Chapter 7 – Identifying Strategies in Student Problem Solving

Steven Ritter[1], Ryan S. Baker[2], Vasile Rus[3], and Gautam Biswas[4]
[1]Carnegie Learning, [2]University of Pennsylvania, [3]University of Memphis, [4]Vanderbilt University

Introduction

As instructional systems support more open-ended problem-solving, it becomes essential to understand the particular strategies or approaches that students take to solving problems. Increasingly, machine learning techniques to detect student strategies will be used in these instructional systems. One advantage of such self-improving systems is that, as usage broadens, they become more sophisticated in their analysis of student behavior and ability to support a wide variety of learning approaches. In this way, such systems become more educationally effective, particularly for students who employ relatively rare approaches to solving problems.

We consider a strategy to be a sequence of steps or operators taken in a problem space with the goal of accomplishing a given task or solving a problem (Newell & Simon, 1972). In theory, any variation in problem solving may represent a different strategy. In practice, however, we often group insignificant variations in problem solving steps into a single strategy and consider those that represent "significantly" different approaches to represent different strategies. Consider Figure 1. The solutions represented in Strategy A and Strategy B both consist of three similar steps. In the first step, the student subtracts a variable term from both sides of the equation (5x in Strategy A; 3x in Strategy B). In the second step, the student using Strategy A subtracts 4 from both sides, and the student using Strategy B adds 6 to both sides. In the third step, each student divides both sides of the equation by the coefficient. A student employing Strategy C combines the first two steps into a single step, subtracting 3x-6 from both sides of the equation. Clearly Strategy A and Strategy B are similar approaches and might be considered variants of a single strategy. A student who is able to recognize and execute Strategy C is illustrating a more sophisticated approach to problem-solving and might be considered to be using a strategy very different from either A or B.

3x+4 = 5x-6	3x+4 = 5x-6	3x+4 = 5x-6
-2x+4 = -6	4 = 2x-6	10 = 2x
-2x = -10	10 = 2x	5 = X
x = 5	5 = x	
Strategy A	*Strategy B*	*Strategy C*

Figure 1. Three strategies for solving an equation.

The decision about whether to consider a different sequence of steps to be a different strategy depends on the goals of the educational system, so it is important to consider the reasons why we might need to understand and distinguish different strategies. Identification of strategies may allow systems to:

- **Infer the student's level of knowledge**. Sometimes, use of a particular strategy is an indicator of the student's level of understanding. Lemaire and Siegler (1995) found that students' use of more sophisticated strategies and of better sensitivity to appropriateness of strategies changed as students learned single-digit multiplication.

- **Identify misconceptions**. Strategies are approaches to problem solving, but they need not be correct approaches. A student may, for example, incorrectly believe that numbers with more decimal places are larger than numbers with fewer decimal places (Isotani et al., 2010). In the above example, the student may have a misconception about the negation operator, and consider $3x - 5x = 2x$ (always subtract the smaller number from the large). Identification of incorrect problem-solving steps relating to this misconception can greatly aid remediation.

- **Indicate student's sensitivity to problem characteristics**. In some cases, different strategies are more appropriate for different problem types, and part of the target of learning may be that students understand how to develop strategies for different problem types, and then apply appropriate strategies when solving problems. For example, in solving simultaneous equations, substitution and linear combination are both valid strategies that can be applied to a set of equations, but, depending on the equations, one of the strategies may be easier to apply than the other. Proficient students should be flexible enough to choose the appropriate strategy for the problem.

- **Identify use of suboptimal or incorrect strategies**. Like the last bullet, this may occur when students have just learned the individual skills needed to solve problems, but have not had opportunities and practice to combine the use of these skills to solve problems. Students may combine their skills in suboptimal ways when trying to come up with a sequence of steps to solve a problem.

- **Provide opportunities for reflection and generalization**. Exposing students to multiple strategies may encourage greater procedural flexibility (Crowley & Siegler, 1999; Rittle-Johnson & Star, 2007). An educational system may wish to provide students with worked examples and problem-solving opportunities that employ different strategies. This approach would require knowledge of the strategies a particular student employs.

- **Identify metacognitive strategies**. In addition to strategies used to solve individual problems, students employ metacognitive strategies, including self-explanation and use of worked examples. Instructional systems will be more effective to the extent that systems can identify and support productive metacognitive strategies.

- **Identify conceptual understanding.** Work by Rowe and colleagues (2014, 2017) finds that behavioral strategies within learning games can be indicative of differences in conceptual understanding, correlating highly with alternate measures of those same concepts.

- **Indicate students' sensitivity to contextual characteristics.** Different strategies may require different amounts of time to execute or different resources (such as working memory load). Siegler (1988) found that some students employed different strategies under time pressure than they would apply with unlimited time. Such sensitivities may interact with student characteristics. Beilock and DeCaro (2007) found that low working memory capacity students more readily switched their strategies (to simpler heuristics) under pressure than high working memory capacity students.

Methods

In this section, we consider several methods for determining student strategies, given a stream of data describing the student's actions in a problem-solving episode.

Model Tracing

Model-tracing tutors (Corbett, 2001) monitor student activities in problem solving and attempt to map the student's actions to particular problem-solving strategies. Such actions are generally classified as belonging to a correct strategy, an incorrect strategy or an unrecognized strategy. Since strategies (both correct and incorrect) are pre-specified in the model, such systems can recognize strategies that students employ. Steps that are not recognized as belonging to a strategy are considered to be incorrect (but uninterpretable). One downside of this approach is that unique or rare strategies may not be recognized.

The strategies encoded in model-tracing tutors are typically initially discovered through cognitive task analysis (Clark & Estes, 1996; Lovett, 1998), but such systems can be extended through analysis of data collected through such systems, as described here.

In some cases, model-tracing systems can recognize cases where strategy recognition has failed and adapt to incorporation recognition of the new strategy in the future. Ritter (1997) describes one such method by which model-tracing tutors can learn to recognize new strategies. In model-tracing tutors for domains where correct solutions can be evaluated but not modelled, the tutor can recognize and provide feedback for more optimal strategies. For example, an equation solving tutor can learn more efficient strategies that are "demonstrated" by the student.

Detector-based strategy recognition

Another method for identifying student strategies is to develop a machine-learned model, often referred to as a "detector", which recognizes strategies that human beings can identify but cannot easily reduce to a straightforward set of rules, as is necessary for most uses of model tracing. This approach relies upon first obtaining human labels of when the strategy is present or absent, distilling features of the data that are reasonably likely to correspond to that strategy, and then using machine learning to train a model that can replicate the human judgments.

In the first step of this process, software is developed to display a substantial number of examples of student interaction to a set of human coders who can recognize the behavior. These displays can be presented through either a screen replay of learner behavior (e.g. Aleven et al., 2004) or a text replay (e.g. Baker, Corbett, & Wagner, 2006), pretty-printed log files. Then, typically, two coders code the same subset of the total data set and check for acceptable inter-rater reliability. After this, one or both of the coders label the remaining data.

In the second step of the process, data features thought to correspond to the strategy of interest are distilled from the data. This step, often termed feature engineering, can vary in its degree of sophistication, from a single researcher brainstorming a set by herself/himself, to a more structured brainstorming process involving multiple types of expertise, to an in-depth process of interviewing the coders and discussing draft models of their reasoning processes (Paquette et al., 2014).

In the third step of the process, off-the-shelf machine learning algorithms are used to derive a model that replicates the human judgments with reasonable reliability, and the resultant models are tested using cross-validation.

The first strategy this approach was used to study was gaming the system, when a student misuses learning software to proceed without learning. Baker and de Carvalho (2008) took data on student use of a Cognitive Tutor, distilled text replay clips of student behavior, labeled them, and built a decision tree to capture this strategy. Later work extended this approach to SQL-Tutor (Baker, Mitrovic, & Mathews, 2010) and demonstrated that better results could be obtained through a more sophisticated feature engineering process (Paquette et al., 2014). Related work investigated whether a player was seriously attempting to complete quests in an online story-based role playing game (DiCerbo & Kidwai, 2013).

Sao Pedro and his colleagues extended this approach to modeling scientific inquiry strategies, including both whether a student could design a controlled experiment across a set of trials of a simulation (Sao Pedro et al., 2013), and whether the learner planned using a table (Montalvo et al., 2010). A refined version of Sao Pedro's original detector is now used in the commercial Inq-ITS platform (Gobert et al., 2015).

Elizabeth Rowe and her colleagues extended this approach to modeling the strategies students engaged in while playing a conceptual physics game, demonstrating that it was possible to capture gameplay strategies associated with conceptual understanding of Newton's Laws (Rowe et al., 2014) and that these detectors correlated with external measures of Newton's Laws (Rowe et al., 2017).

Together, these examples demonstrate the feasibility of identifying student strategies through an automated detector machine learning approach.

Sequence Mining Methods for Strategy Detection

In the learning sciences and educational psychology research, strategies have been defined as consciously-controllable processes for completing tasks (Pressley, et al., 1989). Within this framework, it is possible to characterize strategies as a sequence of actions that a learner performs to complete a task or subtask in the learning environment. Strategies are further characterized by the context in which they are applied and the specific relations among component activities that make up a strategy. Take, for example, learning environments such as Betty's Brain, where students learn about scientific processes (e.g., climate change) by teaching a virtual agent named Betty (Leelawong & Biswas, 2008: Biswas, et al., 2016). They do this by constructing a visual causal map that represents the relevant scientific process as a set of concepts connected by directed links that represent causal relations. Once taught, Betty can use the map to answer causal questions and explain those answers. The goal for students using Betty's Brain is to teach Betty a correct causal map that matches a hidden, expert model of the domain. The students' learning and teaching tasks are organized around three activities: (1) reading hypertext resources that provide information on the science concepts and causal relations between the concepts, (2) building the causal map using a visual drag and drop interface, and (3) assessing the correctness of the map by getting the agent Betty to take quizzes and evaluating her answers.

Students strategies in this environment revolve around how they combine the three activities to accomplish higher level goals or tasks, such as acquiring information and constructing a part of the causal map (e.g., human activities that cause the greenhouse effect) or correcting errors in a section of their map (e.g., analyze Betty's quiz results, identify causal links that are related to incorrect answers, and correct the erroneous links). However, identifying students' strategies from their activity logs is not an easy task. In open-ended learning environments (OELEs), such as Betty's Brain, the fact that students have choice in the way they go about constructing their models, interpreting their strategies requires an understanding of the task that they are currently working on and an interpretation of their actions in the context of this task.

We have developed sequence mining approaches to derive frequent action sequences when they work in the Betty's Brain environment (Kinnebrew, Loretz, & Biswas, 2013; Kinnebrew, Segedy, & Biswas, 2014). In general, Sequential Pattern Mining (Agrawal & Srikant, 1995) algorithms are designed to find frequent sequential patterns, i.e., series of action that occur in many of the students' activity sequences provided (e.g., the sequential pattern A then B then C occurs in both of the sequences C → A → B → C and A → B → C → A). Researchers have applied sequence mining techniques to a variety of educational data in order to better understand learning behaviors (e.g., Amershi & Conati, 2009; Kinnebrew et al., 2013; Nesbit et al., 2007; Perera et al., 2009; Su et al., 2006; Tang & McCalla, 2002).

To extract the activity sequences of student work in Betty's Brain for sequence mining, log events captured by the learning environment abstracted student activities into a few primary categories with some additional subcategories (Kinnebrew & Biswas, 2012; Kinnebrew, et al., 2013). The primary actions extracted from the logs to generate the action sequences were:

1. *Information acquisition (IA) actions*: (a) *Read*: reading one or more of the science resource pages; and (b) *Note*: entering information into the note-taking tool provided in the system;

2. *Solution construction (SC), i.e., Edit actions*: these include operations on the causal map, with actions further divided by: (i) whether they operate on a causal link or concept and whether the action was an addition (Add), removal (Remove), or modification (Change), e.g., LinkAdd or ConceptRemove;

3. *Solution assessment (SA) actions*: (a) *Query*: students use a template to ask Betty a question, and she answers the question using a causal reasoning algorithm (Leelawong & Biswas, 2008); (b) *Quiz*: students assess how well they have taught Betty by having her take a quiz, which is a set of questions chosen and graded by the Mentor agent; and (c) *Explain*: students probe Betty's reasoning by asking her to explain her answer to a question (either from the quiz or from a query).

Strategies derived from analyzing students' logs across multiple studies with Betty's brain are discussed in a number of our papers (e.g., Kinebrew & Biswas, 2012; Kinnebrew, et al., 2013; Kinnebrew, et al., 2014; Kinnebrew, et al., 2017; Munshi, et al., 2018). We provide some illustrative examples in this chapter. For example, a study conducted in 6th grade science classrooms in 2015, showed frequent use of the *IA → SC* strategy, i.e., they read the resources and then added to or made changes in their causal map. In this case, frequent implied five or more uses of this strategy per student during the course of the intervention. When comparing the use of this strategy by high and low performers, i.e., those who had high versus low scores in their final map scores, we found that high performers used this strategy to make correct changes to their map (i.e., added a correct link or deleted an incorrect one) 62% (*SD* = 9%) percent of the time, whereas low performers made correct changes to their map only 53% (*SD* = 16%) of the time.

To better understand how students employed this general strategy, we considered two specific variants of this strategy in the Betty's Brain environment: *IA → Add a causal link* to the map and *IA → Correct a causal link* in the map by changing or removing an incorrect causal link in the map. Results indicate that *IA → Add a causal link* was used by high performing students on average 23.9 (*SD* = 16) times, and they performed this task correctly 59.3% (*SD* = 13.1%) of the time, whereas low performers performed this strategy on average only 8.9 (*SD* = 7.5) with a 48.4% (*SD* = 22.2%) correct use. Similarly, for the *IA → Correct a causal link* strategy, the numbers were 5.3 (*SD* = 4) with 75.1% (*SD* = 17.2%) correct use for the high performers, whereas the numbers were 2.9 (*SD* = 3.6) with 81.1% (*SD* = 23.5%) correct use by low performers. Though the accuracy numbers are not much different, the high performers use the strategy many more times than the low performers, thus generating better maps and better learning gains overall than the low performers. Linked to this strategy, we also found that the high performers had significantly longer *read* time per pattern than the low performers.

Another frequent pattern used by students corresponds to a debugging strategy, i.e., *SA → LinkAdd → SA* with two specific variants: (1) *Quiz Explanation → Link Add → Quiz* and *Quiz → LinkAdd → Quiz.*

These patterns suggest an informed *guess-and-check* strategy in which the quiz results (either the overall results or the information gleaned from a *Quiz Explanation* for a specific quiz question) are used to suggest a potentially-missing link, which is then added to the map. This is followed by checking the correctness of the "*guess*" by taking another quiz. As expected, for a guess-and-check strategy, the link added was usually incorrect (average percentage of correct additions per student = 19%, *SD* = 14%).

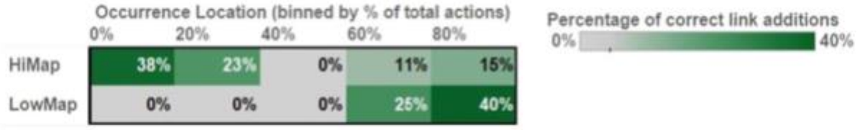

Figure 2. Heatmap of percentage correct links added in *SA* → *LinkAdd* → *SA* over time.

Given the more detailed information available from the quiz question explanation, we initially expected better performance in adding a correct link for that variant of the strategy. However, this variant actually had a marginally lower percentage of correct additions compared to the other variant. Further, analysis of effectiveness over the course of students' work in the environment illustrated that performance (correctness percentage) with *SA* → *LinkAdd* → *SA* was better early and late while being especially poor in the middle. The performance heat map shown in Figure 2, indicates that while the high performing (HiMap) students performed best with this strategy early and (to a lesser extent) late in the intervention, the low performing (LowMap) students did not make correct link additions with this strategy until relatively late (after at least 60 percent of their total actions on the system). This may imply that it took the low performing students until late in the intervention to understand how to interpret and use the quiz results. On the other hand, the high performers used this strategy with more success in the early phases of map building. The HiMap students' effectiveness with this strategy may have dropped off once they started dealing with the more difficult material (for which they had little prior knowledge) toward the middle of their activities, finally rebounding some as they gained proficiency. In addition to illustrating the importance of incorporating the overall informed guess-and-check strategy in the strategy model, analysis of this high lift pattern suggests that there may be additional interactions with prior knowledge and skills worth investigating through further experiments.

The results of combining sequence mining algorithms with additional analysis of the relations between actions, showed potentially important differences between high- and low-performing students in terms of their strategy use. Overall, an effective analysis framework applied to the rich behavioral data produced by OELEs has the potential to enable deeper analyses of students' cognitive and metacognitive behavior in complex learning tasks. Ultimately, we believe that this analysis framework can form the basis for designing richer learner modeling schemes that characterize students' activities by analyzing their learning behaviors and performance with respect to their cognitive and metacognitive processes.

Discriminant Sub-sequence Analysis from Tutorial Dialogues

A key research question in intelligent tutoring systems and in the broader instructional research community is understanding what expert tutors do (Rus, D'Mello, Hu, & Graesser, 2013). This goal is motivated by research showing that expert tutors are very effective (Bloom, 1984).

Indeed, understanding what expert tutors do has been a research goal undertaken by theoreticians and empiricist alike. A typical operationalization of this goal of understanding of what good tutors do is to define the behavior of tutors based on their actions. To this end, the learner-tutor interactions are broken down into primitive actions and then significant differences between expert tutors and less accomplished tutors are reported. For instance, Boyer and colleagues (2011) modelled the learner-tutor interaction as sequences of task actions (e.g., opening a file) and dialogue acts, i.e. actions behind utterances, while Cade and colleagues (2008) used just dialogue acts to model the learner-tutor interaction.

In a discriminant sub-sequence analysis approach (Rus, et al., 2017; Maharjan, Gautam, & Rus, 2018), tutorial dialogues are modeled as dialogue-act sequences because there are no other types of actions, e.g. task actions as in Boyer and colleagues (2011), considered in the analysis (Rus, et al., 2017; Maharjan, Gautam, & Rus, 2018). This view of a tutorial dialogue as a sequence of actions is based on the language-as-action theory (Austin, 1962; Searle, 1969). According to the language-as-action theory, when we say something we do something. Therefore, all utterances in a tutorial dialogue are mapped into corresponding dialogue acts using, in our case, a predefined dialogue or speech act taxonomy. The taxonomy was defined by educational experts and resulted in a two-level hierarchy of 17 top-level dialogue acts and a number of dialogue subacts. The exact number of subacts differs from dialogue act to dialogue act. The taxonomy identifies 129 distinct dialogue act and sub-act combinations. Further, we have a set of 17 different dialogue modes defined by experts as in the following: Assessment, Closing, Fading, ITSupport, Metacognition, MethodID, Modeling, OffTopic, Opening, ProblemID, ProcessNegotiation, RapportBuilding, RoadMap, SenseMaking, Scaffolding, SessionSummary and Telling. A detailed description of the dialogue modes is available (Morrison et al., 2015). It should be noted that automatically discovered dialogue act taxonomies are currently being built (e.g. Rus, Graesser, Moldovan, & Niraula, 2012) but it is beyond the scope of this chapter to automatically discover the dialogue acts in our tutoring sessions.

A large corpus of about 19K tutorial sessions between professional human tutors and actual college-level, adult students was collected via an online human tutoring service. Students taking two college-level developmental mathematics courses (pre-Algebra and Algebra) were offered these online human tutoring services at no cost. The same students had access to computer-based tutoring sessions through Adaptive Math Practice, a variant of Carnegie Learning' Cognitive Tutor (Ritter et al., 2007). A subset of 500 tutorial sessions containing 31,299 utterances was randomly selected from this large corpus for annotation with the requirement that a quarter of these 500 sessions would be from students who enrolled in one of the Algebra courses (Math 208), another quarter from the other course (Math 209), and half of the sessions would involve students who attended both courses.

This research investigated which distinctive subsequences of dialogues, dialogue acts and modes comprise effective and less-effective sessions. To this end, each tutorial session was rated by Subject Matter Experts (SMEs) using a 1-5 scale (5 being best score) along two dimensions: evidence of learning (EL) and evidence of soundness (ES). The ES score reflects how well tutors applied pedagogically sound tactics in tutorial sessions. On the other hand, the EL score reflects how well students learned from tutorial sessions. The EL and ES scores were found to be highly correlated (Pearson coefficient of 0.7). The research categorized all human annotated sessions having ES and EL scores less than 2 as ineffective, and all sessions rated with ES = 5 and EL = 4 as good or effective sessions.

Then sequence pattern mining was conducted using the Traminer package in R. The Traminer algorithm first finds the most frequent subsequences by counting their distinct occurrences and then applies a Chi-squared test (Bonferroni-adjusted) to identify sub-sequences that are statistically more (or less) frequent in each group. A *p*-value < 0.4 threshold was used to select likely distinctive sub-sequences, with dialogue acts, actsubacts and mode-switches used as observations. The observations were granularized further by adding speaker information.

It should be noted that a subsequence is not necessarily a contiguous sequence of observations, but the order of the observations is preserved. For example, (Assertion)-(Expressive) is a valid sub-sequence of dialogue acts formed from the (Assertion)-(Request)-(Expressive) contiguous sequence fragment. Sub-sequences were generated up to length 7 from all the annotated tutorial sessions.

The discriminant sub-sequences mined indicated that good tutors use more Expressives and prompt students more in sessions of high learning gains. It was observed that all discriminant sub-sequences of acts contain Expressive acts initiated by tutors or students. The good tutors often prompt students to confirm the students are following their tutoring or, to elicit further answers or reasoning from the students. Furthermore, the tutors' expressions of praise (T-Expressive-Positive) and farewell (T-Expressive-Farewell) and, the students expressing thanks (S-Expressive-Thanks) are highly predictive of effective sessions. The tutors often praise students to keep them engaged to the task or, when they answer correctly. The tutee expressing thanks (S-Expressing-Thanks) might suggest the tutee is satisfied with the tutoring. Moreover, the tutor expressing farewell indicates the tutoring is coming towards the end. The sessions having proper closing also might suggest that both student and tutor are satisfied with the tutorial Session.

The discriminant subsequent analysis for modes revealed interesting patterns as well when analyzed through a pedagogical lens. Good tutorial sessions have Scaffolding (S) and Fading (F) as the dominant strategies i.e. the good tutors do more Scaffolding and Fading to get the problem solved by the students themselves. The sub-sequences S-S, F, S-F, F-F are very strong indicators of good sessions (p-value<0.05) while F-S, F-F-S, S-F-S also fairly indicate the sessions of top quality. Another interesting observation is that the Closing mode (p-value=0.0475) is also a very strong indicator of top sessions. Moreover, a Fading-Closing (p-value=0.002) sub-sequence is even more predictive than Closing alone. We also observed that switching to Scaffolding or Fading modes after ProblemIdentification is more effective as evidenced by sub-sequences O-P-F (p-value=0.1764), P-F (p-value=0.0198) and P-S-S (p-value=0.0362).

Discussion

The methods described here allow us to identify differences in students' approaches to solving problems. It is important to remember that not all differences may be instructionally relevant. Within each method (or combination of methods), instructional system designers will need to make decisions about which approaches represent strategies that are indicative of different instructional needs or differing assessments of students' capabilities and knowledge.

In many cases, it is inappropriate to talk about a student's strategy as being a property of that student. The particular strategy employed may differ due to problem characteristics and also due to that student's state of knowledge. A particularly interesting case is where changes in a student's strategy over time can be considered a measure of the student's learning of the target knowledge. Corbett et al. (2000) describe one such case. Students were asked to complete a table representing the mathematical encoding of a word problem. Novice students typically started to complete the table in an order that did not take the problem structure into account: top-to-bottom and left-to right. As students came to understand the hierarchical structure of the mathematical terms underlying the problem, the order in which they filled in the cells in the table came to match the underlying structure. In cases like this, the students' strategy may be an indicator of the student's understanding of the mathematics, somewhat independent of their success in the task itself.

Recommendations and Future Research

Given the importance of understanding the particular strategies that students employ when solving problems, it is essential that such systems be able to detect and respond to different student strategies. In more open educational environments, where different solution strategies are encouraged, the ability to understand varied student strategies is even more important. In fact, widely used educational systems that support a wide variety of strategies may be the best source for data on the strategies that students use. Using the techniques outlined here, such systems may greatly benefit from becoming self-improving systems that are able to detect and react to novel student problem-solving approaches.

Within the Generalized Intelligent Framework for Tutoring (GIFT) architecture, instructional systems would benefit from recording and sharing student strategies employed while solving particular problems, in addition to the activity and evaluative information already stored. Domain models might benefit from knowledge of likely strategies, which could be used for activity selection, especially in cases where appropriate strategy choice and problem characteristics are strongly linked. Task designers should take strategy use into account. In some cases, tasks might be designed to be maximally flexible, allowing students to employ strategies that they pick. In other cases, an educational goal might be to ensure that students master more than one strategy. For this kind of goal, task designers might design multiple variants of the user interface, each of which constrains the student to employ a particular strategy.

References

Agrawal, R., & Srikant, R. (1995, March). Mining sequential patterns. In *ICDE* (p. 3). IEEE.

Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R. (2004). Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS 2004)*, 227-239.

Amershi, S., & Conati, C. (2009). Combining unsupervised and supervised classification to build user models for exploratory. JEDM| *Journal of Educational Data Mining*, *1*(1), 18-71.

Austin, J.L. (1962). *How to do Things with Words*. Oxford University Press, 1962.

Baker, R.S.J.d., Corbett, A.T., Wagner, A.Z. (2006). Human Classification of Low-Fidelity Replays of Student Actions. *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems*, 29-36.

Baker, R.S.J.d., de Carvalho, A. M. J. A. (2008). Labeling Student Behavior Faster and More Precisely with Text Replays. *Proceedings of the 1st International Conference on Educational Data Mining*, 38-47.

Baker, R.S.J.d., Mitrovic, A., Mathews, M. (2010). Detecting Gaming the System in Constraint-Based Tutors. *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization*, 267-278.

Beilock, S. & DeCaro, M. (2007). From poor performance to success under stress: Working memory, strategy selection, and mathematical problem solving under pressure. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *33*, 983-98. 10.1037/0278-7393.33.6.983.

Biswas, G., Segedy, J.R., & Bunchongchit, K. (2016). From design to implementation to practice – A learning by teaching system: Betty’s brain. *International Journal of Artificial Intelligence in Education*, *26*(1), 350-364.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4-16.

Boyer, K. E.; Phillips, R.; Ingram, A.; Ha, E. Y.; Wallis, M.; Vouk, M.; and Lester, J. 2011. Investigating the relationship between dialogue structure and tutoring effectiveness: A hidden markov modeling approach. *International Journal of Artificial Intelligence in Education*, *21*(1-2), 65–81.

Cade,W., Copeland, J., Person, N., & DMello, S. (2008). Dialogue modes in expert tutoring. In *Intelligent tutoring systems*, 470–479. Springer.

Clark, R. E., & Estes, F. (1996). Cognitive task analysis. *International Journal of Educational Research*, *25*(5), 403–417.

Corbett, A. T., McLaughlin, M., Scarpinatto, C. & Hadley, W. S. (2000). Analyzing and Generating Mathematical Models: An Algebra II Cognitive Tutor Design Study. In G. Gauthier, C. Frasson and K. VanLehn (Eds.) *Intelligent Tutoring Systems: 5th International Conference, ITS 2000*. New York: Springer, pp. 314-323.

Corbett, A.T. (2001). Cognitive computer tutors: Solving the two-sigma problem. *User Modeling: Proceedings of the Eighth International Conference, UM 2001*, 137-147.

Crowley, K., & Siegler, R. S. (1999). Explanation and generalization in young children's strategy learning. *Child Development*, *70*, 304-316.

DiCerbo, K., Kidwai, K. (2013) Detecting Player Goals from Log Files. *Proceedings of the 6th International Conference on Educational Data Mining*.

Gobert, J. D., Kim, Y. J., Sao Pedro, M. A., Kennedy, M., & Betts, C. G. (2015). Using educational data mining to assess students' skills at designing and conducting experiments within a complex systems microworld. *Thinking Skills and Creativity*, *18*, 81-90.

Isotani, S., McLaren, B.M.,& Altman, M. (2010). Towards intelligent tutoring with erroneous examples: A taxonomy of decimal misconceptions. In V. Aleven, J. Kay, J. Mostow (Eds.). *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS-10),* Lecture Notes in Computer Science, 6094 (pp. 346-348). Berlin: Springer.

Kinnebrew, J., Segedy, J.R. & Biswas, G. (2017). Integrating Model-Driven and Data-Driven Techniques for Analyzing Learning Behaviors in Open-Ended Learning Environments. *IEEE Transactions on Learning Technologies, 10*(2), 140-153.

Kinnebrew, J.S., Segedy, J.R., & Biswas, G. (2014). Analyzing the Temporal Evolution of Students' Behaviors in Open-Ended Learning Environments. *Metacognition and Learning, 9*(2), 187-215.

Kinnebrew, J.S., Loretz, K.M., & Biswas, G. (2013). A Contextualized, Differential Sequence Mining Method to Derive Students' Learning Behavior Patterns. *Journal of Educational Data Mining*, *5*(1), 190-219.

Kinnebrew, J.S., & Biswas, G. (2012). Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In *Proceedings of the 5th International Conference on Educational Data Mining*. Chania, Greece.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, *18*(3), 181-208.

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contributions to children's learning of multiplication. *Journal of Experimental Psychology: General*, *124*(1), 83-97.

Lovett, M. C. (1998). Cognitive task analysis in service of intelligent tutoring system design: a case study in statistics. In *Intelligent Tutoring Systems, Proceedings of the Fourth International Conference*. (pp. 234-243).

Maharjan, N., Gautam, D., & Rus, V. (2018). Discovering Effective Tutorial Strategies in Human Tutorial Sessions, *Proceedings of the 31st International Florida Artificial Intelligence Research Society Conference (FLAIRS 2018),* Melbourne Beach, FL, USA May 21-23, 2018.

Montalvo, O., Baker, R.S.J.d., Sao Pedro, M.A., Nakama, A., Gobert, J.D. (2010) Identifying Student' Inquiry Planning Using Machine Learning. *Proceedings of the 3rd International Conference on Educational Data Mining*, 141-150.

Morrison, D. M., Nye, B., Rus, V., Snyder, S., Boller, J., & Miller, K. (2015, June). Tutorial dialogue modes in a large corpus of online tutoring transcripts. In International Conference on Artificial Intelligence in Education (pp. 722-725). Springer, Cham.

Munshi, A., Rajendran, R., Ocumpaugh, J., Biswas, G., Baker, R. S., & Paquette, L. (2018). Modeling Learners Cognitive and Affective States to Scaffold SRL in Open Ended Learning Environments. In *Proceedings of the International Conference on User Modelling, Adaptation and Personalization* (UMAP), Singapore, pp. 131-138.

Nesbit, J. C., Zhou, M., Xu, Y., & Winne, P. (2007). Advancing log analysis of student interactions with cognitive tools. In *12th biennial conference of the European association for research on learning and instruction* (EARLI) (pp. 2-20).

Newell, A. and Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall.

Paquette, L., de Carvalho, A.M.J.A., Baker, R.S., Ocumpaugh, J. (2014) Reengineering the Feature Distillation Process: A Case Study in the Detection of Gaming the System. *Proceedings of the 7th International Conference on Educational Data Mining,* 284-287.

Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. (2009). Clustering and sequential pattern mining of online collaborative learning data. *IEEE Transactions on Knowledge and Data Engineering*, 21(6), 759-772.

Pressley, M., Goodchild, F., Fleet, J., Zajchowski, R., & Evans, E. D. (1989). The challenges of classroom strategy instruction. *The Elementary School Journal*, *89*(3), 301-342.

Ritter, S. (1997). Communication, cooperation and competition among multiple tutor agents. In du Boulay, B. and Mizoguchi, R. (Eds). *Artificial Intelligence in Education: Knowledge and media in learning systems* (pp 31-38). Amsterdam: IOS Press.

Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. (2007). The cognitive tutor: Applied research in mathematics education. *Psychonomics Bulletin & Review*, *14*(2), pp. 249-255.

Rittle-Johnson, B. and Star, J. R. (2007). Does comparing solution methods facilitate conceptual and procedural knowledge? An experimental study on learning to solve equations. *Journal of Educational Psychology*, *99*(3), 561–574.

Rowe, E., Baker, R.S., Asbell-Clarke, J., Kasman, E., Hawkins, W.J. (2014) Building Automated Detectors of Gameplay Strategies to Measure Implicit Science Learning. *Proceedings of the 7th International Conference on Educational Data Mining*, 337-338.

Rowe, E., Asbell-Clarke, J., Baker, R.S., Eagle, M., Hicks, A.G., Barnes, T.M., Brown, R.A., Edwards, T. (2017) Assessing implicit science learning in digital games. *Computers in Human Behavior*, *76C*, 617-630.

Rus, V., Maharjan, N., Tamang, L.J., Yudelson, M., Berman, S., Fancsali, S.E., Ritter, S. (2017). An Analysis of Human Tutors' Actions In Tutorial Dialogues, *Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2017)*, San Marco Island, Florida, USA May 16-18, 2017.

Rus, V., Moldovan, C., Niraula, N., & Graesser, A. C. (2012). Automated Discovery of Speech Act Categories in Educational Games. International Educational Data Mining Society.

Sao Pedro, M.A., Baker, R.S.J.d., Gobert, J., Montalvo, O. Nakama, A. (2013) Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, *23* (1), 1-39.

Searle, J. 1969. *Speech acts*. Cambridge University Press.

Siegler, R. S. (1988). Individual differences in strategy choices: Good students, not-so-good students, and perfectionists. *Child Development*, *59*, 833-851.

Su, J. M., Tseng, S. S., Wang, W., Weng, J. F., Yang, J. T. D., & Tsai, W. N. (2006). Learning portfolio analysis and mining for SCORM compliant environment. *Educational Technology & Society*, *9*(1), 262-275.

Tang, T. Y., & McCalla, G. (2002, July). Student modeling for a web-based learning environment: A data mining approach. In *AAAI/IAAI* (pp. 967-968).