Chapter Title: **Quick Red Fox: An App Supporting a New Paradigm in Qualitative Research on AIED for STEM**

Authors: Stephen Hutt, Ryan S. Baker, Jaclyn Ocumpaugh, Anabil Munshi, J.M.A.L. Andres, Shamya Karumbaiah, Stefan Slater, Gautam Biswas, Luc Paquette, Nigel Bosch, Martin van Velsen

Abstract

Artificial Intelligence in Education research for STEM domains has largely been quantitative in nature, but qualitative research offers several advantages as part of a mixed-methods approach. In particular, qualitative research enables researchers to develop deeper phenomenological understanding of how learners represent their activity to themselves. However, qualitative research can be challenging to apply in classrooms: it is resource-intensive, does not scale well, and the phenomena of the greatest interest to AIED researchers are often intermittent and occasional. For example, researchers may be interested in studying situations where a learning activity is known to be overly time-consuming or difficult, or in theoretical investigations of shifts in student affect such as transitions from confusion to frustration. However, given multiple potential learners to interview (e.g., a classroom of students), it can be difficult for a researcher embedded in the classroom to prioritize which learner to speak with next. Simple strategies, whether sequential or random, may miss (often fleeting) key moments in a participant's experience (e.g., affective transitions).

We address this problem with a new app that leverages user modeling techniques (e.g., behavior and affect-sensing) to direct interviewers to learners at critical, theory-driven moments as they learn with AIED technologies in the classroom. This paper details the design and implementation of this research paradigm as an alternative method for studying learning and using existing STEM AIED technologies in research. We examine the potential of this paradigm through the lens of two case studies where 99 students interacted with a computer-based learning environment as part of their regular classroom instruction. Unscripted interviews were triggered at or immediately after critical moments (such as peak frustration or shifts from confusion to boredom). The app facilitated 594 interviews, each averaging 1-2 minutes in length. Our findings indicate that by using machine learned models to optimize researcher time, we can gain a deeper insight into students' behaviors and their motivations, thus furthering AIED research. We discuss the potential broader applications of this app and the research it affords.

1 Introduction

Educational software and computer-based learning environments have become an increasingly prominent part of K-12 education. Even before the COVID-19 pandemic, there was a considerable increase in the use of these technologies (Marcus-Quinn & Hourigan, 2017), and this trend has amplified in the last year as teachers who previously did not have technology (or the training to use it) were asked to convert to virtual teaching in a very short time. As students return to the classroom with these technological investments in place, interactive learning environments, such

as intelligent tutoring systems, simulations, and problem-solving platforms are likely to be even more ubiquitous than they were two years ago.

As we seek to improve these technologies and create richer, more dynamic experiences, we must first gain a deeper understanding of how students learn with technology, and how these processes may be different than those encountered in traditional classroom learning. In order to truly understand, we must often tap into a student's internal cognitive and noncognitive processes, which may be hard for students (especially younger students) to articulate in traditional quantitative survey instruments. More qualitative methods such as think alouds, interviews, and open-ended self-reports, by contrast, can provide a clearer window into these processes.

Qualitative research can be critical in improving education as it gives information on "how" and "why" research questions that may be otherwise unanswerable (Cleland, 2017). Qualitative data can focus on thoughts, concepts or experiences that may in turn be used to gain a deeper understanding of phenomena and context. By asking questions that cannot be boiled down to "how many" or "please rate", it becomes possible to collect a rich dataset that can complement the quantitative data that is already frequently collected (e.g., log data, student models etc.) within AIED research. However, conducting qualitative research on AIED technologies has proven challenging, and the logistics of collecting enough qualitative data (and the right qualitative data) has often proven highly resource-intensive (e.g. Schofield, 1995). For example, one option might be to use think-alouds or emote-alouds to get students to vocalize processes. However, this approach is resource-intensive, does not scale well, and is challenging to apply in real classrooms, often requiring researchers to pull students out of context and into a separate room (e.g., Kelley et al., 2015). Other researchers have conducted interviews or observational approaches in classrooms -- however, doing so has also proven highly time-consuming when attempting to exhaustively capture key events (e.g., Cobb et al., 2001).

Beyond this, in cases where a classroom observer is trying to study students' internal cognitive and affective processes as they use an AIED system, it may be difficult to capture events and processes of interest, without capturing huge amounts of data that is then difficult and time-consuming to filter through. Both observation and interview methods are vulnerable to what Wessel (2015) describes as the "one shot" problem, where events occur only once, and if missed, cannot be studied. Two sampling methods that are often used in classroom studies are momentary time sampling, (as with the widely-used BROMP quantitative observation protocol -- Baker et al., 2020; Ocumpaugh et al., 2015) and scan methods, where the observer monitors an entire classroom full of students at once. Momentary time sampling methods are known to bias towards events of longer duration and/or which occur more frequently (Meany-Daboul, Roscoe, Bourret, & Ahearn, 2007).Scan methods tend to bias towards more dramatic behaviors, while more subtle student actions may go unnoticed (Ostrov & Hart, Emily, 2013). This is of particular concern in observation of classroom learning with technology, where specific uncommon events may be particularly essential to study, whether changes in student engagement (Andres et al., 2019) or critical but brief activities within a broader learning task (Bernacki, 2017; Jeong & Biswas, 2008).

Some researchers have instead focused on collecting qualitative data in the form of videos, and then coding those videos in depth (Kane, Kerr, & Pianta, 2014; Lehrer & Schauble, 2011). Videos can be re-watched an indefinite number of times, and coded in terms of a variety of constructs, often using complex coding schemes. Video addresses the "one shot" problem – at the cost of spending far more time coding data (Baker, Corbett, & Wagner, 2006). In cases where video captures rich dialogue between learners – or between learners and teachers – it is possible to make inferences about cognition as well as behavior. However, when learners work with a computer-based learning system (or individually, in general), this type of inference from video may be more challenging. A student may be working silently while complex activities are occurring internally. Even in video of classroom dialogue, we may not always have access to the complex reasoning (for instance, around self-presentation – Juvonen & Murdock, 1995) that impacts why students choose to say what they say.

The methods described so far provide detailed insights into *what* students are doing but may not always help understand *why* students make the choices they do. One method that gets at students' phenomenological understanding of why they do what they do is interviews. Interviews have a rich history in educational technology research. Perhaps the seminal work in this area was Schofield's book Computers and Classroom Culture (Schofield, 1995), which involved months of ethnographic embedding in schools, conducting interviews both during and outside of class. Shorter-term classroom interview studies of students using technology have investigated student attitudes towards specific educational technologies and student understanding of what they learned (Warren, Shen, Park, Baylor, & Perez, 2005; Yoon, Anderson, Lin, & Elinich, 2017). However, interview methods – like observational methods – suffer from the "one shot" problem. While interviews can be conducted retrospectively, students may not always recall their exact reasoning around a decision made half an hour or a week earlier, or the emotions surrounding that decision. Even if interviews are conducted during class, in real-time, it can be a challenge to identify relatively rare events of interest, within a class of 25 students working quietly on computers. Take, for instance, a researcher attempting to understand why a sequence of emotions occurs. Trying to spot a student going through a specific sequence of these emotions would be difficult for an observer to catch, especially while trying to monitor multiple students at the same time.

Finding the "right time" to interview has often been a logistical barrier to efficient qualitative data collection, however, new technologies may provide a solution to this long-standing problem. Specifically, the last decade has seen major advances in recognizing complex student behaviors and states from technology. Detectors have now been developed that can recognize engagement indicators (e.g., boredom, confusion, engaged concentration, frustration, off-task behavior, on-task behavior, etc.) from student interactions with learning software. These indicators have been validated to agree with human judgment for a wide variety of educational systems (Bosch & D'Mello, In Press; Botelho et al., 2017; D'Mello, 2018; Wixon et al., 2014). These detectors of student affect and behavior have been used in both fine-grained and coarse-grained analyses, from studying the characteristic shifts in engagement over a matter of seconds (D'Mello & Graesser, 2012) to studying how these measures correlate with long-term outcomes such as college attendance (San Pedro, Ocumpaugh, Baker, & Heffernan, 2014) or career choices

(Makhlouf & Mine, 2020). As such, these detectors can be used to identify critical moments in a learning process (Lodge, Panadero, Broadbent, & de Barba, 2018). Whereas previous work has considered using these detectors to drive in the moment automated intervention (e.g. Hutt, Krasich, Brockmole, & D'Mello, 2021; C. Mills, Gregg, Bixler, D'Mello, & D'Mello, 2021) or teacher reporting (Holstein, McLaren, & Aleven, 2017) in this work we use these detectors to drive data collection for qualitative research. This approach can in turn inform future design work to adapt to student behavior and affect.

In this chapter, we discuss an approach that attempts to address these limitations of existing qualitative methods – proposing a new way of conducting interviews, and a tool to facilitate this approach. In this approach, we leverage existing AIED technologies to target interviews, so that the depth of understanding that interviews facilitate can be combined with the ability to capture key moments in the learning process.

In the remainder of the chapter, we will detail the design of Quick Red Fox (QRF), a new research tool developed to facilitate targeted in-the-moment interviews. QRF is an open-source[1] server-client Android app. Specifically, QRF optimizes researcher time by directing interviewers to users that have just displayed an interesting behavior (previously defined by the research team). QRF integrates with existing student modeling technologies (e.g., behavior-sensing, affect-sensing, detection of self-regulated learning) to alert researchers to key moments in a user's experience. QRF listens for events (e.g., interaction patterns) and identifies moments of interest, prioritizes them, and directs interviewers, accordingly, allowing the interviewer to record their interview directly in the app along with relevant metadata (e.g., participant ID). We demonstrate the efficacy of this approach through a case study involving classroom research on student engagement and self-regulated learning. Finally, we discuss additional future applications of this tool in AIED technologies.

2 Design

QRF was designed based on principles for Minimal Attention User Interfaces (MAUIs). According to Pascoe et al., (2000), MAUIs for field work should consider four characteristics that are important to observational research: (a) the *dynamic user configuration* (i.e., working conditions of the fieldwork researcher), which are unlikely to include a desk or even a chair, (b) the *limited attention capacity* of the fieldworker, who necessarily needs to observe the object of their research, (c) the need for *high speed interactions,* should the research subject suddenly have a spurt of relevant activities that need to be documented, and (d) the *context dependency* needs of the field work, some of which (location, timestamps, etc.) can be automated by the system so that the researcher can focus on other things.

With ubiquity of mobile phones, they present an attractive option for developing apps for field work, with existing mobile apps already being used in the classroom (Ocumpaugh, Baker, Rodrigo, et al., 2015; Shapiro, 2011). Existing work has often linked classroom observations to student interaction data following the learning session in post hoc data processing. However, the

[1] https://github.com/pcla-code/QRF

QRF design requires knowledge of what students were doing before an observation or interview can be conducted, as data collection is targeted to events of interest, meaning that a stand-alone app would not be suitable. As such, QRF consists of two major components: (1) a server-side process that listens for events and assigns interviews, and (2) a client app (implemented in Java for Android mobile devices) that receives interview prompts and facilitates interview recordings. Interview triggers must also be defined, but are integrated with the learning environment rather than QRF (see section 2.1)

When designing the client side app for QRF, we align with the design considerations outlined by (Pascoe et al., 2000) for MAUIs. In doing so we acknowledge that the classroom is a complex environment and observations, and interviews need to be as simple as possible to record as researchers will likely have many other issues to negotiate. As such, QRF's design aligns carefully with an interview research protocol and facilitates context-aware coding (e.g., timestamps and the recording of triggering conditions) while allowing the researcher to focus their attention on the student. As described in more detail below, QRF displays details to direct them to a student, including the student's username and triggering conditions. The app also includes functionality for the researcher to take notes and/or record an audio file of an interview with the student. All data gathered is saved automatically, allowing the researcher to move on to the next observation with minimal effort devoted to the screen and reduced possibility of error.

In this section we outline the design for both major components of QRF, as well as give detail on the process required for Interview Triggers.

2.1 Interview Triggers

In order for QRF to appropriately detect events, interview triggers must be defined. Detection services for QRF identify key moments in students' learning processes in real time. They do this by parsing student log data or other available data streams. There are two components required to build a detection service: the individual detectors and (if necessary) the relevant patterns. First, automated detectors of constructs such as affect and behavior must have been trained beforehand, likely using a previously collected dataset often from the same learning system (i.e., Jiang et al., 2018). This process will also typically involve feature engineering, wherein a set of predictor variables was designed based on the student activity in the system. An example of a detector is a simple logistic regression model inferring boredom, which is a weighted combination of the selected features. The model output is then thresholded to predict a binary outcome (e.g., bored, or not bored).

Once the individual detectors are developed, they are embedded in the detection service code. If the researcher is interested in patterns of constructs/behaviors (e.g., bored then confused, vs. just bored), these patterns must be distilled from the data. These patterns are limited only by the detectors trained and are defined by the researcher for their specific study. Pattern detection then becomes an additional layer of detection that uses the output of individual detectors.

When a student starts interacting with the learning system, the detection service pulls in the student's activities from the system's interaction log data at a set interval of time (e.g., once every

20 seconds). Based on the student activity in that interval, feature values are derived. Depending on the type of feature, the service may have to keep a history of the student activity. For example, some features may only require student activities in the current interval, while others may need data from the start of a session or from the student's past sessions. These feature values are then fed into the individual detectors which in turn output predictions of relevant constructs (e.g., off-task behavior, frustration). According to the prediction in the current interval and past few intervals (depending on the length of the pattern), the detection service alerts the server of any detected pattern and its corresponding priority level (assigned beforehand). In addition, the service also keeps track of clearing the history and updating the past feature and prediction values as required. This is repeated regularly at the predefined time interval.

The detectors used to trigger interviews, are (though necessary) separate processes from the main QRF infrastructure. The machine learning (or other model definition) is outside the control of the app. Instead, QRF applies these methods to drive data collection. That means that researchers have the flexibility to develop detectors in whatever language they choose, relevant to their broader platform. For example, if studying an application written in Python, researchers may wish to integrate detectors also written in Python, whereas that may not be appropriate if studying an iOS application. The only requirement is that the detectors be able to send a package over a network (a feature present in almost all contemporary programming languages). This flexibility is crucial to the QRF design as it opens the door to a wider variety of future applications and research environments.

2.2 Server Side Platform

QRF's server side receives packages from the detectors (containing student ID, pattern/trigger, and priority) and assembles the interview queue. Each incoming pattern message is assigned a timestamp and inserted into a priority queue. The priority queue handles the selection & dispatch of pattern messages to the client side (described below). It sorts pattern messages (the detected triggers) based on their priorities, so that interviewers will be notified of the highest priority interview first. The sorting algorithm also includes two parameters to ensure that the same student is not interviewed too frequently: (a) '*maxInterviews'* (default value: 4), i.e., the maximum number of interviews that can be conducted with a student in the current session, and (b) *'interviewsGap'* (default value: 10 minutes), i.e., the minimum time gap between two successive interviews with a student. A pattern message lives in the priority queue until (a) it is sent to one of the QRF apps, or (b) the message expires, in which case it is no longer relevant to the student's current activities and is hence expelled from the queue.

The framework also includes a QRF Ruby library which handles the registration, initialization and communication between the server and the QRF app through a RabbitMQ message broker. Once the interviewer starts the QRF app and registers their handheld device on it, a direct communication line is established between the mobile app and the Betty's Brain server. Then pattern messages flow back & forth between the server and the app as requests for new patterns are made, accepted, or rejected by the interviewer.

2.3 Client Side

QRF's client side was implemented for Android devices using Java due to Android's strong support for app development and dissemination, and cost considerations (Android devices are often cheaper and more durable than other tablets). QRF synchronizes to internet time using an NTP server for logging purposes.

To make the process as smooth as possible, the app is heavily streamlined to require minimal interaction and thus allow the researcher to focus their resources elsewhere. The app was designed to avoid subpages that may be confusing to the researcher or result in erroneous recordings and aligns with several of the design principles from previous work discussed above. In addition to facilitating context awareness and limiting the number of times the researcher needs to enter the same information, QRF also allows the researcher to work with a small screen, thus reducing its obtrusiveness in the classroom. Finally, and perhaps most critically, QRF presents a user interface that aligns with the research protocol.

2.3.1 Set-up & Login

QRF requires a login with username and password, during which time it registers with the server-side application. It then requests information about the research session (e.g., classroom name, etc.) to verify which data to receive and provide annotations for later research. Once the Android device is registered with the server, messages between the two flow back and forth (e.g., requests for new interviews are sent to the server, which responds as soon as a prioritized trigger event is identified.). This process is completed once per research session (e.g., class period). Following set-up, the researcher is presented with the primary interface (see below).

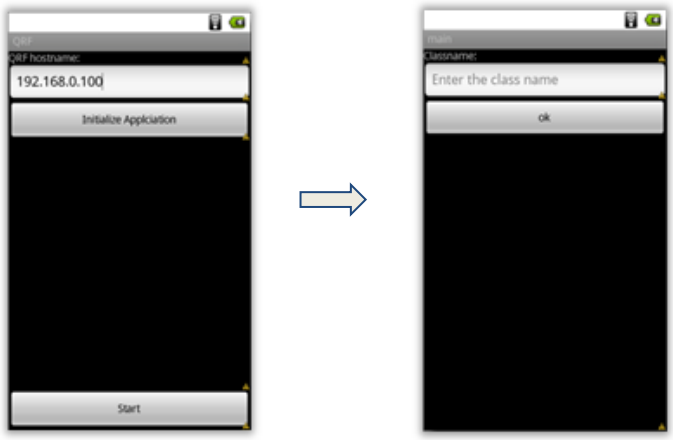

Figure 1. *QRF Login screens where researchers login (left) and enter the class session ID (right)*

2.3.2 Presentation of Student and Trigger Information

Figure 2 shows the primary QRF interface. From this screen, researchers receive information regarding which student to interview next, and can record said interview from the same screen. When a prioritized trigger event is identified, QRF presents this to the researcher by displaying

the User ID (i.e., what the student uses to log in with *or* a deidentified number, in cases where regulatory compliance requires it) at the top of the screen. Immediately below this information, QRF presents the trigger for the interviewer's use.

***Figure 2**. User Interface*

2.3.3 Interview Recordings & Notes

Should the researcher choose to interview the student, they can tap the "start recording" button, and an integrated recording system records the material to the SD card on the android device. The recording is then stopped when the interview taps the "stop recording" button. A timestamp is recorded at both the start and end of the interview. Interviewers can generate more than one interview with a student from the same trigger (this functionality was rarely used in the case study). If the interviewer notices additional useful information, they can type notes into a textbox. This functionality can also be used to add contextual notes, including why a student was not interviewed (e.g., they were already talking to their teacher or if they declined to be interviewed). The text box was also occasionally used to note if a neighboring student also participated in an interview. These notes are automatically saved to the interview log file to avoid any potential data loss.

2.3.4 Moving on (Next, Skip, End)

Once the interview and/or observation concludes, the interviewer selects the "next" button to advance. If the interviewer wishes to ignore a certain trigger (e.g., if a student should not be interrupted at that time) they can "skip" that observation. Both "next" and "skip" send a new trigger request to the server. If the class (or observation) session has ended, the interviewer can deregister from the server-side process by pressing "end." Notably, there is not a back button to return to a previous interview, in keeping with design principles that suggest not allowing real-time corrections if they might introduce cascading errors.

2.3.5 Data

Data produced by the QRF app records all of the transactions made by the interviewer in CSV format. This includes timestamps of all button presses, the interviewer currently logged in, the student being observed or interviewed, the recorded affect and behavior, the trigger that prompted the interview recommendation, and any other notes that the interviewer provides to the system. Also stored are audio files for each of the recorded interviewer. Files are names are stored with the transaction data to allow recordings to be linked to the appropriate participant. These data can then be analyzed by researchers, or synchronized alongside system data using the timestamps from the app.

3 Case Study

We demonstrate the use of, and opportunities made possible by, QRF with a case study examining self-regulated learning (SRL) behaviors as students interact with AIED technologies. Understanding self-regulation lends itself to this data collection paradigm, as research has shown that multiple data sources are needed to evaluate SRL strategies (Azevedo, Johnson, Chauncey, & Burkett, 2010; Winne, 2010), and that students may not be conscious enough of their approach to provide complete information in traditional self-report instruments.

This case study involved middle school students, interacting with a computer-based learning environment, Betty's Brain (Biswas et al., 2005), in an urban public school in Tennessee that serves approximately 700 5th-8th grade students. The school reports a student population that is 60% White, 25% Black, 9% Asian, and 5% Hispanic. Around 8% were enrolled in the free and reduced lunch program. 99 6th graders used Betty's Brain during the 2018–2019 school year as part of their regular science classes. No demographic data was collected from individual students.

Students interacted with the Betty's Brain software in the classroom, as part of their regular science instruction. Students interacted with the software for 45-50 minutes per day for eight days total. As students interacted, two interviewers were directed to students at key moments in the learning process by QRF. QRF listened for two types of events: affective sequences and behaviors related to self-regulated learning strategies (more details in section 3.2). Using previously integrated affect detectors (outlined below and in Jiang et al., 2018), we set the server-side process to listen for affect sequences that are aligned with theoretical models of affect dynamics in educational contexts (D'Mello & Graesser, 2012), and predefined action sequences relevant to SRL (Jeong & Biswas, 2008).

3.1 Betty's Brain

Betty's Brain uses a learning-by-teaching model (Biswas et al., 2005), where students must teach a virtual agent named "Betty" by creating a causal map of a scientific process (e.g., climate change or thermoregulation). Students then check their maps' validity by having Betty answer questions about scientific relationships, which she can only answer with information they have entered into the map. Betty demonstrates her "learning" by taking quizzes that are graded by a mentor agent, Mr. Davis. As students construct Betty's map, they must navigate a variety of learning resources,

including hypermedia resources about the subject matter and a teaching manual that explains how to represent causal reasoning. In this open-ended system, students choose how they build their maps, and how often they quiz Betty. They may also interact with Mr. Davis, who can support their learning and teaching endeavors (Biswas, Segedy, & Bunchongchit, 2016).

Betty's Brain presents a suitable environment for examining SRL behaviors for two reasons. Firstly, students choose when and how to perform each step of the learning process (both their own and Betty's) (Kinnebrew, Biswas, Sulcer, & Taylor, 2013; Roscoe, Segedy, Sulcer, Jeong, & Biswas, 2013). Indeed, the pedagogical agents in Betty's Brain are designed to facilitate the development of SRL behaviors by providing a framework for the gradual internalization of effective learning strategies. Secondly, students' interactions with Betty's Brain are logged to an online database with detailed timing information, enabling the microanalysis of student actions (Siadaty, Gasevic, & Hatala, 2016) for the measurement of SRL strategies.

3.2 Developing Interview Triggers

The affect and behavior detectors used in Betty's Brain were developed using the data collected in 2017 from 93 sixth-grade students recruited from four urban public schools in the southeastern region of the United States (Jiang et al., 2018). The predictors were derived from interaction log data (146,141 actions). The outcome labels (five affective states and off-task behavior) were collected from real-time classroom observation by two coders using BROMP (Baker et al., 2020). Interrater reliability had Cohen's Kappa >= 0.60 for every construct, between the two coders. The 5,212 observations (~56 per student) of affect were distributed as follows: 78% engaged concentration, 6% confusion, 4.6% frustration, 4.2% boredom, 2.9% delight. Off-task behavior comprised 10.2% of the total observations.

Using feature engineering, a set of meaningful features of student activity with Betty's Brain was generated as predictors for the automated detectors of affect and behavior. A total of 249 features were chosen from three categories. First, 41 basic features were designed including, time-based features (e.g., time spent reading a resource), count-based features (e.g., number of causal maps viewed), proportion features (e.g., percentage of effective actions), and descriptive features (e.g., average quiz score). Each feature was calculated in three different ways based on the time interval: since the student first started using the system (both total and normalized by time elapsed) and within the last 20-seconds. This led to a total of 123 basic features. Second, 30 sequence features were chosen based on the most frequent three action sequences (e.g., answer quiz -> read resource -> add concept). These were similarly conceptualized in three different ways (within a 20-second clip, thus far, and thus far divided by time elapsed), leading to a total of 90 sequence features. Third, 36 threshold features were developed using the optimized threshold values that led to the best correlation between the feature and student's post-test performance. The feature set, integrating across these three types of features, was then optimized to remove highly collinear features.

Finally, affect and behavior classifiers were built in RapidMiner 5.3 using selected features and binary outcome variables (e.g., off-task versus on-task, bored versus not bored). Due to the outcome labels being highly skewed, the data samples were resampled to balance the classes.

In addition, a forward selection algorithm was used to only pick features that led to better model performance. The Logistic Regression, Step Regression, Naive Bayes, C4.5 (J48), and RIPPER (JRIP) algorithms were used to train the model (a selection based on previous affect detectors). The models were evaluated using Cohen's kappa and AUC ROC on 10-fold student-level cross-validation. Experiments conducted by Jiang and colleagues (2018) showed that models with only basic features worked well for the detectors of engaged concentration, frustration, and delight. A combination of basic, sequence, and threshold features showed better results for confusion and boredom, and off-task behavior. In all cases model performance exceeded chance (average Kappa = 0.183, AUC ROC = 0.614). The final models were implemented in the server-side code to automatically detect students' affect and behavior based on the feature values that are continuously computed as the students interacted with Betty's Brain in real-time.

Patterns selected to be interview triggers were a mixture of theoretically selected patterns (e.g., the affective pattern of engaged concentration -> confusion -> frustration -> boredom developed by D'Mello and Graesser, 2012) or patterns empirically identified as important (e.g., a high correlation between sustained boredom and poor post-test performance). SRL patterns, including both strategic behaviors and affect transitions, obtained from the detector algorithms in the Betty's Brain student-end, are packaged as *[pattern, priority, student_ID]* messages and communicated to a Betty's Brain data server via a router.

3.3 Procedure

Two sessions of data collection were conducted, over the course of seven school days each (not all days were spent interacting with Betty's Brain). The first data collection was conducted in December 2018, and the second occurred two months later. Students completed two different scenarios within the Betty's Brain system in these two sessions, climate change (session 1) and thermoregulation (session 2). Minor alterations were made to the feedback system within the platform between the first and second session, based on the findings from the interviews conducted in the first session (Ocumpaugh et al., 2021). The alterations consisted mainly of feedback providing scaffolds and prompts to users who followed previously identified sequences of actions or affect while using the system. More specifically, conversation trees between the user and the virtual agents (either Betty or Mr. Davis) were adjusted to provide better guidance/hints and encouragement to students who ineffectively use within-platform resources or transition towards boredom. For example, students who incorrectly place or edit causal links on their concept map and take a quiz are prompted with one of three possible conversation scripts to help them recognize that the additions made to their concept maps were incongruent with the information they were given.

3.4 Data

As students interacted with Betty's Brain, automatic detectors of educationally relevant affective states (Jiang et al., 2018) and behaviors (Munshi et al., 2018), already embedded in the software, identified key moments in the students' learning processes (see above), either from specific affective patterns or theoretically aligned behavioral sequences. This detection was then used to prompt student interviews via QRF. Interviewers assumed a helpful but non-authoritative role

when interacting with students. Interviews were open-ended and occurred without a set script; however, students were often asked what their strategies were (if any) for getting through the system. As new information emerged in these open-ended interviews, questions were designed to elicit information about intrinsic interest (e.g., “What kinds of books do you like to read and why?”) were added. Overall, however, students were encouraged to provide feedback about their experience with the software, their goals while using the software, and their choices.

A total of 594 interviews (358 from session 1 and 236 from session 2) were conducted during classroom sessions, and audio recordings were simultaneously collected during these interactions. These interviews lasted no longer than 260 seconds. Audio files were collected from the QRF app and stored on a secure file management system available only to members of the research team. Three members of the research team manually transcribed the interviews, having agreed upon formatting and style. Metadata, including associated timestamps and recording IDs, were preserved, but student-level information was deidentified (i.e., each student was assigned an alphanumeric identifier, used across data streams). Transcriptions of each interview were organized together along with their respective unique timestamps, filenames, interviewer, and student ID of the student being interviewed.

3.5 Data Coding

Interview transcripts were then coded for qualitative categories that correspond with SRL constructs. These constructs were based on several theoretical frameworks and perspectives (Bandura, 1986; Boekaerts, 1999; Efklides, 2011), primarily focusing on the COPES model (Winne & Hadwin, 1998). These previously published works were examined during the development process for the interview codes to identify relevant constructs that would support deeper analysis and understanding of the data in relation to experiences of self-regulated learning. Previously published findings and models guided our approach to the development of codes and their subsequent analysis. It is important, however, to note that individual contexts, implicit biases, and perspectives of the members of the research team inherently influence the entire process of the study and interpretations from the data collected (Constas, 1992; Howe & Eisenhart, 1990).

The process followed a recursive, iterative method used in (Weston et al., 2001) that includes seven stages: conceptualization of codes, generation of codes, refinement of the first coding system, generation of the first codebook, continued revision and feedback, coding implementation, and continued revision of the codes (Weston et al., 2001). The conceptualization of codes included a review of related literature to capture meaningful experiences relevant to affect and SRL. Using grounded theory (Charmaz, 1983), we worked to identify categories that were (1) relevant to affective theory (i.e. D’Mello & Graesser, 2012) and self-regulated learning theory (e.g., Winne & Hadwin, 1998) and (2) likely to saliently emerge in the interviews. A draft lexicon and multiple criteria were generated for a coding system to help identify these constructs.

The draft lexicon was discussed with all members of the research team to build a common understanding of the constructs being examined and the features of the system. Feedback was provided by team members and the lexicon further refined. This process was repeated until the

entire research team had reached a shared understanding of the criteria and constructs being examined by the codebook.

A total of 12 interview codes were developed and applied to the interview data (see Table 1).

Table 1. Interview Coding Categories.

Code	Description
Difficult	Negative evaluations, confusion, or frustration while interacting with the platform
Helpfulness	Utility of within-game resources in learning, improvement, and positive evaluations of the resources
Interestingness	Interestingness of within-game resources in learning and a continued desire to use the platform
Strategic Use	Indicates a plan for interacting with the platform, notes changes in strategy or interaction with the platform based on experiences
Perceived Familiarity	The content has been previously learned or encountered and the student mentions ease in answering questions/ completing modules with familiar content
Positive Mr. Davis Attribution	Explicitly mentions interactions with Mr. Davis as positive or negative experiences
Positive Science Attribution	Explicitly mentions science in relation to books read, future careers, subjects in school, expressed interest, and overall evaluations of science
Positive Persistence	Expression of a desire for challenge and that the current task is a challenge, there is active pursuit of a goal, and repeated attempts to complete a step/problem
Procedural Strategy	Step by step approach to the learning activity, active use of within-platform tools and interaction with the system, references a previous step or step following current actions
Motivational Strategy	Explicit indication of an expected outcome from behaviors/actions, explicitly mentions a pursuit for mastery, contains a positive attribution/emotion towards completion of an activity, and mentions a desire to meet task demands
Task Adaptation	Indicates a comparison between learning modules/activities, describes a change in activity in response to achievement or failure with a previous action
Self-Confidence	Positive description of one's own progress or ability, implied monitoring of progress while learning, willingness to encounter challenges while learning, recognition of helpful resources

Following the production of the codebook and accompanying manual, multiple coders simultaneously coded a subset of the data to reach inter-rater reliability between them before applying the coding system to all of the transcripts. The resulting kappa values for each of the

interview codes are summarized in Table 2. Table 2 also summarizes the rates at which each interview code was observed across all students and all interviews. Throughout the coding process, the external coders met and clarified any concerns with authors of the codebook to avoid misinterpretation or miscoding of the data. As these qualitative codes are not mutually exclusive, a single interview may be coded under multiple categories.

Table 2. Inter-rater reliability and frequency of each interview code across all students and interviews.

		Student Level		Interview Level	
	κ	Study 1 (93)	Study 2 (89)	Study 1 (358)	Study 2 (236)
Difficult	.911	76.77%	73.74%	40.78%	59.32%
Helpfulness	.463	35.35%	63.64%	12.29%	50.85%
Interestingness	.726	8.08%	18.18%	2.23%	8.90%
Strategic Use	.911	78.79%	77.78%	48.88%	73.73%
Perceived Familiarity	.789	16.16%	10.10%	4.47%	4.24%
Positive Mr. Davis Attribution	.838	6.06%	48.48%	1.68%	30.08%
Positive Science Attribution	.837	21.21%	17.17%	6.70%	7.20%
Positive Persistence	.911	48.48%	66.67%	22.35%	52.12%
Procedural Strategy	.862	80.81%	79.80%	52.79%	75.85%
Motivational Strategy	.870	65.66%	72.73%	37.99%	62.29%
Task Adaptation	.808	75.76%	81.82%	45.81%	74.58%
Self-Confidence	.877	71.72%	79.80%	41.62%	70.76%

3.6 Impact on Scholarly Work

These interviews, the codes, and by extension, the QRF method have led to several scientific papers examining self-regulation, affect, and the interplay between the two. Bosch et al., (2021) used QRF interview transcripts to better understand metacognition and affect in AIED technologies. This work leveraged the affect data collected as well as automatically analyzing interview transcripts for markers of metacognition. Work by Hutt et al., (2021) showed that QRF interviews could be combined with log data for more effective predictions of future self-regulated learning behaviors than if predicted from log data alone. These predictions were subsequently predictive of future learning, more so than log data. Baker et al., (2021) used the data collected for an in-depth analysis of frustration in AIED systems, considering both the causes and the effects of frustration in different students. Ocumpaugh et al., (2021) demonstrated the potential of targeted interviews for identifying "pain points" in the AIED software and subsequent iterative design process, refining the design of Mr. Davis and Betty in ways that improved outcomes. Taken together, these articles demonstrate the wide potential of this data collection approach. By

facilitating the collection of rich, time synchronized interviews with theoretically grounded triggers, we can pursue a wide variety of research questions.

4 General Discussion and Conclusions

4.1 Summary

This chapter introduces the Quick Red Fox (QRF) handheld app for targeted classroom observation and the associated backend software that enables its functionality. QRF informs a researcher when predefined events of interest occur in the classroom and provides support for collecting interviews and collecting qualitative observations. The key innovation in QRF is the idea of targeting qualitative data collection in real-time, thus optimizing researcher time.

We then presented a case study that used QRF to study self-regulated learning and affect in multiple classes as students interacted with the Betty's Brain learning system. This case study demonstrates the potential of this approach, yielding several findings around the manifestation of both affect and self-regulation that would be difficult to obtain using previous methodology.

In general, QRF helps to address the "one shot" problem by alerting researchers to infrequent or unseen behaviors. Though this approach does not fully solve the "one-shot" issue – unexpected patterns may still be missed, and a single researcher still cannot be in two places at once – QRF comes closer to optimizing researcher time. In principle, any event that can be automatically detected (either through interaction analysis or more complex sensors) could be used as an interview trigger. Selecting triggers remains highly context-dependent and relies on researcher judgment, but the app and approach can support a wide range of use cases.

4.2 Applications

QRF can be used in a variety of education and training contexts. This is due to the combination of two factors. First, in classroom and training contexts, there are typically a number of people interacting with a given learning system at the same time. Second, research has consistently shown that internal cognitive and affective processes greatly influence how we learn but are often challenging to observe (Duckworth & Yeager, 2015; Linnenbrink, 2007). In K-12 educational research, there has been increased interest in understanding how complex internal processes such as emotion regulation, or engagement, impact learning, and how we might scaffold beneficial learning behaviors for students (Azevedo & Hadwin, 2005; Dumdumaya et al., 2017). QRF facilitates data collection that could allow interviewers to tap into a number of constructs that are crucial for effective learning but typically challenging to collect data on.

As such, QRF could be used for several potential applications, for both research and design. The case study above shows its potential usefulness for studying self-regulated learning and affect. In problem-solving domains such as mathematics and science, targeted interviews could be used to collect students' explanations of their problem-solving strategies, allowing researchers to better

understand student misconceptions. QRF could be used to interview students who become stuck in a puzzle game, to figure out if the learner is not perceiving a key part of the interface or task. Furthermore, QRF could be used when a learner is wheel-spinning (Beck & Gong, 2013), to see what hints or scaffolding could get them back on track. QRF could also be used to evaluate new AIED technologies as they are developed, interviewing students about their experiences as software is refined.

Beyond education, QRF or apps like QRF could be used in usability research. QRF might be used to trigger interviews when users make actions not initially expected by developers, or to better understand how users respond to error messages from the system. QRF could also be used to study usability outside the lab in real-world contexts, a crucial step for many projects, when real-world conditions may impact usability (Bevan & Macleod, 1994). For example, QRF may be useful in studying the usability of medical technologies (Acharya, Thimbleby, & Oladimeji, 2010), a field where researchers would want to limit the number of interviews so as not to distract the users from their primary task of caregiving. Similar to educational environments, there are often multiple interactions happening at any given time (multiple caregivers each with multiple patients) thus optimizing interviewer time would be critical.

Though we provide these sample applications, a key benefit of the QRF infrastructure is it can be used to address the "one shot" problem in almost any environment, providing researchers can detect the event of interest. Detection must be timely, and somewhat accurate, and similarly the environment must be suited to interviews (e.g., not a silent theatre). QRF leverages existing detection, likely machine learned models, but could also be triggered by other kinds of event such as rationally defined interaction patterns. Put simply, if you can define the event, and detect the event, you can interview after the event with QRF.

4.3 Limitations

The interviewing approach that QRF enables is not without its limitations. Many of those limitations center around the ways that QRF is targeted. QRF's targeting is based on pre-defined triggers. The approach is therefore limited by the triggers that are chosen. Interesting and useful opportunities may be missed if the research team was not aware in advance that a specific event would be important to study. This may occur either because of limited relevant theory or researchers' limited knowledge of the system. Therefore, it may be useful to conduct a round of more open observation or data analysis prior to commencing work with QRF. Similarly, if the detectors used are inaccurate, and do not correctly identify the moments of interest, then the method is severely weakened. That said, it will still facilitate an interview approach that avoids students being interviewed too frequently. But in this situation the data may not provide the same level of insight on specific events of interest as it would with accurate detectors.

Furthermore, even if the research team knows what is relevant and important to study, the approach may be limited by the quality of detection available. QRF research in the context of Betty's Brain was largely enabled by the availability of high-quality detectors of self-regulated learning and affect. Learning systems for which sophisticated detection is unavailable may find that there are limits to what can be studied using QRF. It may still be possible to identify when a

student spends substantially more time on a learning task than their peers or performs more poorly on a task relative to the average, but more complex constructs may be unavailable. In these cases, approaches such as clustering, sequence mining, or outlier detection may be used to provide more information for triggering interviews but may be unable to achieve the clarity of high-quality detectors of specific, well-understood constructs.

As such, an approach like QRF that focuses researcher time on key events is only as good as our ability to automatically detect that key event. Fortunately, the last decade has seen considerable progress within the educational data mining community on developing high-quality detectors of the types of constructs that might serve as triggers in QRF. A methodology like QRF's targeted interviews has only now become feasible now due to that progress.

4.4 Future Development

The next key step for QRF is expansion: to a broader range of constructs, and to a broader range of learning systems. Expanding the use of QRF in these fashions will naturally lead to enhancements to the app and infrastructure, to tailor their application for other uses. Potential extensions could include providing more information on the learner to the interviewer, and suggested questions for less experienced interviewers when the app is used at greater scale. All a platform needs to be used with QRF is a high-quality interaction data stream, and a server architecture where the communications architecture can be integrated. Our code for QRF is available online and fully open source, at https://github.com/pcla-code/QRF. We also invite researchers interested in using QRF to reach out to us to discuss potential collaborations.

Acknowledgments

This work was supported by NSF #DRL-1561567.

References

Acharya, C., Thimbleby, H., & Oladimeji, P. (2010). Human computer interaction and medical devices. *Proceedings of HCI 2010 24*, 168–176.

Andres, J. M. A. L., Ocumpaugh, J., Baker, R. S., Slater, S., Paquette, L., Jiang, Y., … others. (2019). Affect sequences and learning in betty's brain. *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, 383–390.

Azevedo, R., & Hadwin, A. F. (2005). *Scaffolding self-regulated learning and metacognition--Implications for the design of computer-based scaffolds*. Springer.

Azevedo, R., Johnson, A., Chauncey, A., & Burkett, C. (2010). Self-regulated Learning with MetaTutor: Advancing the Science of Learning with MetaCognitive Tools. In M. S. Khine & I. M. Saleh (Eds.), *New Science of Learning: Cognition, Computers and Collaboration in Education* (pp. 225–247). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4419-5716-0_11

Baker, R. S., 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*, *2002*, 29–36.

Baker, R. S., Nasiar, N., Ocumpaugh, J., Hutt, S., Andres, J. M. A. L., Slater, S., … Biswas, G. (2021). Affect-Targeted Interviews for Understanding Student Frustration. *Artificial Intelligence in Education 2021*.

Baker, R. S., Ocumpaugh, J. L., & Andres, J. (2020). BROMP quantitative field observations: A review. *Learning Science: Theory, Research, and Practice. New York, NY: McGraw-Hill*.

Bandura, A. (1986). The explanatory and predictive scope of self-efficacy theory. *Journal of Social and Clinical Psychology*, *4*(3), 359–373.

Beck, J. E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. *International Conference on Artificial Intelligence in Education*, *7926 LNAI*, 431–440. https://doi.org/10.1007/978-3-642-39112-5_44

Bernacki, M. L. (2017). Examining the cyclical, loosely sequenced, and contingent features of self-regulated learning: Trace data and their analysis. In *Handbook of self-regulation of learning and performance* (pp. 370–387). Routledge.

Bevan, N., & Macleod, M. (1994). Usability measurement in context. *Behaviour \& Information Technology*, *13*(1–2), 132–145.

Biswas, G., Leelawong, K., Schwartz, D., Vye, N., Davis, J., Belynne, K., … Katzlberger, T. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*. https://doi.org/10.1080/08839510590910200

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*. https://doi.org/10.1007/s40593-015-0057-9

Boekaerts, M. (1999). Self-regulated learning: Where we are today. *International Journal of Educational Research*, *31*(6), 445–457.

Bosch, N., & D'Mello, S. K. (n.d.). *Detecting Wandering Minds from Facial Features Across Multiple Domains in the Lab and the Wild*.

Bosch, N., Zhang, Y., Paquette, L., Baker, R. S., Ocumpaugh, J., & Biswas, G. (2021). Students' Verbalized Metacognition during Computerized Learning. *ACM SIGCHI: Computer-Human Interaction*, 12. Association for Computing Machinery. https://doi.org/10.1145/3411764.3445809

Botelho, A. F., Baker, R. S., & Heffernan, N. T. (2017). Improving Sensor-Free Affect Detection Using Deep Learning. In *Artificial Intelligence in Education*. https://doi.org/10.1007/978-3-319-61425-0_4

Charmaz, K. (1983). The grounded theory method: An explication and interpretation. *Contemporary Field Research*, 109–126.

Cleland, J. A. (2017). The qualitative orientation in medical education research. *Korean Journal*

of Medical Education, *29*(2), 61–71. https://doi.org/10.3946/kjme.2017.53

Cobb, P., Stephan, M., McClain, K., & Gravemeijer, K. (2001). Participating in classroom mathematical practices. *The Journal of the Learning Sciences*, *10*(1–2), 113–163.

Constas, M. A. (1992). Qualitative analysis as a public event: The documentation of category development procedures. *American Educational Research Journal*, *29*(2), 253–266.

D'Mello, S. K. (2018). What do we Think About When we Learn? In K. K. Mills, D. Long, J. Magliano, & K. Wierner (Eds.), *Deep Comprehension* (pp. 52–67). Routledge.

D'Mello, S. K., & Graesser, A. (2012). Dynamics of affective states during complex learning. *Learning and Instruction*, *22*(2), 145–157. https://doi.org/10.1016/j.learninstruc.2011.10.001

Duckworth, A. L., & Yeager, D. S. (2015). Measurement Matters: Assessing Personal Qualities Other Than Cognitive Ability for Educational Purposes. *Educational Researcher*, *44*(4), 237–251. https://doi.org/10.3102/0013189X15584327

Dumdumaya, C. E., Banawan, M. P., Rodrigo, M., Mercedes, T., Ogan, A., Yarzebinski, E., & Matsuda, N. (2017). Investigating the Effects of Cognitive and Metacognitive Scaffolding on Learners using a Learning by Teaching Environment. *International Conference on Computers in Education (ICCE)*.

Efklides, A. (2011). Interactions of metacognition with motivation and affect in self-regulated learning: The MASRL model. *Educational Psychologist*, *46*(1), 6–25.

Holstein, K., McLaren, B. M., & Aleven, V. (2017). Intelligent Tutors as Teachers' Aides: Exploring Teacher Needs for Real-Time Analytics in Blended Classrooms. *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, 257–266.

Howe, K., & Eisenhart, M. (1990). Standards for qualitative (and quantitative) research: A prolegomenon. *Educational Researcher*, *19*(4), 2–9.

Hutt, S., Krasich, K., Brockmole, J. R., & D'Mello, S. K. (2021). Breaking Out of the Lab: Mitigating Ming Wandering with Gaze-Based Attention-Aware Technology in Classrooms. *ACM SIGCHI: Computer-Human Interaction*.

Hutt, S., Ocumpaugh, J., Andres, J. M. A. L., Bosch, N., Paquette, L., Biswas, G., & Baker, R. S. (2021). Investigating SMART Models of Self-Regulation and their Impact on Learning. *Proceedings of the International Conference on Educational Data Mining*.

Jeong, H., & Biswas, G. (2008). Mining Student Behavior Models in Learning-by-Teaching Environments. *EDM*, 127–136.

Jiang, Y., Bosch, N., Baker, R. S., Paquette, L., Ocumpaugh, J., Andres, J. M. A. L., … Biswas, G. (2018). Expert Feature-Engineering vs. Deep Neural Networks: Which is Better for Sensor-Free Affect Detection? *Artificial Intelligence in Education*, 198–211. https://doi.org/10.1007/978-3-319-93843-1_15

Juvonen, J., & Murdock, T. B. (1995). Grade-level differences in the social value of effort: Implications for self-presentation tactics of early adolescents. *Child Development*, *66*(6),

1694–1705.

Kane, T., Kerr, K., & Pianta, R. (2014). *Designing teacher evaluation systems: New guidance from the measures of effective teaching project*. John Wiley \& Sons.

Kelley, T. R., Capobianco, B. M., & Kaluf, K. J. (2015). Concurrent think-aloud protocols to assess elementary design students. *International Journal of Technology and Design Education*, *25*(4), 521–540.

Kinnebrew, J. S., Biswas, G., Sulcer, B., & Taylor, R. S. (2013). Investigating Self-Regulated Learning in Teachable Agent Environments. In *International handbook of metacognition and learning technologies.* (pp. 451–470). New York, NY: Springer. https://doi.org/10.1007/978-1-4419-5546-3_29

Lehrer, R., & Schauble, L. (2011). Designing to support long-term growth and development. In *Theories of learning and studies of instructional practice* (pp. 19–38). Springer.

Linnenbrink, E. A. (2007). The role of affect in student learning: A multi-dimensional approach to considering the interaction of affect, motivation, and engagement. In R. Pekrun (Ed.), *Emotion in Education* (pp. 107–124). Elsevier. https://doi.org/10.1016/B978-012372545-5/50008-3

Lodge, J. M., Panadero, E., Broadbent, J., & de Barba, P. G. (2018). Supporting self-regulated learning with learning analytics. In *Learning analytics in the classroom* (pp. 45–55). Routledge.

Makhlouf, J., & Mine, T. (2020). Analysis of Click-Stream Data to Predict STEM Careers from Student Usage of an Intelligent Tutoring System. *Journal of Educational Data Mining*, *12*(2), 1–18.

Marcus-Quinn, A., & Hourigan, T. (2017). *Handbook on digital learning for K-12 schools*. Springer.

Meany-Daboul, M. G., Roscoe, E. M., Bourret, J. C., & Ahearn, W. H. (2007). A comparison of momentary time sampling and partial-interval recording for evaluating functional relations. *Journal of Applied Behavior Analysis*, *40*(3), 501–514.

Mills, C., Gregg, J. M., Bixler, R., & D'Mello, S. K. (2021). Eye-Mind reader: an intelligent reading interface that promotes long-term comprehension by detecting and responding to mind wandering. *Hum. Comput. Interact.*, *36*(4), 306–332. https://doi.org/10.1080/07370024.2020.1716762

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. *UMAP 2018 - Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, 131–138. https://doi.org/10.1145/3209219.3209241

Ocumpaugh, J., Baker, R. S., Rodrigo, M. M., Salvi, A., Van Velsen, M., Aghababyan, A., & Martin, T. (2015). HART: The human affect recording tool. *Proceedings of the 33rd Annual International Conference on the Design of Communication*, 1–6.

Ocumpaugh, J., Baker, R. S., & Rodrigo, M. M. T. (2015). *Baker Rodrigo Ocumpaugh monitoring protocol (BROMP) 2.0 technical and training manual*.

Ocumpaugh, J., Hutt, S., Andres, J. M. A. L., Baker, R. S., Biswas, G., Bosch, N., … Munshi, A. (2021). Using Qualitative Data from Targeted Interviews to Inform Rapid AIED Development. *Proceedings of the 29th International Conference on Computers in Education*.

Ostrov, J. M., & Hart, Emily, J. (2013). Observational Methods. In T. Little (Ed.), *The Oxford Handbook of Quantitative Methods in Psychology, Vol. 1* (pp. 286–319). https://doi.org/10.1093/oxfordhb/9780199934874.013.0015

Pascoe, J., Ryan, N., & Morse, D. (2000). Using while moving: HCI issues in fieldwork environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *7*(3), 417–437.

Roscoe, R. D., Segedy, J. R., Sulcer, B., Jeong, H., & Biswas, G. (2013). Shallow strategy development in a teachable agent environment designed to support self-regulated learning. *Computers and Education*. https://doi.org/10.1016/j.compedu.2012.11.008

San Pedro, M. O., Ocumpaugh, J., Baker, R. S., & Heffernan, N. T. (2014). Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. *Educational Data Mining*, 276–279.

Schofield, J. W. (1995). *Computers and classroom culture*. Cambridge University Press.

Shapiro, E. S. (2011). Behavioral observation of students in schools. In *Academic Skills and Problems fourth edition workbook* (pp. 35–56). New York: Guilford press.

Siadaty, M., Gasevic, D., & Hatala, M. (2016). Trace-Based Microanalytic Measurement of Self-Regulated Learning Processes. *Journal of Learning Analytics*, *3*(1), 183–214. https://doi.org/https://doi.org/10.18608/jla.2016.31.11

Warren, D., Shen, E., Park, S., Baylor, A. L., & Perez, R. (2005). Adult learner perceptions of affective agents: Experimental data and phenomenological observations. *Proceedings of the 2005 Conference on Artificial Intelligence in Education*, 944–946.

Wessel, D. (2015). The Potential of Computer-Assisted Direct Observation Apps. *International Journal of Interactive Mobile Technologies*, *9*(1).

Weston, C., Gandell, T., Beauchamp, J., McAlpine, L., Wiseman, C., & Beauchamp, C. (2001). Analyzing interview data: The development and evolution of a coding system. *Qualitative Sociology*, *24*(3), 381–400. https://doi.org/10.1023/A:1010690908200

Winne, P. H. (2010). Improving measurements of self-regulated learning. *Educational Psychologist*, *45*(4), 267–276. https://doi.org/10.1080/00461520.2010.517150

Winne, P. H., & Hadwin, A. F. (1998). Studying as self-regulated learning. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 277–304). Hillsdale, NJ: Erlbaum.

Wixon, M., Arroyo, I., Muldner, K., Burleson, W., Rai, D., & Woolf, B. (2014). The opportunities and limitations of scaling up sensor-free affect detection. *Educational Data Mining 2014*.

Yoon, S., Anderson, E., Lin, J., & Elinich, K. (2017). How augmented reality enables conceptual understanding of challenging science content. *Journal of Educational Technology \& Society*, *20*(1), 156–168.