi.org/10.3102/00346543070002181.

Baker, R. S. J. d., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies, 68*(4), 223-241, doi:https://doi.org/10.1016/j.ijhcs.2009.12.003.

Belland, B. R. (2014). Scaffolding: Definition, current debates, and future directions. In J. M. Spector, M. D. Merrill, J. Elen, & M. J. Bishop (Eds.), *Handbook of Research on Educational Communications and Technology* (4th ed., pp. 505-518). New York, NY, USA: Springer.

Bimba, A. T., Idris, N. I., Al-Hunaiyyan, A., Mahmud, R. B., & Shuib, L. B. M. (2017). Adaptive feedback in computer-based learning environments: A review. *Adaptive Behavior, 25*(5), 217-234, doi:https://doi.org/10.1177/1059712317727590.

Blessing, S. B., Gilbert, S. B., Ourada, S., & Ritter, S. (2009). Authoring model-tracing cognitive tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 189-210.

Brawner, K., & Graesser, A. C. (2014). Natural language, discourse, and conversational dialogues within intelligent tutoring systems: A review. In R. A. Sottilare, A. C. Graesser, X. Hu, & B. S. Goldberg (Eds.), *Design Recommendations for Intelligent Tutoring Systems* (Vol. 2, Instructional Management, pp. 189-204, Adaptive Tutoring Series). Orlando, Florida: U.S. Army Research Laboratory.

Britland, M. (2010). *Accuracy of a learner's edits to their learner model*. Masters by Research thesis, University of Birmingham, Birmingham, UK.

Brownfield, K. (2016). *Scaffolding in literacy learning and teaching: The impact of teacher responsiveness during writing on first grade students' literacy learning*. PhD dissertation, The Ohio State University, Columbus, Ohio, USA.

Brownfield, K., & Wilkinson, I. A. G. (2018). Examining the impact of scaffolding on literacy learning: A critical examination of research and guidelines to advance inquiry. *International Journal of Educational Research, 90*, 177-190, doi:https://doi.org/10.1016/j.ijer.2018.01.004.

Brusilovsky, P., Somyürek, S., Guerra, J., Hosseini, R., & Zadorozhny, V. (2015). The value of social: Comparing open student modeling and open social student modeling. In F. Ricci, K. Bontcheva,

O. Conlan, & S. Lawless (Eds.), *Proceedings of the 23rd International Conference on User Modeling, Adaptation, and Personalization, UMAP 2015, Dublin, Ireland, June 29 - July 3 2015* (Vol. 9146, pp. 44-55, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-20267-9_4.

Bull, S. (2016). Negotiated learner modelling to maintain today's learner models. *Research and Practice in Technology Enhanced Learning, 11*(1), 10, doi:https://doi.org/10.1186/s41039-016-0035-3.

Bull, S. (2020). There are open learner models about! *IEEE Transactions on Learning Technologies, 13*(2), 425-448, doi:10.1109/TLT.2020.2978473.

Bull, S., Johnson, M. D., Alotaibi, M., Byrne, W., & Cierniak, G. (2013). Visualising multiple data sources in an independent open learner model. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education, AIED 2013, Memphis, TN, USA, July 9-13 2013* (Vol. 7926, pp. 199-208, Lecture Notes in Artificial Intelligence). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-642-39112-5_21.

Bull, S., Johnson, M. D., Masci, D., & Biel, C. (2015). Integrating and visualising diagnostic information for the benefit of learning. In P. Reimann, S. Bull, M. Kickmeier-Rust, R. Vatrapu, & B. Wasson (Eds.), *Measuring and Visualizing Learning in the Information-Rich Classroom* (pp. 167-171). New York, NY, USA: Routledge.

Bull, S., & Kay, J. (2007). Student models that invite the learner in: The SMILI☺ open learner modelling framework. *International Journal of Artificial Intelligence in Education, 17*(2), 89-120.

Bull, S., & Kay, J. (2013). Open learner models as drivers for metacognitive processes. In R. Azevedo, & V. Aleven (Eds.), *International Handbook of Metacognition and Learning Technologies* (Vol. 28, pp. 349-365, Springer International Handbooks of Education). New York, NY, USA: Springer.

Bull, S., & Kay, J. (2016). SMILI☺: A framework for interfaces to learning data in open learner models, learning analytics and related fields. *International Journal of Artificial Intelligence in Education, 26*(1), 293-331, doi:https://doi.org/10.1007/s40593-015-0090-8.

Bull, S., & Mabbott, A. (2006). 20000 inspections of a domain-independent open learner model with individual and comparison views. In M. Ikeda, K. D. Ashley, & T. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, ITS 2006, Jhongli, Taiwan, June 26-30 2006* (Vol. 4053, pp. 422-432, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/11774303_42.

Bull, S., & McKay, M. (2004). An open learner model for children and teachers: Inspecting knowledge level of individuals and peers. In J. C. Lester, R. M. Vicari, & F. Paraguaçu (Eds.), *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004, Maceió, Alagoas, Brazil, August 30 - September 3 2004* (Vol. 3220, pp. 646-655, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-540-30139-4_61.

Bull, S., & Pain, H. (1995). "Did I say what I think I said, and do you agree with me?": Inspecting and questioning the student model. In *Proceedings of the 7th World Conference on Artificial Intelligence in Education, AIED 1995, Washington, DC, USA, Aug. 16-19 1995* (pp. 501-508): Association for the Advancement of Computing in Education

Bull, S., & Wasson, B. (2016). Competence visualisation: Making sense of data from 21st-century technologies in language learning. *ReCALL: The Journal of EUROCALL, 28*(2), 147-165, doi:https://doi.org/10.1017/S0958344015000282.

Chang, K.-M., Beck, J., Mostow, J., & Corbett, A. (2006). A Bayes net toolkit for student modeling in intelligent tutoring systems. In M. Ikeda, K. D. Ashley, & T. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, ITS 2006, Jhongli, Taiwan, June 26-30 2006* (Vol. 4053, pp. 104-113, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/11774303_11.

Chen, Z.-H., & Chan, T.-W. (2008). Learning by substitutive competition: Nurturing My-Pet for game competition based on open learner model. In *Proceedings of the 2nd IEEE International*

Conference on Digital Game and Intelligent Toy Enhanced Learning, DIGITEL '08, Banff, BC, Canada, Nov. 17-19 2008 (pp. 124-131): IEEE. doi:https://doi.org/10.1109/DIGITEL.2008.36.

Chi, M., Jordan, P., & VanLehn, K. (2014). When is tutorial dialogue more effective than step-based tutoring? In S. Trausan-Matu, K. E. Boyer, M. Crosby, & K. Panourgia (Eds.), *Proceedings of the 12th International Conference on Intelligent Tutoring Systems, ITS 2014, Dresden, Germany, Nov. 16-19 2014* (Vol. 8474, pp. 210-219, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-07221-0_25.

Chi, M., Koedinger, K. R., Gordon, G. J., Jordan, P., & VanLehn, K. (2011). Instructional Factors Analysis: A cognitive model for multiple instructional interventions. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, & J. Stamper (Eds.), *Proceedings of the 4th International Conference on Educational Data Mining, EDM 2011, Eindhoven, Netherlands, July 6-8 2011* (pp. 61-70). http://educationaldatamining.org/EDM2011/wp-content/uploads/proc/edm11_proceedings.pdf. Accessed 25 Sept. 2020.

Chounta, I. A., Albacete, P., Jordan, P., Katz, S., & McLaren, B. M. (2017a). The "Grey Area": A computational approach to model the zone of proximal development. In É. Lavoué, H. Drachsler, K. Verbert, J. Broisin, & M. Pérez-Sanagustín (Eds.), *Proceedings of the 12th European Conference on Technology Enhanced Learning, EC-TEL 2017, Tallinn, Estonia, Sept. 12-15 2017* (Vol. 10474, pp. 3-16, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-66610-5_1.

Chounta, I. A., Albacete, P., Jordan, P., Katz, S., & McLaren, B. M. (2017b). Modeling the zone of proximal development with a computational approach. In X. Hu, T. Barnes, A. Hershkovitz, & L. Paquette (Eds.), *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017, Wuhan, China, June 25-28 2017* (pp. 56-57)

Chounta, I. A., & Carvalho, P. F. (2019). Square it up!: How to model step duration when predicting student performance. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK '19, Tempe, Arizona, USA, March 4-8 2019* (pp. 330-334). New York, NY USA: Association for Computing Machinery. doi:https://doi.org/10.1145/3303772.3303827.

Conati, C., & Kardan, S. (2013). Student modeling: Supporting personalized instruction, from problem solving to exploratory open-ended activities. *AI Magazine, 34*(3), 13-26, doi:10.1609/aimag.v34i3.2483.

Cooper, G., & Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology, 79*(4), 347-362, doi:https://doi.org/10.1037/0022-0663.79.4.347.

Desmarais, M. C., & d. Baker, R. S. J. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction, 22*(1-2), 9-38, doi:https://doi.org/10.1007/s11257-011-9106-8.

Dimitrova, V. (2003). STyLE-OLM: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education, 13*(1), 35-78.

Dimitrova, V., & Brna, P. (2016). From interactive open learner modelling to intelligent mentoring: STyLE-OLM and beyond. *International Journal of Artificial Intelligence in Education, 26*(1), 332-349, doi:https://doi.org/10.1007/s40593-015-0087-3.

Dimitrova, V., Self, J., & Brna, P. (2001). Applying interactive open learner models to learning technical terminology. In M. Bauer, P. J. Gmytrasiewicz, & J. Vassileva (Eds.), *Proceedings of the 8th International Conference on User Modeling, UM 2001, Sonthofen, Germany, July 13-17 2001* (Vol. 2109, pp. 148-157, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/3-540-44566-8_15.

Doroudi, S., & Brunskill, E. (2019). Fairer but not fair enough: On the equitability of knowledge tracing. In J. Cunningham, N. Hoover, S. Hsiao, G. Lynch, K. McCarthy, C. Brooks, et al. (Eds.), *Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK '19, Tempe, Arizona, USA, March 4-8 2019* (pp. 335-339). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/3303772.3303838.

Duan, D., Mitrovic, A., & Churcher, N. (2010). Evaluating the effectiveness of multiple open student models in EER-Tutor. In S. L. Wong (Ed.), *Proceedings of the 18th International Conference on Computers in Education, ICCE 2010, Putrajaya, Malaysia, Nov. 29 - Dec. 3 2010* (pp. 86-88). http://hdl.handle.net/10092/5052. Accessed 25 Sept. 2020.

Dzikovska, M., Moore, J. D., Steinhauser, N., & Campbell, G. (2010). The impact of interpretation problems on tutorial dialogue. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, Short Papers, Uppsala, Sweden, July 11-16 2010* (pp. 43-48): Association for Computational Linguistics. doi:https://dl.acm.org/doi/10.5555/1858842.1858851.

Dzikovska, M., Steinhauser, N., Farrow, E., Moore, J., & Campbell, G. (2014). BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence in Education, 24*(3), 284-332, doi:https://doi.org/10.1007/s40593-014-0017-9.

Epp, C. D., & Bull, S. (2015). Uncertainty representation in visualizations of learning analytics for learners: Current approaches and opportunities. *IEEE Transactions on Learning Technologies, 8*(3), 242-260, doi:10.1109/TLT.2015.2411604.

Epp, C. D., Perez, R., Phirangee, K., Hewitt, J., & Toope, K. (2019). *User-centered dashboard design: Iterative design to support teacher informational needs in online learning contexts.* Paper presented at the American Educational Research Association (AERA) Annual Meeting, Toronto, Canada, April 5-9.

Evens, M., & Michael, J. (2006). *One-on-one Tutoring by Humans and Computers*. Mahwah, NJ USA: Lawrence Erlbaum Associates, Inc.

Ferreira, H. N. M., Araújo, R. D., Dorça, F. A., & Cattelan, R. G. (2017). Open student modeling for academic performance visualization in ubiquitous learning environments. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017, Banff, Canada, Oct. 5-8 2017* (pp. 641-646): IEEE. doi:10.1109/SMC.2017.8122679.

Forbes-Riley, K., & Litman, D. (2011). Designing and evaluating a wizarded uncertainty-adaptive spoken dialogue tutoring system. *Computer Speech & Language, 25*(1), 105-126, doi:https://doi.org/10.1016/j.csl.2009.12.002.

Fox, B. A. (1991). Cognitive and interactional aspects of correction in tutoring. In P. Goodyear (Ed.), *Teaching Knowledge and Intelligent Tutoring* (pp. 149–172): Ablex Publishing.

Gavelek, J. R., & Raphael, T. E. (1985). Metacognition, instruction, and the role of questioning activities. In D. L. Forrest-Pressley, G. E. MacKinnon, & T. G. Waller (Eds.), *Metacognition, Cognition, and Human Performance* (Vol. 2, pp. 103-136, Instructional Practices). Orlando, FL, USA: Academic Press, Inc.

Ginon, B., Boscolo, C., Johnson, M. D., & Bull, S. (2016). Persuading an open learner model in the context of a university course: An exploratory study. In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), *Proceedings of the 13th International Conference on Intelligent Tutoring Systems, ITS 2016, Zagreb, Croatia, June 6-10 2016* (Vol. 9684, pp. 307-313, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-39583-8_34.

Girard, S. (2011). *Traffic Lights and Smiley Faces: Do Children Learn Mathematics Better with Affective Open-Learner Modelling Tutors?* PhD Dissertation, University of Bath, Bath, UK.

Girard, S., & Johnson, H. (2008). DividingQuest: Opening the learner model to teachers. *Computer Science Technical Report Series* (Vol. CSBU-2008-01). Bath, U. K.: University of Bath.

Gong, Y. (2014). *Student Modeling in Intelligent Tutoring Systems*. PhD Dissertation, Worcester Polytechnic Institute, Worcester, MA, USA.

Graesser, A. C. (2016). Conversations with AutoTutor help students learn. *International Journal of Artificial Intelligence in Education, 26*(1), 124-132, doi:https://doi.org/10.1007/s40593-015-0086-4.

Graesser, A. C., Cai, Z., Morgan, B., & Wang, L. (2017a). Assessment with computer agents that engage in conversational dialogues and trialogues with learners. *Computers in Human Behavior, 76*, 607-616, doi:https://doi.org/10.1016/j.chb.2017.03.041.

Graesser, A. C., D'Mello, S. K., & Strain, A. C. (2014). Emotions in advanced learning technologies. In R. Pekrun, & L. Linnenbrink-Garcia (Eds.), *International Handbook of Emotions in Education* (pp. 473-493, Educational Psychology Handbook Series). New York, NY, USA: Routledge/Taylor & Francis Group.

Graesser, A. C., Jackson, G. T., Matthews, E. C., Mitchell, H. H., Olney, A., Ventura, M., et al. (2003). Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog. In R. Alterman, & D. Kirsh (Eds.), *Proceedings of the 25th Annual Meeting of the Cognitive Science Society, Boston, MA, USA, July 31 - Aug. 2 2003* (Vol. 25): Cognitive Science Society. https://www.academia.edu/33989380/Why_AutoTutor_A_test_of_learning_gains_from_a_physics_tutor_with_natural_language_dialog. Accessed 25 Sept. 2020.

Graesser, A. C., Lippert, A. M., & Hampton, A. J. (2017b). Successes and failures in building learning environments to promote deep learning: The value of conversational agents. In J. Buder, & F. Hesse (Eds.), *Informational Environments* (pp. 273-298). Cham: Springer.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., et al. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers, 36*(2), 180-192, doi:https://doi.org/10.3758/BF03195563.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal, 31*(1), 104-137, doi:https://doi.org/10.3102/00028312031001104.

Greer, J. E., & McCalla, G. I. (1991). Student modelling: The key to individualized knowledge-based instruction. In J. E. Greer, & G. I. McCalla (Eds.), *Proceedings of the NATO Advanced Research Workshop on "Student Modelling: The Key to Individualized Knowledge-Based Instruction", Ste. Adele, Quebec, Canada, May 4-8 1994* (Vol. 125). Berlin, Heidelberg: Springer-Verlag. doi:https://link.springer.com/book/10.1007/978-3-662-03037-0.

Grigoriadou, M., Papanikolaou, K., Tsaganou, G., Gouli, E., & Gogoulou, A. (2010). Introducing innovative e-learning environments in higher education. *International Journal of Continuing Engineering Education and Life-Long Learning, 20*(3-5), 337-355, doi:10.1504/IJCEELL.2010.037050.

Gunasekara, R. C., Nahamoo, D., Polymenakos, L. C., Ciaurri, D. E., Ganhotra, J., & Fadnis, K. P. (2019). Quantized Dialog—A general approach for conversational systems. *Computer Speech & Language, 54*, 17-30, doi:https://doi.org/10.1016/j.csl.2018.06.003.

Halloun, I. A., & Hestenes, D. (1985). The initial knowledge state of college physics students. *American Journal of Physics, 53*(11), 1043-1055, doi:https://doi.org/10.1119/1.14030.

Holstein, K., Hong, G., Tegene, M., McLaren, B. M., & Aleven, V. (2018a). The classroom as a dashboard: Co-designing wearable cognitive augmentation for K-12 teachers. In S. B. Shum, R. Ferguson, A. M. Merceron, & X. Ochoa (Eds.), *Proceedings of the 8th International Conference on Learning Analytics and Knowledge, LAK '18, Sydney, New South Wales, Australia, March 3-9 2018* (pp. 79-88). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/3170358.3170377.

Holstein, K., McLaren, B. M., & Aleven, V. (2017a). Intelligent tutors as teachers' aides: Exploring teacher needs for real-time analytics in blended classrooms. In X. Ochoa, I. Molenaar, & S. Dawson (Eds.), *Proceedings of the 7th International Learning Analytics & Knowledge Conference, LAK '17, Vancouver, British Columbia, Canada, March 13-17 2017* (pp. 257-266). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/3027385.3027451.

Holstein, K., McLaren, B. M., & Aleven, V. (2017b). SPACLE: Investigating learning across virtual and physical spaces using spatial replays. In X. Ochoa, I. Molenaar, & S. Dawson (Eds.), *Proceedings of the 7th International Learning Analytics & Knowledge Conference, LAK '17, Vancouver,* *British Columbia, Canada, March 3-9 2017* (pp. 358-367). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/3027385.3027450.

Holstein, K., McLaren, B. M., & Aleven, V. (2018b). Informing the design of teacher awareness tools through causal alignment analysis. In J. Kay, & R. Luckin (Eds.), *Proceedings of the 13th International Conference of the Learning Sciences, ICLS '18, London, UK, June 23-27 2018* (Vol. 1, pp. 104-111): International Society of the Learning Sciences, Inc. http://kenholstein.com/ICLS18_CAA.pdf. Accessed 25 Sept. 2020.

Holstein, K., McLaren, B. M., & Aleven, V. (2018c). Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, et al. (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence in Education, AIED 2018, London, UK, June 27-30 2018* (Vol. 10947, pp. 154-168, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-93843-1.

Holstein, K., McLaren, B. M., & Aleven, V. (2019). Co-designing a real-time classroom orchestration tool to support teacher–AI complementarity. *Journal of Learning Analytics, 6*(2), 27–52, doi:https://doi.org/10.18608/jla.2019.62.3.

Holstein, K., Xhakaj, F., Aleven, V., & McLaren, B. M. (2010). Luna: A dashboard for teachers using intelligent tutoring systems. *Education, 60*(1), 159-171.

Hothorn, T., & Everitt, B. S. (2014). *A Handbook of Statistical Analyses Using R* (3rd ed.). Boca Raton, FL: Chapman and Hall/CRC.

Hsiao, I.-H., & Brusilovsky, P. (2017). Guiding and motivating students through open social student modeling: Lessons learned. *Teachers College Record, 119*(3).

Hume, G., Michael, J., Rovick, A., & Evens, M. (1996). Hinting as a tactic in one-on-one tutoring. *The Journal of the Learning Sciences, 5*(1), 23-47, doi:https://doi.org/10.1207/s15327809jls0501_2.

Jackson, G. T., Dempsey, K. B., & McNamara, D. S. (2012). Game-based practice in a reading strategy tutoring system: Showdown in iSTART-ME. In H. Reinders (Ed.), *Digital Games in Language Learning and Teaching* (pp. 115-138, The New Language Learning and Teaching Environments (NLLTE) Series). London: Palgrave Macmillan.

Jordan, P., Albacete, P., & Katz, S. (2015a). Exploring the effects of redundancy within a tutorial dialogue system: Restating students' responses. In A. Koller, G. Skantze, F. Jurcicek, M. Araki, & C. P. Rosé (Eds.), *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, Sept. 2-4 2015*. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.18653/v1/W15-46.

Jordan, P., Albacete, P., & Katz, S. (2015b). When is it helpful to restate student responses within a tutorial dialogue system? In C. Conati, N. Heffernan, A. Mitrovic, & M. Verdejo (Eds.), *Proceedings of the 17th International Conference on Artificial Intelligence in Education, AIED 2015, Madrid, Spain, June 22-26 2015* (Vol. 9112, pp. 658-661, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-19773-9_85.

Jordan, P., Albacete, P., & Katz, S. (2016). Exploring contingent step decomposition in a tutorial dialogue system. In L. Aroyo, & S. D'Mello (Eds.), *Extended Proceedings of the 24th Conference on User Modeling, Adaptation and Personalization (UMAP 2016): Late Breaking Results, Halifax, Nova Scotia, Canada, July 13-17 2016*. New York, NY, USA: Association for Computing Machinery. doi:https://dl.acm.org/doi/proceedings/10.1145/2930238.

Jordan, P., Albacete, P., & Katz, S. (2017). Adapting step granularity in tutorial dialogue based on pretest scores. In E. André, R. S. J. d. Baker, X. Hu, M. Rodrigo, & B. du Boulay (Eds.), *Proceedings of the 18th International Conference on Artificial Intelligence in Education, AIED 2017, Wuhan, China, June 28 - July 1 2017* (Vol. 10331, pp. 137-148, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-61425-0_12.

Jordan, P., Albacete, P., & Katz, S. (2018). A comparison of tutoring strategies for recovering from a failed attempt during faded support. In C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, et al. (Eds.), *Proceedings of the 19th International*

Conference on Artificial Intelligence in Education, AIED 2018, London, UK, July 27-30 2018 (Vol. 10947, pp. 212-224, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-93843-1_16.

Jordan, P., Katz, S., Albacete, P., Ford, M., & Wilson, C. (2012). Reformulating student contributions in tutorial dialogue. In B. Di Eugenio, & S. McRoy (Eds.), *Proceedings of the 7th International Natural Language Generation Conference, INLG 2012, Utica, IL, USA, May 30 - June 1 2012* (pp. 95-99). Stroudsburg, PA, USA: Association for Computational Linguistics. https://www.aclweb.org/anthology/W12-1500. Accessed 25 Sept. 2020.

Jordan, P., Ringenberg, M., & Hall, B. (2006). Rapidly developing dialogue systems that support learning studies. In *Proceedings of the ITS 2006 Workshop on Teaching with Robots, Agents, and NLP, Jhongli, Taiwan, June 26-30 2006* (pp. 1-8). https://www.learnlab.org/uploads/mypslc/publications/overview.pdf. Accessed 25 Sept. 2020.

Katz, S., & Albacete, P. (2013). A tutoring system that simulates the highly interactive nature of human tutoring. *Journal of Educational Psychology, 105*(4), 1126 - 1141, doi:https://doi.apa.org/doiLanding?doi=10.1037%2Fa0032063.

Katz, S., Albacete, P., & Jordan, P. (2016). Do summaries support learning from post-problem reflective dialogues? In A. Micarelli, J. C. Stamper, & K. Panourgia (Eds.), *Proceedings of the 13th International Conference on Intelligent Tutoring Systems, ITS 2016, Zagreb, Croatia, June 7-10 2016* (Vol. 9684, pp. 519-520, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer-Verlag. doi:https://dl.acm.org/doi/proceedings/10.5555/2959507.

Katz, S., Albacete, P., Jordan, P., Lusetich, D., Chounta, I. A., & McLaren, B. M. (2018). Operationalizing contingent tutoring in a natural-language dialogue system. In S. Craig (Ed.), *Tutoring and Intelligent Tutoring Systems* (pp. 187-220). New York: Nova Science Publishers.

Kerly, A., & Bull, S. (2008). Children's interactions with inspectable and negotiated learner models. In B. P. Woolf, E. Aïmeur, R. Nkambou, & S. Lajoie (Eds.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems, ITS 2008, Montreal, QC, Canada, June 23-27 2008* (Vol. 5091, pp. 132-141, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-540-69132-7_18.

Kerly, A., Ellis, R., & Bull, S. (2008). CALMsystem: A conversational agent for learner modelling. *Knowledge-Based Systems, 21*(3), 238-246, doi:https://doi.org/10.1016/j.knosys.2007.11.015.

Kerly, A., Hall, P., & Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems, 20*(2), 177-185, doi:https://doi.org/10.1016/j.knosys.2006.11.014.

King, A. (1990). Enhancing peer interaction and learning in the classroom through reciprocal questioning. *American Educational Research Journal, 27*(4), 664-687, doi:https://doi.org/10.3102/00028312027004664.

King, A. (1994). Guiding knowledge construction in the classroom: Effects of teaching children how to question and how to explain. *American Educational Research Journal, 31*(2), 338-368, doi:https://doi.org/10.3102/00028312031002338.

Koedinger, K. R., & Corbett, A. (2006). Cognitive tutors: Technology bringing learning sciences to the classroom. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (Cambridge Handbooks in Psychology): Cambridge University Press.

Kopp, K. J., Britt, M. A., Millis, K., & Graesser, A. C. (2012). Improving the efficiency of dialogue in tutoring. *Learning and Instruction, 22*(5), 320-330, doi:10.1016/j.learninstruc.2011.12.002.

Lane, H. C., & VanLehn, K. (2005). Teaching the tacit knowledge of programming to novices with natural language tutoring. *Computer Science Education, 15*(3), 183-201, doi:10.1080/08993400500224286.

Litman, D., & Forbes-Riley, K. (2006). Correlations between dialogue acts and learning in spoken tutoring dialogues. *Natural Language Engineering, 12*(2), 161-176, doi:https://doi.org/10.1017/S1351324906004165.

Long, Y., & Aleven, V. (2017). Enhancing learning outcomes through self-regulated learning support with an open learner model. *User Modeling and User-Adapted Interaction, 27*(1), 55-88, doi:https://doi.org/10.1007/s11257-016-9186-6.

MacLellan, C. J., Liu, R., & Koedinger, K. R. (2015). Accounting for slipping and other false negatives in logistic models of student learning. In O. C. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, et al. (Eds.), *Proceedings of the 8th International Educational Data Mining Society Conference, EDM 2015, Madrid, Spain, June 26-29 2015* (pp. 53-60): International Educational Data Mining Society (IEDMS). https://files.eric.ed.gov/fulltext/ED560503.pdf. Accessed 25 Sept. 2020.

McCarthy, K. S., Watanabe, M., Dai, J., & McNamara, D. S. (2020). Personalized learning in iSTART: Past modifications and future design. *Journal of Research on Technology in Education, 52*(3), 301-321, doi:https://doi.org/10.1080/15391523.2020.1716201.

McLaren, B. M., van Gog, T., Ganoe, C., Karabinos, M., & Yaron, D. (2016). The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior, 55*(Part A), 87-99, doi:https://doi.org/10.1016/j.chb.2015.08.038.

McNamara, D. S., O'Reilly, T., Rowe, M., Boonthum, C., & Levinstein, I. B. (2007). iSTART: A web-based tutor that teaches self-explanation and metacognitive reading strategies. In D. S. McNamara (Ed.), *Reading Comprehension Strategies: Theories, Interventions, and Technologies* (pp. 397-420). Mahwah, NJ, USA: Lawrence Erlbaum Associates.

Mestre, J. P., Ross, B. H., Brookes, D. T., Smith, A. D., & Nokes, T. J. (2009). How cognitive science can promote conceptual understanding in physics classrooms. In I. M. Saleh, & M. S. Khine (Eds.), *Fostering Scientific Habits of Mind: Pedagogical Knowledge and Best Practices in Science Education* (Vol. 3, pp. 145-171, Contemporary Approaches to Research in Learning Innovations Series). Rotterdam, The Netherlands: Sense Publishers.

Mitrovic, A. (2012). Fifteen years of constraint-based tutors: What we have achieved and where we are going. *User Modeling and User-Adapted Interaction, 22*(1-2), 39-72, doi:https://doi.org/10.1007/s11257-011-9105-9.

Olney, A., D'Mello, S., Person, N., Cade, W., Hays, P., Williams, C., et al. (2012). Guru: A computer tutor that models expert human tutors. In S. A. Cerri, W. J. Clancey, G. Papadourakis, & K. Panourgia (Eds.), *Proceedings of the 11th International Conference on Intelligent Tutoring Systems, ITS 2012, Chania, Crete, Greece, June 14-18 2012* (Vol. 7315, pp. 256-261, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-642-30950-2_32.

Olney, A., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. C. (2003). Utterance classification in AutoTutor. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing, 2003* (Vol. 2, pp. 1-8). Stroudsburgh, PA, USA: Association for Computational Linguistics. doi:https://doi.org/10.3115/1118894.1118895.

Palincsar, A. S. (1998). Keeping the metaphor of scaffolding fresh—A response to C. Addison Stone's "The metaphor of scaffolding: Its utility for the field of learning disabilities". *Journal of Learning Disabilities, 31*(4), 370-373, doi:https://doi.org/10.1177/002221949803100406.

Pavlik, P. I., Brawner, K., Olney, A., & Mitrovic, A. (2013). A review of student models used in intelligent tutoring systems. In R. A. Sottilare, A. C. Graesser, X. Hu, & H. Holden (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Volume 1—Learner Modeling* (pp. 39-68). Orlando, FL, USA: US Army Research Laboratory.

Pérez-Marín, D., & Pascual-Nieto, I. (2010). Showing automatically generated students' conceptual models to students and teachers. *International Journal of Artificial Intelligence in Education, 20*(1), 47-72, doi:10.3233/JAI-2010-0002.

Pino-Pasternak, D., Whitebread, D., & Tolmie, A. (2010). A multidimensional analysis of parent–child interactions during academic tasks and their relationships with children's self-regulated learning. *Cognition and Instruction, 28*(3), 219-272, doi:https://doi.org/10.1080/07370008.2010.490494.

Pratt, M. W., Green, D., MacVicar, J., & Bountrogianni, M. (1992). The mathematical parent: Parental scaffolding, parenting style, and learning outcomes in long-division mathematics homework. *Journal of Applied Developmental Psychology, 13*(1), 17-34, doi:https://doi.org/10.1016/0193-3973(92)90003-Z.

Pratt, M. W., Kerig, P., Cowan, P. A., & Cowan, C. P. (1988). Mothers and fathers teaching 3-year-olds: Authoritative parenting and adult scaffolding of young children's learning. *Developmental Psychology, 24*(6), 832-839, doi:https://doi.org/10.1037/0012-1649.24.6.832.

Pratt, M. W., & Savoy-Levine, K. M. (1998). Contingent tutoring of long-division skills in fourth and fifth graders: Experimental tests of some hypotheses about scaffolding. *Journal of Applied Developmental Psychology, 19*(2), 287-304, doi:https://doi.org/10.1016/S0193-3973(99)80041-0.

Reber, A., Rhiannon, A., & Reber, E. (2009). *Penguin Dictionary of Psychology* (4th ed., Penguin Reference Library): Penguin Press.

Riofrío-Luzcando, D., Ramírez, J., Moral, C., de Antonio, A., & Berrocal-Lobo, M. (2019). Visualizing a collective student model for procedural training environments. *Multimedia Tools and Applications, 78*(8), 10983-11010, doi:https://doi.org/10.1007/s11042-018-6641-x.

Rodgers, E. (2017). Scaffolding word solving while reading: New research insights. *The Reading Teacher, 70*(5), 525-532, doi:https://doi.org/10.1002/trtr.1548.

Rodgers, E., D'Agostino, J. V., Harmey, S. J., Kelly, R. H., & Brownfield, K. (2016). Examining the nature of scaffolding in an early literacy intervention. *Reading Research Quarterly, 51*(3), 345-360, doi:https://doi.org/10.1002/rrq.142.

Rosé, C. P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). Interactive conceptual tutoring in Atlas-Andes. In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), *Proceedings of the 10th International Conference on Artificial Intelligence in Education, AIED 2001, San Antonio, TX, USA, May 19-23 2001* (Vol. 68, pp. 256-266, Frontiers in Artificial Intelligence and Applications): IOS Press. https://www.semanticscholar.org/paper/Interactive-Conceptual-Tutoring-in-Atlas-Andes-Ros-h-eedman/65c14e3e04a9438142fb0a0dbf9647f1b6d11863. Accessed 25 Sept. 2020.

Rosé, C. P., Kumar, R., Aleven, V., Robinson, A., & Wu, C. (2006). CycleTalk: Data driven design of support for simulation based learning. *International Journal of Artificial Intelligence in Education, 16*(2), 195-223. https://psycnet.apa.org/record/2006-10773-006. Accessed 25 Sept. 2020.

Rueda, U., Arruarte, A., & Elorriaga, J. A. (2007). A visual concept mapping medium to open student and group models. In E. Hamilton, & A. Hurford (Eds.), *Supplementary Proceedings of the 13th International Conference of Artificial Intelligence in Education (AIED 2007): Workshop on Assessment of Group and Individual Learning Through Intelligent Visualization (AGILEeViz), Marina del Rey, California, USA, July 9-13 2007* (pp. 4-7). https://www.academia.edu/26781777/Assessment_of_Group_and_Individual_Learning_through_Intelligent_Visualization_Workshop_AGILeViz_. Accessed 25 Sept. 2020.

Rueda, U., Larrañaga, M., Ferrero, B., Arruarte, A., & Elorriaga, J. (2003). Study of graphical issues in a tool for dynamically visualising student models. In *Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED 2003): Workshop on Learner Modelling for Reflection., Sydney, Australia, July 20-24 2003*.

Rus, V., Conley, M., & Graesser, A. C. (2014a). The DENDROGRAM model of instruction: On instructional strategies and their implementation in DeepTutor. In R. A. Sottilare, A. Graesser, X. Hu, & B. Goldberg (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Instructional Management* (Vol. 2, pp. 311-325, Adaptive Tutoring Series). Orlando, FL, USA: US Army Research Lab.

Rus, V., D'Mello, S., Hu, X., & Graesser, A. C. (2013a). Recent advances in conversational intelligent tutoring systems. *AI Magazine, 34*(3), 42-54, doi:https://doi.org/10.1609/aimag.v34i3.2485.

Rus, V., Niraula, N., Lintean, M., Banjade, R., Stefanescu, D., & Baggett, W. (2013b). Recommendations for the generalized intelligent framework for tutoring based on the development of the DeepTutor tutoring service. In *Proceedings of the User Meeting on Generalized Intelligent Framework for Tutoring (GIFT), in conjunction with the 16th International Conference on Artificial Intelligence in Education (AIED 2013), Memphis, TN, USA, 2013* (pp. 9-13).

Rus, V., Niraula, N. B., & Banjade, R. (2015). DeepTutor: An effective, online intelligent tutoring system that promotes deep learning. In B. Bonet, & S. Koenig (Eds.), *Proceedings of the 29th AAAI Conference on Artificial Intelligence, AIED 2015, Austin, TX, USA, Jan. 25-29 2015* (pp. 4294–4295): AAAI Press.

Rus, V., Stefanescu, D., Baggett, W., Niraula, N., Franceschetti, D., & Graesser, A. C. (2014b). Macro-adaptation in conversational intelligent tutoring matters. In S. Trausan-Matu, K. E. Boyer, M. Crosby, & K. Panourgia (Eds.), *Proceedings of the 12th International Conference on Intelligent Tutoring Systems, ITS 2014, Honolulu, HI, USA, June 5-9 2014* (Vol. 8474, pp. 242-247, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-07221-0_29.

San Martin, M. G. (2018). Scaffolding the learning-to-teach process: A study in an ESL teacher education programme in Argentina. *Profile: Issues in Teachers' Professional Development, 20*(1), 121-134, doi:https://doi.org/10.15446/profile.v20n1.63032.

Shahrour, G., & Bull, S. (2009). Interaction preferences and learning in an inspectable learner model for language. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. C. Graesser (Eds.), *Proceedings of the 14th Conference on Artificial Intelligence in Education, AIED 2009, Brighton, UK, July 6-10 2009* (Vol. 200, pp. 659-661). Amsterdam, Netherlands: IOS Press. doi:10.3233/978-1-60750-028-5-659.

Shute, V. J. (1995). SMART: Student modeling approach for responsive tutoring. *User Modeling and User-Adapted Interaction, 5*, 1-44, doi:https://doi.org/10.1007/BF01101800.

Suleman, R. M., Mizoguchi, R., & Ikeda, M. (2016). A new perspective of negotiation-based dialog to enhance metacognitive skills in the context of open learner models. *International Journal of Artificial Intelligence in Education, 26*(4), 1069-1115, doi:https://doi.org/10.1007/s40593-016-0118-8.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*(1), 59-89, doi:https://doi.org/10.1207/s1532690xci0201_3.

Tchetagni, J., Nkambou, R., & Bourdeau, J. (2007). Explicit reflection in Prolog-Tutor. *International Journal of Artificial Intelligence in Education, 17*(2), 169-215. https://telearn.archives-ouvertes.fr/hal-00190040. Accessed 25 Sept. 2020.

Thomson, D., & Mitrovic, A. (2010). Preliminary evaluation of a negotiable student model in a constraint-based ITS. *Research and Practice in Technology Enhanced Learning, 5*(01), 19-33, doi:https://doi.org/10.1142/S1793206810000797.

Tsiriga, V., & Virvou, M. (2002). Initializing the student model using stereotypes and machine learning. In P. Borne (Ed.), *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, Yasmine Hammamet, Tunisia, Oct. 6-9 2002* (Vol. 2, IEEE Xplore): IEEE. doi:10.1109/ICSMC.2002.1173446.

van de Pol, J., Mercer, N., & Volman, M. (2019). Scaffolding student understanding in small-group work: Students' uptake of teacher support in subsequent small-group interaction. *Journal of the Learning Sciences, 28*(2), 206-239, doi:https://doi.org/10.1080/10508406.2018.1522258.

van de Pol, J., Volman, M., & Beishuizen, J. (2010). Scaffolding in teacher–student interaction: A decade of research. *Educational Psychology Review, 22*(3), 271-296, doi:https://doi.org/10.1007/s10648-010-9127-6.

van de Pol, J., Volman, M., Oort, F., & Beishuizen, J. (2014). Teacher scaffolding in small-group work: An intervention study. *Journal of the Learning Sciences, 23*(4), 600-650, doi:https://doi.org/10.1080/10508406.2013.805300.

van de Pol, J., Volman, M., Oort, F., & Beishuizen, J. (2015). The effects of scaffolding in the classroom: Support contingency and student independent working time in relation to student achievement, task effort and appreciation of support. *Instructional Science, 43*(5), 615-641, doi:https://doi.org/10.1007/s11251-015-9351-z.

van Gog, T., Paas, F., & van Merriënboer, J. J. G. (2006). Effects of process-oriented worked examples on troubleshooting transfer performance. *Learning and Instruction, 16*(2), 154-164, doi:https://doi.org/10.1016/j.learninstruc.2006.02.003.

Van Labeke, N., Brna, P., & Morales, R. (2007). Opening up the interpretation process in an open learner model. *International Journal of Artificial Intelligence in Education, 17*(3), 305-338.

VanLehn, K. (1988). Student modeling. In M. C. Polson, & J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems* (Vol. 55, pp. 55-78). Hillsdale, NJ, USA: Lawrence Erlbaum Associates, Inc.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265, doi:https://dl.acm.org/doi/10.5555/1435351.1435353.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*(1), 3-62, doi:https://doi.org/10.1080/03640210709336984.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Mental Process*. Cambridge, MA, USA: Harvard University Press.

Ward, A., & Litman, D. (2011). Adding abstractive reflection to a tutorial dialog system. In P. McCarthy, & C. Murray (Eds.), *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS-24, Palm Beach, FL, USA, May 18-20 2011*: AAAI Press. doi:https://dblp.org/db/conf/flairs/flairs2011.html.

Weerasinghe, A., Mitrovic, A., & Martin, B. (2009). Towards individualized dialogue support for ill-defined domains. *International Journal of Artificial Intelligence in Education, 19*(4), 357-379.

Weerasinghe, A., Mitrovic, A., Thomson, D., Mogin, P., & Martin, B. (2011). Evaluating a general model of adaptive tutorial dialogues. In G. Biswas, S. Bull, J. Kay, & A. Mitrović (Eds.), *Proceedings of the 15th International Conference on Artificial Intelligence in Education, AIED 2011, Auckland, New Zealand, 2011* (Vol. 6738, pp. 394-402, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer-Verlag. doi:https://dl.acm.org/doi/10.5555/2026506.2026559.

Weerasinghe, A., Mitrovic, A., Van Zijl, M., & Martin, B. (2010). Evaluating the effectiveness of adaptive tutorial dialogues in database design. In *Proceedings of the 18th International Conference on Computers in Education, ICCE 2010, Putrajaya, Malaysia, Nov. 29 - Dec. 3 2010* (pp. 33-40). http://hdl.handle.net/10092/5050. Accessed 25 Sept. 2020.

Wood, D. (2001). Scaffolding, contingent tutoring, and computer-supported learning. *International Journal of Artificial Intelligence in Education, 12*(3), 280-292.

Wood, D. (2003). The why? what? when? and how? of tutoring: The development of helping and tutoring skills in children. *Literacy, Teaching and Learning, 7*(1/2), 1-30. https://eric.ed.gov/?id=EJ966143. Accessed 25 Sept. 2020.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89-100, doi:https://doi.org/10.1111/j.1469-7610.1976.tb00381.x.

Wood, D., & Middleton, D. (1975). A study of assisted problem-solving. *British Journal of Psychology, 66*(2), 181-191, doi:https://doi.org/10.1111/j.2044-8295.1975.tb01454.x.

Wood, D., Wood, H., & Middleton, D. (1978). An experimental evaluation of four face-to-face teaching strategies. *International Journal of Behavioral Development, 1*(2), 131-147, doi:https://doi.org/10.1177/016502547800100203.

Xhakaj, F., Aleven, V., & McLaren, B. M. (2016). How teachers use data to help students learn: Contextual inquiry for the design of a dashboard. In K. Verbert, M. Sharples, & T. Klobučar (Eds.), *Proceedings of 11th European Conference on Technology Enhanced Learning (EC-TEL 2016): Adaptive and Adaptable Learning, Lyon, France, Sept. 13-16 2016* (Vol. 9891, pp. 340-354, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-319-45153-4_26.

Xhakaj, F., Aleven, V., & McLaren, B. M. (2017a). Effects of a teacher dashboard for an intelligent tutoring system on teacher knowledge, lesson planning, lessons and student learning. In É. Lavoué, H. Drachsler, K. Verbert, J. Broisin, & M. Pérez-Sanagustín (Eds.), *Proceedings of the 12th European Conference on Technology Enhanced Learning (EC-TEL 2017): Data Driven Approaches in Digital Education, Tallinn, Estonia, Sept. 12-15 2017* (Vol. 10474, pp. 582-585, Lecture Notes in Computer Science). Cham: Springer. doi:10.1007/978-3-319-66610-5_23.

Xhakaj, F., Aleven, V., & McLaren, B. M. (2017b). Effects of a teacher dashboard for an intelligent tutoring system on teacher knowledge, lesson plans and class sessions. In E. André, R. S. J. d. Baker, X. Hu, M. M. T. Rodrigo, & B. du Boulay (Eds.), *Proceedings of the 18th International Conference on Artificial Intelligence in Education, AIED 2017, 2017* (Vol. 10331, pp. 582-585, Lecture Notes in Computer Science). Cham: Springer. doi:10.1007/978-3-319-66610-5_23.

Yacef, K. (2005). The Logic-ITA in the classroom: A medium scale experiment. *International Journal of Artificial Intelligence in Education, 15*(1), 41-62. https://telearn.archives-ouvertes.fr/hal-00257107/document. Accessed 25 Sept. 2020.

Zapata-Rivera, D. (2019). Supporting human inspection of adaptive instructional systems. In R. Sottilare, & J. Schwarz (Eds.), *Proceedings of the 21st International Conference on Human-Computer Interaction, HCII 2019: Adaptive Instructional Systems, Orlando, FL, USA, July 26–31 2019* (Vol. 11597, pp. 482-490, Lecture Notes in Computer Science). Cham: Springer. doi:https://doi.org/10.1007/978-3-030-22341-0_38.

Zapata-Rivera, D., & Greer, J. E. (2001). Externalising learner modelling representations. In S. Ainsworth, & R. Cox (Eds.), *Proceedings of the Workshop on External Representations of AIED: Multiple Forms and Multiple Roles, in conjunction with the 10th International Conference on Artificial Intelligence in Education, AIED 2001, San Antonio, TX, USA, May 20 2001* (pp. 71-76). https://www.researchgate.net/publication/228782374_Externalising_learner_modelling_representations. Accessed 25 Sept. 2020.

Zapata-Rivera, D., & Greer, J. E. (2002). Exploring various guidance mechanisms to support interaction with inspectable learner models. In S. A. Cerri, G. Gouardères, & F. Paraguaçu (Eds.), *Proceedings of the 6th International Conference on Intelligent Tutoring Systems, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2-7 2002* (Vol. 2363, pp. 442-452, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer. doi:10.1007/3-540-47987-2_47.

Zapata-Rivera, D., & Greer, J. E. (2003). Analyzing student reflection in the learning game. In S. Bull, P. Brna, & V. Dimitrova (Eds.), *Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education, AIED 2003, Sydney, Australia, July 20-24 2003* (pp. 288-298): University of Sydney. https://books.google.com/books/about/AIED_2003.html?id=KozBNQAACAAJ. Accessed 25 Sept. 2020.

Zapata-Rivera, D., & Greer, J. E. (2004a). Inspectable Bayesian student modelling servers in multi-agent tutoring systems. *International Journal of Human-Computer Studies, 61*(4), 535-563, doi:https://doi.org/10.1016/j.ijhcs.2003.12.017.

Zapata-Rivera, D., & Greer, J. E. (2004b). Interacting with inspectable Bayesian student models. *International Journal of Artificial Intelligence in Education, 14*(2), 127-163, doi:https://dl.acm.org/doi/10.5555/1434858.1434859.

Zapata-Rivera, D., Hansen, E., Shute, V. J., Underwood, J. S., & Bauer, M. (2007). Evidence-based approach to interacting with open student models. *International Journal of Artificial Intelligence in Education, 17*(3), 273-303, doi:https://dl.acm.org/doi/10.5555/1435391.1435395.

Zapata-Rivera, D., & Katz, I. R. (2014). Keeping your audience in mind: Applying audience analysis to the design of interactive score reports. *Assessment in Education: Principles, Policy & Practice, 21*(4), 442-463, doi:https://doi.org/10.1080/0969594X.2014.936357.

Zapata-Rivera, D., Lehman, B., Sparks, J. R., Por, H.-H., & James, K. (2018). Identifying and addressing unexpected responses in conversation-based assessments. In J. Carlson (Ed.), *ETS Research Memorandum Series* (Vol. RM-18-13). Princeton, NJ, USA.

Zapata-Rivera, D., Zwick, R., & Vezzu, M. (2016). Exploring the effectiveness of a measurement error tutorial in helping teachers understand score report results. *Educational Assessment, 21*(3), 215-229, doi:https://doi.org/10.1080/10627197.2016.1202110.

Zhou, Y., & Evens, M. W. (1999). A practical student model in an intelligent tutoring system. In *Proceedings of the 11th International Conference on Tools with Artificial Intelligence, Chicago, IL, USA, Nov. 8-11 1999* (pp. 13-18): IEEE Computer Society. doi:10.1109/TAI.1999.809759.