British Journal of Educational Technology Vol 50 No 5 2019 2346–2364
doi:10.1111/bjet.12869

Understanding when students are active-in-thinking through modeling-in-context

Zachari Swiecki, Andrew R. Ruis, Dipesh Gautam, Vasile Rus and David Williamson Shaffer

Zachari Swiecki is a PhD student in the Department of Educational Psychology (learning sciences program) at the University of Wisconsin–Madison. His research focuses on modeling complex and collaborative thinking. Andrew R. Ruis is a research scientist in the Wisconsin Center for Education Research at the University of Wisconsin–Madison. His research focuses on modeling complex thinking and deliberative interaction in medical and surgical contexts. Dipesh Gautam is a PhD student in the Department of Computer Science at the University of Memphis. His research focuses on natural language processing, machine learning and semantic analysis. Vasile Rus is William Dunavant Professor of Computer Science at the University of Memphis. His research focuses on natural language processing, intelligent systems and data science. David Williamson Shaffer is the Vilas Distinguished Professor of Learning Sciences at the University of Wisconsin–Madison in the Department of Educational Psychology, the Obel Professor of Learning Analytics at the Aalborg University in Copenhagen and a Data Philosopher at the Wisconsin Center for Education Research. His research focuses on modeling complex and collaborative thinking and his most recent book is Quantitative Ethnography. Address for correspondence: Andrew R. Ruis, University of Wisconsin–Madison, 489 Educational Sciences Bldg., 1025 W. Johnson St., Madison, WI 53706-1706. Email: arruis@wisc.edu

Abstract

Learning-in-action depends on interactions with learning content, peers and real world problems. However, effective learning-in-action also depends on the extent to which students are *active-in-thinking*, making meaning of their learning experience. A critical component of any technology to support active thinking is the ability to ascertain whether (or to what extent) students have succeeded in internalizing the disciplinary strategies, norms of thinking, discourse practices and habits of mind that characterize deep understanding in a domain. This presents what we call a dilemma of *modeling-in-context*: teachers routinely analyze this kind of thinking for small numbers of students in activities they create or customize for the needs of their students; however, doing so at scale and in real-time requires some automated processes for modeling student work. Current techniques for developing models that reflect specific pedagogical activities and learning objectives that a teacher might create require either more expertise or more time than teachers have. In this paper, we examine a theoretical approach to addressing the problem of modeling active thinking in its pedagogical context that uses teacher-created rubrics to generate models of student work. The results of this examination show how appropriately constructed learning technologies can enable teachers to develop custom automated rubrics for modeling active thinking and meaning-making from the records of students' dialogic work.

Introduction

Many immersive educational technologies, such as digital games and simulations, enable students to take consequential action in a realistic context and to interact with peers, mentors and pedagogical agents. However, the extent to which such *learning-in-action* is effective depends on the extent to which students are *active-in-thinking*: engaging deeply with, reflecting on and

Practitioner Notes

What is already known about this topic

- Many immersive educational technologies, such as digital games and simulations, enable students to take consequential action in a realistic context and to interact with peers, mentors and pedagogical agents. Such technologies help students to be *active-in-thinking*: engaging deeply with, reflecting on and otherwise making meaning of their learning experience.
- There are now many immersive educational technologies with integrated authoring tools that enable teachers to customize the learning experience with relative ease, reducing barriers to adoption and improving student learning.
- Educational technologies that support learning-in-action typically contain student models that operate in real-time to control the behavior of pedagogical agents, deliver just-in-time interventions, select an appropriate content or otherwise measure and promote active thinking, but these student models may not work appropriately if teachers customize the learning experience.
- Much as there are authoring tools that allow teachers to customize the *curriculum* of a given learning technology, there is a need for authoring tools that allow teachers to customize the associated *student models* as well.

What this paper adds

- This paper presents a novel, rubric-based approach to develop automated student models for new activities that teachers develop in digital learning environments that promote active thinking.
- Our approach combines machine learning techniques with teacher expertise, allowing teachers to participate in the design of automated student models of active thinking that with further development could be scaled by leveraging their skills in rubric development.
- Our results show that a rubric-based approach can outperform a machine learning approach in this context. More importantly, in some cases, the rubric-based approach can produce reliable automated models based on the information that a teacher can easily provide.

Implications for practice and/or policy

- If integrated into authoring tools, the rubric-based approach could allow teachers to participate in the design of automated models for educational technologies customized to their instructional needs.
- Through this design process, teachers could develop a better understanding of how the automated modeling system works, which in turn could increase the adoption of educational technologies that promote active thinking.
- Because the rubric-based approach enables teachers to identify key connections among concepts relevant to the pedagogical context, rather than general concepts or linguistic features, it is more likely to facilitate targeted feedback to help promote the development of active thinking.

otherwise making meaning of their learning experience. While much work has been done by developing technologies and theories of learning to promote learning-in-action, a critical component of such technologies is the ability to ascertain whether (or to what extent) students have succeeded in internalizing the disciplinary strategies, norms of thinking, discourse practices and habits of mind that characterize deep understanding in a domain.

Many educational technologies that support learning-in-action contain student models that operate in real-time to control the behavior of pedagogical agents, deliver just-in-time interventions, select appropriate content or otherwise measure and promote active thinking (see, eg, Graesser *et al.*, 2018; Sottilare, Graesser, Hu, & Holden, 2013). Critically, these models are developed and validated for specific pedagogical contexts, and thus they cannot reliably be used in other settings. This presents what we have characterized as a dilemma of *modeling-in-context*:

> The use of educational technologies requires automated assessment processes to provide real-time feedback and assessment at scale, but to be effective, such assessments need to reflect the specific pedagogical context, including the learning objectives and student population. (Swiecki, Shaffer, & Misfeldt, 2017)

This dilemma is even more pronounced when teachers customize educational technologies to meet the needs of different student populations or to align the activities with changing standards or learning objectives. Once the "holy grail" of educational technology design (Aleven, McLaren, Sewall, & Koedinger, 2009), there are now many immersive digital learning environments with integrated authoring tools that enable teachers to make such changes with relative ease, reducing barriers to adoption and improving the learning experience (see, eg, Cubillo, Martin, Castro, & Boticki, 2015; Mehm, Göbel, Radke, & Steinmetz, 2009; Nye, Graesser, & Hu, 2015; Ososky, Brawner, Goldberg, & Sottilare, 2016; Sottilare, Graesser, Hu, & Brawner, 2015; Swiecki, Shaffer, & Misfeldt, 2017). However, when teachers modify the content or structure of an educational technology, the integrated models may not reflect the new pedagogical context. That is, adapting the curriculum may invalidate the original models (Gautam, Swiecki, Shaffer, Graesser, & Rus, 2017).

While teachers routinely analyze active thinking for small numbers of students, doing so at scale and in real-time requires some kind of automated student model. Student models are often used for assessment. However, in the context of active thinking, they are more often part of the process of providing generative feedback to students. Such feedback can come from a teacher, who uses student models to better understand students' work; from displays or visualizations of work presented to students; or from characters, agents or other sources within the system. Because most teachers do not have the ability to develop such processes unassisted, systems need to be designed that enable teachers to develop custom models of active thinking. In other words, much as there are authoring tools that allow teachers to customize the *curriculum* of a given learning technology, there is a need for authoring tools that allow teachers to customize the associated *student models* as well.

In this paper, we examine a theoretical approach to addressing the problem of modeling active thinking in its pedagogical context. Specifically, we describe an approach to develop automated student models that combine machine learning techniques with the pedagogical and domain expertise of teachers and demonstrate its utility by comparing it to a more traditional machine learning approach. The results of this examination show how appropriately constructed learning technologies can enable teachers to develop custom automated processes for modeling active thinking and meaning-making from the records of students' dialogic work.

Theory

The structure of automated student models

Automated student models have been used to promote and assess active thinking in a variety of technology-mediated environments. While there are many such environments (see, eg, McNamara, O'Reilly, Best, & Ozuru, 2016; Rowe, Shores, Mott, & Lester, 2011; Rus, Niraula, & Banjade, 2015), two prominent examples are *AutoTutor* and *virtual internships. AutoTutor* uses dialogues with pedagogical agents to facilitate active thinking in domains such as physics. During these dialogues, automated student models are used to assess student understanding of the domain and select appropriate pedagogical support (Graesser, 2016). In virtual internships, students collaborate to solve complex problems in domains such as engineering and urban planning. During the simulated internship, automated student models are used to facilitate reflective discussions (Saucerman, Ruis, & Shaffer, 2017), assess student work (Rus, Gautam, Swiecki, Shaffer, & Graesser, 2016) and provide real-time information to instructors (Shaffer, 2017).

The student models built into such educational technologies generally have three key components (Shermis & Burstein, 2013): (1) *responses*, such as scores, written responses or other feedback that can be assigned to student discourse, including actions, communication and submitted work; (2) one or more *classifiers* that automatically assign the appropriate response to student discourse; and (3) *features* of student discourse that classifiers use to assign responses. For example, the *AutoTutor* system uses automated classifiers based on two kinds of features—semantic similarity and pattern matching—to model student dialogues with conversational agents and generate pedagogical actions (Graesser, Chipman, Haynes, & Olney, 2005).

The development of automated student models typically follows one of the two approaches. *A priori* approaches involve specifying the classification rules that determine how responses are assigned based on the features of student discourse. For example, to develop a model of written work, one could specify a word count threshold, key terms that need to be present or other more complex criteria, such as adherence to a topic or the presence of a claim. To create a priori classification rules that accurately model student work thus requires both pedagogical content knowledge and the ability to compose such rules in a form that an automated classification system can implement. Most teachers have the former but not the latter, making it difficult to express a model of active thinking as a set of rules without appropriate scaffolding (Cai, Graesser, & Hu, 2015; Šimko, 2011; Zapata-Rivera, Jackson, & Katz, 2015).

Inductive approaches, such as machine learning, provide an alternate process for developing classification rules. Machine learning involves training algorithms on large amounts of human-assessed student discourse from the domain. This data is then used to induce relationships between features of the discourse and the human-generated responses. For example, a simple inductive classification approach may involve training a machine learning algorithm to model student writing based on features such as word count, frequency of key terms, the ratio of uppercase to lowercase letters and the number of errors in spelling or grammar. The model would learn the thresholds for each feature that best distinguishes among the different human responses. Such thresholds can then be used to automatically model new essays. However, while inductive approaches can be automated, thus reducing the need for expertise in the classifier development, they require large amounts of human-assessed data from a representative pedagogical context. Because such data are not available when teachers customize digital learning environments—by definition a customized environment is different from any other environment where data were previously collected—inductive approaches cannot typically be used to develop automated models for new curricula.

To address the lack of a suitable approach to the dilemma of modeling-in-context when teachers customize educational technologies, we propose an approach that leverages teachers' skills in assessing active thinking for small numbers of students to generate student models that are both scalable and sensitive to the complexities of active thinking in its pedagogical context. That is, we elicit from teachers the kind of work they already do well and explore a method for converting that work into an automated process. Specifically, we propose an approach to develop automated student models based on a scaffolded process of *rubric* development.

A rubric-based approach to automated student models

When teachers assess students' active thinking in dialogic work, such as writing, they often do so by creating a *rubric*. In many pedagogical contexts, Montgomery (2000) argues, teachers develop rubrics by (a) identifying key concepts and (b) indicating how those key concepts should be expressed and integrated within student work. Each assessment is characterized by the extent to which those criteria are met. In addition, teachers typically include (c) concrete exemplars to model the kind of work associated with each assessment (for more on the rubric development, see Glass, 2004; National Research Council, 2012; Reddy & Andrade, 2010).

Importantly, when teachers construct rubrics, they are providing the content that can seed both a priori and inductive approaches to automated model development. By specifying key concepts and how they are expressed and integrated for each response, teachers are providing classification rules, but those rules are not written in a way that a machine can implement. By specifying exemplars for each model, teachers are providing the kind of data that machine learning algorithms operate on, but they can only reasonably provide a small number of such examples and far fewer than such approaches typically require.

Studies show that teachers are able to modify and author learning technologies when an appropriate set of authoring tools are available (see, eg, Dağ, Durdu, & Gerdan, 2014). Therefore, to enable teachers to produce rubrics in a way that is familiar to them and to serve as the basis for the construction of automated models, we developed and tested an approach to scaffold rubric construction based on the theories of *connectivity*.

There is a considerable body of research that construes active thinking not as the mere possession of particular bits of knowledge or the demonstration of skills in isolation, but as a process of integrating them to frame, investigate and solve complex problems (see, eg, DiSessa, 1988; Linn, Eylon, & Davis, 2004; Madani *et al.*, 2017; Shaffer, 2012). The theory of *epistemic frames* (Shaffer, 2012), for example, suggests that active thinking consists of the cognitive connections that people make among the knowledge, skills, values and ways of making decisions characteristic of some domain. An epistemic frame, however, is not simply the set of concepts, actions and other elements of a domain; rather, it is the particular configuration of linkages among those elements. In other words, active thinking involves acquiring the epistemic frame of a domain and an epistemic frame is a particular *set of cognitive connections* that are revealed through the actions and interactions of an individual engaged in authentic tasks (or simulations of authentic tasks). The development of an epistemic frame, and by extension active thinking, can be modeled in the pedagogical context by measuring the connections that learners make among the frame elements.

A rubric-based approach to develop automated student models thus introduces an additional level of modeling to the classification process. Instead of attempting to develop a classifier that operates directly on features to determine the appropriate response, as in the examples given above, such an approach involves a two-level classification. First, features are classified into *Codes*: concepts, actions or other elements that are meaningful in some domain (for an in-depth discussion of Codes, see Shaffer, 2017). These Codes, which are the elements of a domain's epistemic frame, are thus more specific than overall models of student discourse, but unlike raw features, they have particular meanings or interpretations in the context of the domain. Because active thinking involves integrating Codes into an epistemic frame, the second classification step is to give the appropriate response based on how the Codes are connected in student work. Research has shown that co-occurrence of Codes within some *window* of discourse (eg, within some number of sentences in written work or within some span of time in conversation) is a good indicator of cognitive connections (Dyke, Kumar, Ai, & Rosé, 2012; i Cancho & Solé, 2001; Lund & Burgess, 1996; Ruis, Siebert-Evenstone, Pozen, Eagan, & Shaffer, 2019; Siebert-Evenstone *et al.*, 2017). This co-occurrence structure can then be used to assign the appropriate response. For example, Swiecki and colleagues (in press) used automated classifiers to identify the Codes present in the discourse data collected from military teams during training and used connections between these Codes to model active thinking in various training scenarios.

To seed the development of an automated model based on connectivity, teachers would need to provide three components: (a) the Codes relevant to the pedagogical context, (b) the relationships among those Codes that are associated with particular models and (c) a small number of exemplars in which the teacher has labeled the portions in which the Codes are expressed. These are, in effect, the components that teachers already provide in rubrics: key concepts (ie, Codes), information about how those concepts are integrated (ie, connections) and exemplars that illustrate the key concepts as they would appear in student work.

Thus, we argue that with appropriate scaffolding, teachers could produce rubrics that provide the material necessary to develop effective automated models of students' active thinking. This rubric-based approach is similar to inductive models in the sense that a set of exemplars are used to generate a classifier. However, because Codes are more specific than a general model of student discourse, Code classifiers can be induced from a small number of teacher-provided exemplars. The rubric-based approach is also similar to a priori models, in the sense that there are specific rules that define what combinations of Codes produce a particular model. But again, because Codes are meaningful in some domain, it is easier for teachers to specify rules for how they should connect that can easily be translated into rules that an automated classification system can implement.

In this study, we evaluated a rubric-based approach to developing automated student models. This approach utilizes rubrics composed by a teacher that indicate the Codes and key connections and that provide exemplars with the relevant portions annotated for the presence of those Codes. We compare this method with a machine learning approach operating on a comparable number of exemplars composed by a teacher. We hypothesize that, for a small set of data, (a) automated classifiers with Cohen's kappa significantly above the customary level of .65 to identify Codes from the features of student work can be developed using inductive techniques on a small number of teacher-composed exemplars in which the portions of the text that indicate the presence of a given Code are annotated; and (b) automated classifiers for assigning the appropriate response to student work based on the connectivity among Codes (as defined a priori by teachers) will have Cohen's kappa values that are significantly higher than classifiers induced from a comparable number of exemplars.

To test these hypotheses, we designed a study to answer the following research questions:

RQ1. How reliable are different inductive techniques for developing Code classifiers based on a small number of teacher-composed and labeled exemplars?

RQ2. Do rubric-based approaches to automated student modeling, which combine inductive and a priori approaches, model student discourse more reliably than purely inductive approaches with small amounts of student data?

We address these research questions with a small set of data collected in the context of one specific educational technology that includes an integrated suite of authoring tools. We chose to use a small set of data because, while it is well understood that inductive techniques perform relatively well in developing classifiers using large amounts of data, our purpose here is to explore the efficacy of the rubric-based approach for small sets of data.

Methods

Data

We conducted this study using data collected from the virtual internship *Land Science* (Bagley & Shaffer, 2009; Nash & Shaffer, 2011; Shaffer, 2007). In *Land Science*, students play the role of interns in a fictitious urban planning firm tasked with developing a land use plan for the city of Lowell, Massachusetts. To do this, students conduct a background research, design land use plans and respond to the stakeholder feedback. For example, designing land use plans involves using a geographic information system (GIS) tool to change the land use designations of different parcels and explore the impact of these changes on socioeconomic and environmental indicators.

Virtual internships like *Land Science* give students the opportunity to develop the epistemic frame of a particular profession through interactions with learning content, peers and real-world problems. In other words, *Land Science* is a good example of a technological environment designed to promote active thinking in a dialogic context.

After completing each activity in the virtual internship, students submit online notebook entries that document their work. Once submitted, notebooks are scored by human raters with the help of algorithms that automatically check for commonly made errors. Raters score each notebook entry on a four-item scale ranging from 0 (poor) to 3 (excellent). These notebooks are thus representations of the conclusions that students draw from their active thinking about land use issues.

To address our research questions, we developed automated models for two activities in *Land Science*. In Activity 1 (Recommendations), students document their proposed land use changes by describing where in the city they made changes using the GIS tool, the current land uses of those locations, and their proposed changes. In Activity 2 (Plan Justifications), students justify their proposed land use changes on economic or environmental grounds or in relation to certain stakeholder requirements.

Rubric-based approach

To create automated models using the rubric-based approach, a teacher with experience using *Land Science* developed rubrics for both activities. Each rubric had three main components. First, the teacher defined the *Codes* or relevant concepts for the pedagogical context. Next, she wrote short lists of *keywords* associated with each Code and a small number of *exemplars* in which she labeled the sentences or phrases that expressed the Codes (see Table 1). The teacher was permitted to use the same Codes for different activities and it was permissible for Codes to overlap—ie, for

Table 1: Codes with exemplars and keywords for Activity 1 (Recommendations)

Code name	*Exemplars*	*Exemplar keywords*
Recommended land use	I decided to change most of the land around the river that was industrial to wetlands	Commercial, industrial and open-space
Original land use	I changed open space and commercial	Commercial, industrial and open-space
Location	Land around the river	River, north, south, east and west
Indicator change	Oriole count and the turtle nesting sites went up	Runoff, phosphorous, housing and "nesting sites"
Stakeholder concerns	Natalie's wish to decrease runoff into rivers	Neighborhood protection organization and community action group

a single excerpt to contain multiple Codes or for two Codes to share exemplar keywords. Finally, she defined the relationships—or *connections*—among the Codes associated with particular level of student work (see Table 2).

We did not give specific instructions as to how many exemplars the teacher should write or how long the exemplars should be. Instead, we asked the teacher to write only as many exemplars as needed to provide at least three exemplar texts for each Code. The teacher wrote eight exemplar entries for Activity 1, one to three sentences in length, for a total of 15 sentences (a single exemplar could (and often did) contain more than one Code). She wrote 14 exemplar entries for Activity 2, one to eight sentences in length, for a total of 28 sentences.

Because automated modeling using the teacher-defined rubrics involves two levels of classification—identifying the Codes and identifying the connections among Codes—the first step was to create inductive classifiers for each Code. We tested four kinds of *Code classifiers* which used latent semantic analysis (LSA), regular expression (RGX) matching or combinations of the two in order to identify Codes in student notebook entries. Because the sentences or phrases written by the teacher for a given Code were a maximum of one sentence in length, prior to the classification we segmented for each student's notebook entry into sentences.

Code classifiers

LSA. As described above, each Code has a corresponding set of texts—ie, a set of sentences and phrases—composed by the teacher. For a given student's notebook entry, the LSA classifier calculated the semantic similarity between the set of texts for each Code and each sentence in the entry. For a given Code, if the maximum similarity value was above a threshold, the sentence was classified as containing the Code. For each Code, the threshold was defined as the average semantic similarity between the texts of the Code minus one standard deviation. For some Codes, this resulted in thresholds that were very low (ie, less than 0.2 on a scale from 0 to 1). In these cases, we set the threshold to 0.5 to control for Type I errors. The semantic similarity was calculated using the SEMILAR toolkit (Rus, Lintean, Banjade, Niraula, & Stefanescu, 2013) and an LSA space built using the Touchstone Applied Science Associates (TASA) corpus (Landauer, Foltz, & Laham, 1998).

RGX. The RGX classifiers were developed using the teacher-specified keywords for each Code. After converting the keywords to regular expressions, the RGX classifiers used regular expression matching to identify the presence or absence of Codes in each sentence of a given notebook entry.

AND. The AND classifier classified the sentences of each notebook entry as containing a given Code if both the LSA *and* RGX classifiers classified the sentence as having the Code.

OR. The OR classifier classified the sentences of each notebook entry as containing a given Code if either the LSA *or* the RGX classifier classified the sentence as having the Code.

Rubric classifiers

Next, we used an a priori approach to develop classifiers that model notebook entries based on the connections among Codes defined in the rubrics for each activity.

The rubric classifiers use a moving window to identify connections among Codes in notebook entries—ie, Codes that co-occur within the window are considered connected. In this study, we used a window size of one sentence, which represents the most conservative definition of connectivity. Relevant connections among Codes were defined using the criteria present in the rubric (see Table 2). For a given activity, once all criteria had been checked, the classifiers assigned the *highest* level with matching criteria. Thus, a notebook entry may meet the criteria for levels one, two and three, but the classifier would assign a score of three. We developed and tested four rubric classifiers for both activities, each using one of the Code classification approaches described above.

Table 2: Rubric showing relationships among Codes and the associated levels of student work for Activity 1 (Recommendations)

0 (Poor)	*1 (Baseline)*	*2 (Acceptable)*	*3 (Excellent)*
No sentences include **Recommended Land Use** or **Original Land Use** OR No sentences include **Recommended Land Use** and at least one sentence includes **Indicator Change** OR **Stakeholder Concerns**	At least one sentence includes **Recommended Land Use** or **Original Land Use**	At least half of the sentences that include **Recommended Land Use** also include **Original Land Use**	All sentences that include **Recommended Land Use** also include **Original Land Use** AND At least one sentence that includes **Recommended Land Use** also includes **Location**

Table 3: Sample exemplars for Activity 1 (Recommendations)

Level	*Exemplar*
0	The only thing I needed to change was the CO in the air. My housing, nesting, jobs, sales and birds were already good for my stakeholders
1	I recommended increasing housing options near commercial areas, in addition to limit the bird population
2	I changed the industrial zones to an open space. Then I changed the current open space to commercial space. I also changed residential space to commercial space
3	The changes me and my group made were we moved industrial zones from the river and changed it to wetlands or an open space. Also, I moved industrial zones closer to the residents and changed some industrial zones to commercial. These plans decrease carbon monoxide levels and keep jobs around the area

Inductive approach

To develop automated classifiers using an inductive approach, we asked the same teacher to write three exemplar notebook entries for each level—ie, three exemplars with a level of 0, three with a level of 1 and so on—for a total of 12 exemplars per section (see Table 3). We did not give the teacher-specific guidelines for the exemplar length. Exemplars for Activity 1 ranged from one to three sentences in length, for a total of 29 sentences (compared with 15 for the rubric-based approach). Exemplars for Activity 2 ranged from one to eight sentences, for a total of 46 sentences (compared with 28 for the rubric-based approach).

The inductive classifier used LSA to calculate the semantic similarity between a given student's notebook entry and the exemplar text for each level, assigning the level whose text is most similar to the notebook entry. As with the rubric-based approach, the semantic similarity was calculated using the SEMILAR toolkit and an LSA space was built using the TASA corpus.

Evaluation of classifier performance

To evaluate the performance of the rubric-based and inductive classifiers, we used notebook entries collected from previous implementations of *Land Science*. We randomly selected 50 entries from Activity 1 and 50 entries from Activity 2.

To address our first research question, we evaluated the reliability of the Code classifiers used in the rubric approach. Two raters used social moderation (Frederiksen, Sipusic, Sherin, & Wolfe, 1998; Herrenkohl & Cornelius, 2013; Shaffer, 2017) to code each sentence of each notebook entry for the presence or absence of each Code, which resulted in a complete agreement between the raters for all Codes. We then compared the human and automated classifications using Cohen's κ (kappa) with an agreement threshold of .65. We modeled the generalizability of these kappa values using Shaffer's ρ (rho), which tests whether an achieved level of agreement generalizes to the data collected under similar conditions (Eagan, Rogers, Pozen, Marquart, & Shaffer, 2016). Rho is interpreted in the same way as a *p*-value in standard hypothesis testing. By setting an alpha (acceptable Type I error) level of .05, a rho value less than .05 suggests that the achieved level of agreement generalizes.

To address our second research question, we compared the performance of the rubric-based and inductive classifiers in terms of their agreement with human assessments. Specifically, the student entries from both activities were modeled by all the automated classifiers and also manually assessed by the teacher. To test whether the automated classifiers could make the basic distinction between entries that were poor and those that were not, we combined levels 1 through 3 such that each entry was modeled either as poor (0) or acceptable (1).

We modeled the performance of the automated classifiers for each activity by (a) testing whether any of the automated classifiers achieved an acceptable agreement with the human classifier (ie, the teacher) and (b) testing whether any differences in the level of agreement achieved by different automated classifiers were statistically significant. To perform the first test, we computed kappa between each automated classifier and the teacher to determine the level of agreement and we computed rho to determine whether the level of agreement was significantly greater than .65. To perform the second test, we computed the difference in kappa values between each unique pair of automated classifiers. We then computed rho to determine whether the larger of the two kappa values was significantly greater than a kappa threshold defined by the smaller of the two kappa values. The null hypothesis for such a test using rho is that the data modeled by two classifiers is sampled from a larger pool of modeled data whose kappa is less than a predefined threshold. A rho of less than .05 means that the kappa observed on the sample is greater than 95 percent of the kappa values in the null hypothesis distribution. This allows us to reject the null hypothesis that the true rate of agreement between the two classifiers is below the threshold, supporting the hypothesis that the true rate of agreement is above the threshold. Thus, we can say that classifiers with rho values below .05 perform significantly better than classifiers with kappa values less than or equal to the set threshold. In other words, for two competing classifiers, we can test for statistical differences in their agreement with the teacher by setting the kappa of the null hypothesis distribution equal to the kappa of the classifier with lower performance. If the rho value of the subsequent test is less than .05, we can conclude that the performance of the other classifier is significantly better.

Results

RQ1: How reliable are different inductive techniques for developing Code classifiers based on a small number of teacher-composed and labeled exemplars?

Activity 1 (Recommendations)

As shown in Table 4, the RGX approach performed the best overall for Activity 1, followed by the AND approach. Both approaches had two Code classifiers (location and stakeholder concerns) with kappa values greater than or equal to .65. However, only for the RGX approach were both kappa values statistically significant. The LSA and OR approaches did not have any kappa values greater than or equal to .65.

Activity 2 (Plan Justifications)

As shown in Table 5, the OR approach performed the best overall for Activity 2. This approach had four Code classifiers (Runoff, Jobs, Land use decision and Carbon monoxide) with kappa values greater than or equal to .65. However, the kappa value for Carbon monoxide was not statistically significant. The LSA and RGX approaches had the next best performance. Both had

Table 4: Kappa values for Activity 1 (Recommendations) code classifiers

	Recommended land use	*Original land use*	*Location*	*Indicator change*	*Stakeholder concerns*
LSA	.54	.41	.20	.29	.004
RGX	.55	.47	**.73***	.63	**.80***
AND	.47	.51	**.75***	.37	**.67**
OR	.61	.37	.20	.54	.06

Note: **Bold** indicates $\kappa \geq .65$.
*indicates $\rho(.65) < .05$.

Table 5: Kappa values for Activity 2 (Plan Justifications) code classifiers

	Runoff	*Phosphorous*	*Orioles*	*Housing*	*Nesting sites*	*Jobs*	*Sales*	*Carbon monoxide*	*Land use decision*	*Indicator change*	*Indicator value*
LSA	.51	.33	.59	.55	.14	**.83***	.50	.32	**.67***	.35	.08
RGX	**.72***	.62	.60	.04	.32	**.83***	−.01	**.65**	.58	.51	.06
AND	.51	.33	.58	.05	.20	**.83***	.00	.32	.53	.35	.12
OR	**.72***	.62	.61	.49	.21	**.83***	.48	**.65**	**.73***	.49	.06

Note: **Bold** indicates $\kappa \geq .65$.
*indicates $\rho(.65) < .05$.

two Code classifiers with statistically significant kappa values (Jobs and Land use decisions for LSA and Jobs and Runoff for RGX).

Overall, as shown in Figure 1, 87.5% of the Codes had at least one classifier whose kappa value was greater than or equal to .50, but only 37.5% of the Codes had at least one kappa value greater than or equal to .65. Together, these results suggest that no single approach to Code classification is best across the two activities. Performance across Codes for each of the four classification approaches was relatively low and no approach resulted in statistically significant kappa values for all—or even a majority—of the Codes. However, it is unclear to what extent the performance of the Code classifiers affects the performance of the rubric-based final models, as kappa greater than or equal to .65 is an arbitrary (though widely used) threshold. To evaluate the performance of the rubric-based approach, we compared it to an inductive approach operating on a similar number of exemplars.

RQ2: Do rubric-based approaches to automated student modeling, which combine inductive and a priori approaches, model student discourse more reliably than purely inductive approaches?

Activity 1 (Recommendations)

As shown in Table 6, all rubric-based classifiers performed significantly better than the inductive classifier, which had a kappa value of .26. Among the rubric-based classifiers, the RGX classifier performed best, being the only approach with a kappa value (.84) that was significantly above the kappa threshold of .65. That is, the RGX classifier was the only approach that achieved an acceptable agreement with the teacher. Moreover, the RGX classifier performed significantly better than all others tested.

Activity 2 (Plan Justifications)

As shown in Table 7, all rubric-based classifiers performed significantly better than the inductive classifier. Among the rubric-based classifiers, the OR approach performed significantly better than all others; however, its kappa value was not statistically significant.

Together, these results suggest that rubric-based approaches can significantly outperform inductive approaches when only small numbers of exemplars are available. Interestingly, our results also suggest that rubric-based classifiers perform well even when the reliability of the Code

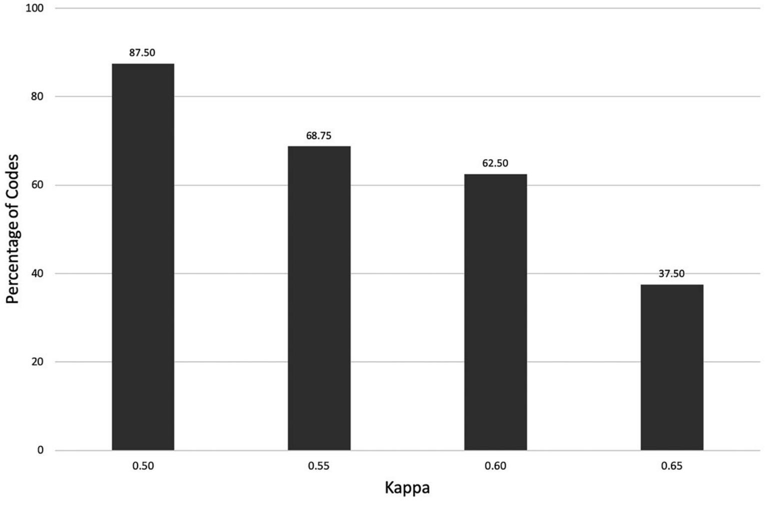

Figure 1: Percentage of Codes with classifier reliability greater than or equal to a given kappa value

Table 6: Classifier kappa, kappa difference and rho comparisons for Activity 1 (Recommendations)

		Difference in kappa			
Model	*Kappa*	*Inductive*	*LSA*	*AND*	*OR*
Inductive	.26				
Rubric-LSA	.59	+.33*			
Rubric-AND	.63	+.37*	+.04		
Rubric-OR	**.65**	+.39*	+.07	+.02	
Rubric-RGX	**.84**§	+.58*	+.25*	+.21*	+.19*

Note: **Bold** indicates $\kappa \geq .65$.
§indicates $\rho(.65) < .05$; *indicates $\rho < .05$ for the difference in kappa.

Table 7: Classifier kappa, kappa difference and rho comparisons for Activity 2 (Plan Justifications)

		Difference in kappa			
Model	*Kappa*	*Inductive*	*AND*	*RGX*	*LSA*
Inductive	.08				
Rubric-AND	.33	+.25*			
Rubric-RGX	.47	+.39*	+.14*		
Rubric-LSA	.53	+.45*	+.20*	+.06*	
Rubric-OR	.60	+.52*	+.27*	+.13*	+.07*

Note: **Bold** indicates $\kappa \geq .65$.
*indicates $\rho < .05$ for the difference in kappa.

classification is not especially high. For example, there were only two Code classifiers for Activity 1 that had statistically significant kappa values at a threshold of .65 using the RGX approach; however, the Rubric-RGX classifier had a statistically significant kappa value of .84.

Discussion

This paper presents a novel, rubric-based approach to developing automated student models for new activities that teachers develop in digital learning environments that promote active thinking. We compared this approach to a more traditional inductive approach implemented under similar constraints. While such an inductive approach would not normally be used when large amounts of modeled student data are not available, it provides a useful baseline for comparison. Our results show that the rubric-based approach can outperform an inductive approach in this context. More importantly, in some cases, the rubric-based approach can produce reliable automated models based on the information that a teacher can easily provide.

Our results thus suggest a new approach to the dilemma of modeling-in-context: that the use of educational technologies requires automated models to provide real-time feedback at scale, but to be effective, such models need to reflect the specific pedagogical context. While many extant educational technologies employ automated student models, these models are typically developed without customization in mind and through a collaboration between domain and computational experts that does not directly involve teachers in the modeling process. In contrast, our approach combines machine learning techniques with teacher expertise, allowing teachers to participate in the design of automated student models of active thinking that with further work might be implemented at scale by leveraging their skills in the rubric development—identifying the key concepts, connections between concepts and exemplars associated with modeling in a particular pedagogical context.

This approach to the automated model development is useful because it leverages existing teacher expertise, but also because it operates on *specific* elements of an epistemic frame (ie, Codes) and their connections, rather than *general* features of student discourse. This feature constrains the machine learning problem and these constraints, we argue, explain why the rubric-based approach was able to outperform the inductive approach using a small dataset.

This having been said, the failure of the rubric-based approach to accurately classify student notebooks in Activity 2 shows that more work is clearly needed before this approach could be implemented and to more completely understand the conditions under which it is most effective. Thus, while these results suggest that appropriately constructed learning technologies can elicit from teachers the information needed to develop custom automated rubrics for modeling active thinking, this study has several important limitations.

First, we only developed and tested the rubric-based approach in one pedagogical context with a small amount of data and only on data from the students' written work, as opposed to discourse or other records of students thinking. However, while the particular Code classifiers we used were domain-specific, the overall approach is domain agnostic; ie, the approach is applicable to any context in which the goal is to model active thinking and the data to be modeled are in the form of text (or are convertible to text). Because many educational technologies meet these criteria, we expect the approach to be effective across a range of contexts. Moreover, the two activities in this study reflect qualitatively different thinking: solutions themselves and their justifications. Further studies could explore whether and how the nature of student activities influence the accuracy of a rubric-based model.

Second, our method for evaluating the performance of the rubric-based approach collapsed the original four levels of student work to a binary classification. This decision likely occluded some classification errors. However, we made this decision to examine whether the approach could make the basic distinction between those notebook entries that were low quality—and thus more likely to need feedback or intervention from a teacher—and those that met a minimum standard for acceptability. Future work will evaluate the performance of the rubric-based approach for finer-grained classifications.

Third, it is possible that several improvements could be made to the rubric-based approach. For example, inductive classification techniques different from those used here, such as neural networks, could yield better results. While our prior work on data from the same pedagogical context suggests that neural networks have similar performance to LSA and RGX (Gautam *et al.*, 2017), future work will continue to test different inductive techniques. Moreover, the rubric-based approach tested here used the same inductive classification techniques for all Codes. Because the classifiers performed differently for a given Code, it is possible that the approach could be improved using the Code classifier that performed the best for each Code in the final automated modeling. Our future work will test this hypothesis. Similarly, further work could compare this rubric-based approach to a wider range of inductive methods; however, we note that because inductive methods are, in general, designed to work with large sets of coded exemplars, it seems likely that we would achieve similar results.

Fourth, the rubric-based approach described here requires researcher expertise to translate teacher provided information into student models. However, this translation could be easily automated. The primary challenge would be to develop a system that scaffolds teachers' rubric construction so as to elicit the type and amount of information needed to develop a reliable model. For example, it was clear from our work on this study that teachers are more likely to say things like “The student explains how their recommended land uses are different from the original land uses” rather than “All sentences that include **Recommended Land Use** also include **Original Land Use**.” How to best help teachers identify Codes explicitly and describe their role in the rubric in precise terms requires further research before the method we describe would be scaleable. In the pedagogical context we used for this study, there exists an integrated suite of authoring tools designed to help teachers customize the educational technology. Using this existing infrastructure, new authoring tools could be designed to scaffold teachers’ design of the rubrics and use this information to automatically generate student models. In theory, this should be possible for other educational technologies designed to promote active thinking that have associated authoring tools.

Fifth, this study did not directly address the question of what additional expertise might be needed to create sound rubrics, such as the expertise of multiple teachers, possibly in collaboration with domain experts. While additional domain or pedagogical expertise could be useful in creating better rubrics, the issue we are addressing in this pilot study is whether a rubric could be generated from exemplars—and how such a rubric would perform. The question of what the right constellation of expertise in creating a rubric might be is an important topic for the future study, but as we argue above, given that teachers often customize activities for their students, we believe that it will be important to develop a method for generating rubrics that a single teacher could use. It is possible, of course, that bias could be introduced by the particular pedagogical aims of the teacher or that the act of reframing rubrics in more formal terms could introduce systematic bias, but this question is beyond the scope of the current study.

Despite these limitations, our results provide proof of concept for teacher-generated automated models of active thinking at scale for small datasets. As such, they have important implications for educational technologies that promote active thinking. If integrated into authoring tools, the rubric-based approach could allow teachers to participate in the design of automated models for educational technologies customized to their instructional needs. Through this design process, teachers could develop a better understanding of how the automated modeling system works, which in turn could increase the adoption of educational technologies that promote active thinking. Moreover, because the rubric-based approach operationalizes the identification of connections among domain-specific semantic concepts, rather than general concepts of lexical/syntactic features, it is more likely to facilitate targeted feedback to help promote the development of active thinking.

Statements on open data, ethics and conflict of interest

All research activities were conducted under a protocol approved by the University of Wisconsin–Madison Institutional Review Board. In accordance with the approved protocol, we are not permitted to disseminate the data used in this study.

The authors have no conflicts of interest to report.

Acknowledgements

This work was funded in part by the National Science Foundation (DRL-1661036, DRL-1713110), the U.S. Army Research Laboratory (W911NF-18-2-0039), the Wisconsin Alumni Research Foundation and the Office of the Vice-Chancellor for Research and Graduate Education at the University of Wisconsin–Madison. The opinions, findings and conclusions do not reflect the views of the funding agencies, cooperating institutions or other individuals.

References

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, *19*(2), 105–154.

Bagley, E. A., & Shaffer, D. W. (2009). When people get in the way: Promoting civic thinking through epistemic game play. *International Journal of Gaming and Computer-Mediated Simulations*, *1*(1), 36–52.

Cai, Z., Graesser, A. C., & Hu, X. (2015). ASAT: AutoTutor script authoring tool. In B. Sottilare, A. C. Graesser, X. Hu, & K. Brawner (Eds.), *Design recommendations for intelligent tutoring systems: Authoring tools* (pp. 199–210). Orlando, FL: U.S. Army Research Laboratory.

Cubillo, J., Martin, S., Castro, M., & Boticki, I. (2015). Preparing augmented reality learning content should be easy: UNED ARLE—An authoring tool for augmented reality learning environments. *Computer Applications in Engineering Education*, *23*(5), 778–789.

Dağ, F., Durdu, L., & Gerdan, S. (2014). Evaluation of educational authoring tools for teachers stressing of perceived usability features. *Procedia-Social and Behavioral Sciences*, *116*, 888–901.

DiSessa, A. A. (1988). Knowledge in pieces. In G. Forman & P. Pufall (Eds.), *Constructivism in the computer age* (pp. 47–70). Hillsdale, NJ: Erlbaum.

Dyke, G., Kumar, R., Ai, H., & Rosé, C. P. (2012). Challenging assumptions: Using sliding window visualizations to reveal time-based irregularities in CSCL processes. In *Proceedings of the 10th International Conference of the Learning Sciences* (Vol. *1*, pp. 363–370). Sydney, Australia.

Eagan, B. R., Rogers, B., Pozen, R., Marquart, C., & Shaffer, D. W. (2016). *rhoR: Rho for inter rater reliability (Version 1.1.0)*. Retrieved from https://cran.r-project.org/web/packages/rhoR/index.html

Frederiksen, J. R., Sipusic, M., Sherin, M., & Wolfe, E. W. (1998). Video portfolio assessment: Creating a framework for viewing the functions of teaching. *Educational Assessment*, *5*(4), 225–297.

Gautam, D., Swiecki, Z., Shaffer, D. W., Graesser, A. C., & Rus, V. (2017). Modeling classifiers for virtual internships without participant data. In *Proceedings of the 10th International Conference on Educational Data Mining* (pp. 278–283). Wuhan, China.

Glass, K. T. (2004). *Curriculum design for writing instruction: Creating standards-based lesson plans and rubrics*. Thousand Oaks, CA: Corwin Press.

Graesser, A. C. (2016). Conversations with AutoTutor help students learn. *International Journal of Artificial Intelligence in Education*, *26*(1), 124–132.

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, *48*(4), 612–618.

Graesser, A. C., Dowell, N., Hampton, A. J., Lippert, A. M., Li, H., & Shaffer, D. W. (2018). Building intelligent conversational tutors and mentors for team collaborative problem solving: Guidance from the 2015 Program for International Student Assessment. In J. Johnston, R. Sottialre, A. M. Sinatra, & C. S. Burke (Eds.), *Building Intelligent Tutoring Systems for Teams* (Vol. *19*, pp. 173–211). Bingley: Emerald Publishing.

Herrenkohl, L. R., & Cornelius, L. (2013). Investigating elementary students' scientific and historical argumentation. *Journal of the Learning Sciences*, *22*(3), 413–461.

i Cancho, R. F., & Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *268*(1482), 2261–2265.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2–3), 259–284.

Linn, M. C., Eylon, B.-S., & Davis, E. A. (2004). The knowledge integration perspective on learning. In M. C. Linn, E. A. Davis, & P. Bell (Eds.), *Internet environments for science education* (pp. 29–46). Mahwah, NJ: Lawrence Erlbaum Associates.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*(2), 203–208.

Madani, A., Vassiliou, M. C., Watanabe, Y., Al-Halabi, B., Al-Rowais, M. S., Deckelbaum, D. L., … Feldman, L. S. (2017). What are the principles that guide behaviors in the operating room? Creating a framework to define and measure performance. *Annals of Surgery*, *265*(2), 255–267.

McNamara, D. S., O'Reilly, T. P., Best, R. M., & Ozuru, Y. (2016). Improving adolescent students' reading comprehension with Istart. *Journal of Educational Computing Research*, *34*(2), 147–171.

Mehm, F., Göbel, S., Radke, S., & Steinmetz, R. (2009). Authoring environment for story-based digital educational games. In *Proceedings of the 1st International Open Workshop on Intelligent Personalization and Adaptation in Digital Educational Games* (pp. 113–124).

Montgomery, K. (2000). Classroom rubrics: Systematizing what teachers do naturally. *The Clearing House*, *73*(6), 324–328.

Nash, P., & Shaffer, D. W. (2011). Mentor modeling: The internalization of modeled professional thinking in an epistemic game. *Journal of Computer Assisted Learning*, *27*(2), 173–189.

National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. Washington, DC: National Academies Press.

Nye, B. D., Graesser, A. C., & Hu, X. (2015). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, *24*(4), 427–469.

Ososky, S., Brawner, K., Goldberg, B., & Sottilare, R. (2016). GIFT Cloud: Improving usability of Adaptive Tutor Authoring Tools within a web-based application. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *60*(1), 1389–1393.

Reddy, Y. M., & Andrade, H. (2010). A review of rubric use in higher education. *Assessment and Evaluation in Higher Education*, *35*(4), 435–448.

Rowe, J. P., Shores, L. R., Mott, B. W., & Lester, J. C. (2011). Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*, *21*(1–2), 115–133.

Ruis, A. R., Siebert-Evenstone, A. L., Pozen, R., Eagan, B. R., & Shaffer, D. W. (2019). Finding common ground: A method for measuring recent temporal context in analyses of complex, collaborative thinking. In *Proceedings of the 13th International Conference on Computer-Supported Collaborative Learning* (pp. 136–143). Lyon, France.

Rus, V., Gautam, D., Swiecki, Z., Shaffer, D. W., & Graesser, A. C. (2016). Assessing student-generated design justifications in virtual engineering internships. In *Proceedings of the 9th International Conference on Educational Data Mining* (pp. 496–501). Raleigh, NC.

Rus, V., Lintean, M., Banjade, R., Niraula, N., & Stefanescu, D. (2013). SEMILAR: The semantic similarity toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 163–168). Sofia, Bulgaria.

Rus, V., Niraula, N., & Banjade, R. (2015). DeepTutor: An effective, online intelligent tutoring system that promotes deep learning. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, TX.

Saucerman, J., Ruis, A. R., & Shaffer, D. W. (2017). Automating the detection of reflection-on-action. *Journal of Learning Analytics*, *4*(2), 207–234.

Shaffer, D. W. (2007). *How computer games help children learn*. New York, NY: Palgrave Macmillan.

Shaffer, D. W. (2012). Models of situated action: Computer games and the problem of transfer. In C. Steinkuehler, K. D. Squire, & S. A. Barab (Eds.), *Games, learning, and society: Learning and meaning in the digital age* (pp. 403–431). Cambridge, UK: Cambridge University Press.

Shaffer, D. W. (2017). *Quantitative ethnography*. Madison, WI: Cathcart Press.

Shermis, M. D., & Burstein, J. (2013). *Handbook of automated essay evaluation: Current applications and new directions*. New York, NY: Routledge.

Siebert-Evenstone, A. L., Irgens, G. A., Collier, W., Swiecki, Z., Ruis, A. R., & Shaffer, D. W. (2017). In search of conversational grain size: Modelling semantic structure using moving stanza windows. *Journal of Learning Analytics*, *4*(3), 123–139.

Šimko, M. (2011). Automated domain model creation for adaptive social educational environments. *Information Sciences and Technologies Bulletin of the ACM Slovakia*, *3*(2), 119–121.

Sottilare, R., Graesser, A., Hu, X., & Brawner, K. (Eds.) (2015). *Design recommendations for intelligent tutoring systems: Authoring tools and expert modeling techniques*. Orlando, FL: U.S. Army Research Laboratory.

Sottilare, R., Graesser, A. C., Hu, X., & Holden, H. (2013). *Design recommendations for intelligent tutoring systems: Learner modeling*. Orlando, FL: Army Research Laboratory.

Swiecki, Z., Shaffer, D. W., & Misfeldt, M. (2017). Dependency-centered design as an approach to pedagogical authoring. In Y. Baek (Ed.), *Game-based learning: Theory, strategies and performance outcomes* (pp. 167–188). Hauppauge, NY: Nova Science Publishers, Inc.
Zapata-Rivera, D., Jackson, G. T., & Katz, I. (2015). Authoring conversation-based assessment scenarios. *Design Recommendations for Intelligent Tutoring Systems*, *3*, 169–178.