Optional ERIC Coversheet - Only for Use with U.S. Department of Education Grantee Submissions

This coversheet should be completed by grantees and added to the PDF of your submission if the information required in this form **is not included on the PDF to be submitted**.

INSTRUCTIONS

- Before beginning submission process, download this PDF coversheet if you will need to provide information not on the PDF.
- Fill in all fields—information in this form **must match** the information on the submitted PDF and add missing information.
- Attach completed coversheet to the PDF you will upload to ERIC [use Adobe Acrobat or other program to combine PDF files]—do not upload the coversheet as a separate document.
- Begin completing submission form at https://eric.ed.gov/submit/ and upload the full-text PDF with attached coversheet when indicated. Your full-text PDF will display in ERIC after the 12-month embargo period.

GRANTEE SUBMISSION REQUIRED FIELDS

Title of article, paper, or other content

Intelligent Tutoring systems: a history and an example of an ITS for science

All author name(s) and affiliations on PDF

Last Name, First Name	Academic/Organizational Affiliation	ORCID ID
Gobert, Janice		
Sao Pedro, Michael		
Li, Haiying		
Lott, Christine		

Publication/Completion Date—(if *In Press* enter year accepted or completed) 2023

Check type of content being submitted and complete one of the following:

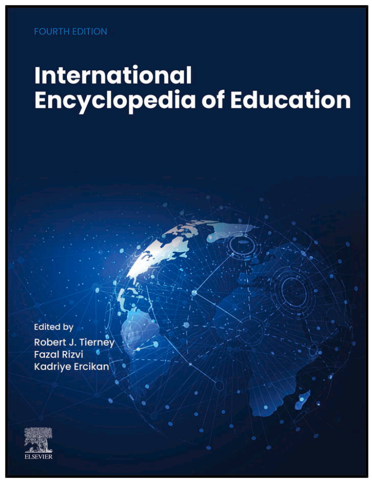
- ○ If article: Name of journal, volume, and issue number if available
- ○ If paper: Name of conference, date of conference, and place of conference
- X If book chapter: Title of book, page range, publisher name and location
- ○ If book: Publisher name and location
- ○ If dissertation: Name of institution, type of degree, and department granting degree

International Encyclopaedia of Education (Vol. 4, pp. 460-470), Elsevier.

DOI or URL to published work (if available) https://doi.org/10.1016/B978-0-12-818630-

Acknowledgement of Funding— Grantees should check with their grant officer for the preferred wording to acknowledge funding. If the grant officer does not have a preference, grantees can use this suggested wording (adjust wording if multiple grants are to be acknowledged). [Fill in Department of Education funding office, grant number, and name of grant recipient institution or organization.]

"This work was supported by **Institute of Education Sciences** U.S. Department of Education, through Grant R305A210432 to **Rutgers University.** The opinions expressed are those of the authors and do not represent views of the **Institute of Education Sciences** or the U.S. Department of Education.



From Gobert, J.D., Sao Pedro, M.A., Li, H., Lott, C., 2023. In: Tierney, R.J., Rizvi, F., Erkican, K. (Eds.), International Encyclopedia of Education, vol. 14. Elsevier.
https://dx.doi.org/10.1016/B978-0-12-818630-5.10058-2.
ISBN: 9780128186305

Elsevier

Intelligent tutoring systems: a history and an example of an ITS for science

Janice D. Gobert[a,b], Michael A. Sao Pedro[b], Haiying Li[c], and Christine Lott[a], [a] Rutgers Graduate School of Education, New Brunswick, NJ, United States; [b] Apprendis, LLC, Massachusetts, MA, United States; and [c] Iowa College Aid, IA, United States

Introduction to ITS

The origins of ITSs can be traced back to the 1950s and what is known as Computer Assisted Instruction, or CAI, which refers to the use of computers in the field of education. It was the introduction of "intelligence" to CAI in the 1970s that paved the way for a new genre known as ITS (Ahuja and Sille, 2013; Corbett et al., 1997). An ITS is a computer program that uses computational techniques to provide personalized instruction to students. This modern day intelligent tutor "knows" what to teach and, in theory, how best to do so. Perhaps most importantly, the tutor knows who is being taught, including their knowledge level, the misconceptions they may hold, their skills and abilities, and in some cases, their motivation and affective state. Research and development of ITSs weave together ideas from three different disciplines, including, computer science, education, and psychology (Alkhatlan and Kalita, 2018; Nwana, 1990). The interconnection between the three disciplines has advantages in terms of the breadth of research questions that can be conducted, which in turn, can contribute to advances in ITS technology. However, there also have been some disadvantages noted, namely that researchers need a firm grasp on all the relevant disciplines, and that those of different backgrounds may have competing goals, varying foci, and disagreement on frameworks and terminology (Nwana, 1990). Nonetheless, this area has generated technological innovations that have begun to successfully bridge the gap between human and computer when it comes to tutoring in education.

An important milestone in ITSs was when the target in terms of efficacy for student learning was set by a classic study by Bloom (1984), when he demonstrated that students who were guided by a human tutor performed two standard deviations better than those who studied in a traditional classroom or group learning environment. This quickly became the goal of ITSs in terms of efficacy at improving students' learning.

A number of meta-analyses have been conducted on the effectiveness of ITSs, since this is critical to their improvement (Kulik and Fletcher, 2016; Ma et al., 2014; VanLehn, 2011). For example, VanLehn (2011) compared the effectiveness of human tutoring, computer tutoring, and no tutoring, and found that ITSs can be nearly as effective as human tutors.

Ma et al. (2014) also conducted a meta-analysis to compare the learning outcomes of students who used ITSs and those who did not; here other factors such as type of ITS, instructional setting (individual, small group, etc.), and subject domain (math, physics, etc.) were included. In this meta-analysis, it was found that a learning environment that included an ITS was more effective than teacher-led or large group instruction, non-ITS computer-based instruction, and textbooks or workbooks. Further, positive effect

 International Encyclopedia of Education, 4th edition, Volume 14 https://doi.org/10.1016/B978-0-12-818630-5.10058-2

International Encyclopedia of Education, Fourth Edition, 2023, 460–470

sizes were found at differing levels of education, across different domains, and for different types of ITSs. Notably, however, no significant difference was found between ITS use and individual human tutoring or small group instruction. In another meta-analytic study by Kulik and Fletcher (2016), for 92% of the 50 studies examined, students who used ITSs outperformed students who did not. Further, the use of ITSs raised test scores, whether they were locally developed or standardized, over conventional classroom teaching environments. Thus, meta-analyses such as these provide evidence that suggests that ITSs can be effective at helping students learn.

Core components of ITS

It is widely accepted that typical intelligent tutoring systems (ITS) have four basic components (Corbett et al., 1997; Nwana, 1990). The *domain model*, also known as the expert model or cognitive model, which represents the knowledge, facts, concepts, and rules that belong to the domain being taught (e.g., physics or mathematics). The *student model*, which represents the students' knowledge, including the cognitive and, in some cases, the affective states of the students while they interact with the tutoring system. As the student responds to questions and solves problems, this information can be used to construct appropriate feedback. The *tutoring model*, also known as the pedagogical or teaching model, devises instructional strategies that can be used to provide customized feedback to students by integrating information from the domain/expert model and the student model. For example, the tutoring model can initiate actions, such as giving a hint to the student. Finally, *the user-interface model* is the interactive environment that the student uses to engage the system. It includes the graphics, text, and multimedia sources available to the student and presents the available actions a student can take during the tutoring process. Later in the entry we will provide linkages between each of these (domain model, student model, etc.) and Inq-ITS (**I**nquiry **I**ntelligent **T**utoring **S**ystem, inqits.com; Gobert et al., 2013, 2016, 2019; Mislevy et al., 2020), where relevant, for illustrative purposes.

History of ITS

Computer-assisted instruction

In its earliest stage, CAI was based on the work of B.F. Skinner and his principles of operant conditioning (Skinner, 1948). In a Skinnerian system, a student was presented with simple questions and then given the correct answer. The goal was to lead the student toward a desired end behavior. The series of "frames" were presented regardless of correctness because wrong answers did not help students learn the correct behavior, according to Skinner. In a frame-oriented system like this, designed to proceed regardless of how the student responds, there is a lack in the capacity to provide any personalized feedback, which later systems compensated for (Nwana, 1990). Further, in this linear style, all of the questions and answers had to be pre-determined, and as such, may not provide appropriate or helpful responses for students (Rickel, 1989).

In the 1960s, Crowder (1959) introduced branching, or intrinsic programs that no longer ignored student responses. Instead, they were used to determine the next frame to be presented to a student (Edwards, 1968; Nwana, 1990). In intrinsic programs, each response to a multiple-choice question is linked to another page in the program. When the student responded with the correct answer, they were presented with subsequent information. If the student selected the wrong choice, they were taken to a page that explained why that choice was incorrect, and then were prompted as needed to retry with a similar question (Edwards, 1968). These branching types of ITSs were also limited as students were confined to multiple choice questions and given no robust feedback, critical to improving learning.

Emerging in the late 1960s and early 1970s, generative systems elucidated and implemented the idea that tutoring material could be generated by the system itself; this soon followed as an improvement to the limited, step-wize nature of frame-oriented systems. Generative systems work by using textual templates, problem-generation grammars, and random numbers so the system can supply as many problems or examples as a student might need to improve learning (Nwana, 1990). An early example is Wexler's system, in which information is encoded in a network and then used to generate new questions and responses (Nwana, 1990; Rickel, 1989). Wexler grouped objects into classes that could then be linked to other classes, such as "family," which could be linked to "mother, father, and children" (Rickel, 1989). These systems were still insufficient however, in that they failed to interact with or adjust to the student's needs and responses. Moreover, they were restricted to drill-type exercises in well-defined domains such as mathematics that have structured content with objective answers (Nwana, 1990; Rickel, 1989). Further, generative systems have an inadequate representation of the students' domain knowledge, making it difficult to tailor feedback in a personalized or adaptive manner (Nwana, 1990; Rickel, 1989).

In early CAI, the student model component used in typical ITS architecture often lacked fine-grained user information and the functionality to provide an adaptive and personalized tutoring experience, which is ultimately the goal of ITSs. In one example, namely, the Leeds Adaptive Arithmetic System (Nwana, 1990), a rudimentary attempt at fulfilling the student model involved using a single integer to represent a student's competence level. In another example of an ITS with a simple student model, the KEYSTROKE model was able to determine the productivity of a user by counting their keystrokes in a basic task such as searching for information (Sleeman, 1985). Another issue, according to Sleeman, is that many early systems did not distinguish an expert from a novice in a given subject area and thus, did not lead to effective tutoring. Sleeman acknowledged that student modeling is an open-ended task and thus requires a system with the capability to interpret and respond to the diverse knowledge, skills, and abilities of students; however, in well-defined domains, the system can make more simplified assumptions about a user than in ill-defined domains. Later in the chapter we provide a more detailed section on student modeling, as it is a key factor in ITS development and efficacy in supporting students' learning.

From CAI to ITS

With a lack of fine-grained user knowledge (i.e., student model) in early CAI systems, experts soon realized the potential gains of incorporating "intelligence" into these systems. Self (1974) argued that to be most effective, CAI should include a representation of what is being taught, who is being taught, and how to teach it. These categories, respectively, correspond to the current ITS architecture of the domain model, student model, and tutoring model. These improved systems have a more sophisticated representation of the student model, one in which the current knowledge state is used to determine appropriate and personalized instruction (Rickel, 1989).

The first intelligent tutoring program, known as SCHOLAR, marked the introduction of AI to CAI, and ushered in a new generation of systems, which then became labeled as Intelligent Tutoring Systems (Nwana, 1990). SCHOLAR engaged students in dialog, allowing them to ask and answer questions (Carbonell, 1970). This system used a semantic network of units, which form a complex network of facts, concepts, and in some cases, procedures (Ahuja and Sille, 2013). Nodes in the network were tagged if the student recognized a concept. Further, this system exemplified a mixed-initiative approach, whereby the student as well as the tutor could ask (initiate) questions and dialog (Shute and Psotka, 1994). Since this system used a natural language dialog, if the system did not understand the student's question or response, the learning process was halted.

ITS developments from the 1980s to present

Buggy-based approach

By the 1980s, ITS development grew and various systems were developed with different foci in mind. One example is the buggy approach (Brown and Burton, 1978), which not only identifies a student's misconceptions, or bugs, but helps to explain why a student is making a mistake. One demonstration of a system based on the buggy approach is PROUST (Johnson and Soloway, 1985), which targets errors in students' computing of descriptive statistics, such as calculations of range and averages (Shute and Psotka, 1994). A report is submitted to PROUST after the student completes the program, and then the system locates errors and sends a diagnostic report (Shute and Psotka, 1994).

Model tracing

Another approach involves model tracing in which the system tracks a student step-by-step during a problem solving task and then compares this to the expert domain to provide personalized feedback as the student progresses (Merrill et al., 1992). The objective is to sequence the student's problem-solving actions within a complex problem space that may contain thousands of solutions (Corbett et al., 1997). One criticism of model-tracing, however, is the frequent interference from the tutor, which can limit a student's opportunity to learn from or reflect on his own mistakes (Merrill et al., 1992).

Case-based reasoning

Case-based reasoning (CBR) also appeared in the 1980s, which involves applying prior experiences and memories, or cases, to new information in order to interpret problems and find solutions (Kolodner, 1992). This can be applied to the human reasoning process as well as in the approach and design of ITS systems. Kolodner (1983) introduced the first CBR system, known as CYRUS, which is an intelligent fact retrieval system that stores and retrieves events of former Secretaries of State, Cyrus Vance and Edmund Muskie. If given a new fact, it will organize it within the existing database and can also call up a fact when asked, such as "When was the last time Vance was in Egypt?". Another example is CHEF, which created new recipes from recipes it already knew. By accessing a recipe with chicken and peas, it was able to substitute beef and broccoli and adjust the steps in the directions accordingly (Kolodner, 1992). Case-based reasoning has the potential to provide a more adaptive learning environment because it suggests approximate answers rather than providing an absolute response. This also makes it more suitable for ill-defined domains (i.e., philosophy), or environments in which there are too many rules or a rule that can be applied in many ways, such as game playing or programming (Ahuja and Sille, 2013; Shute and Psotka, 1994). Science inquiry is another ill-defined domain, and will be addressed later in the entry (Gobert et al., 2013).

Natural language processing

Another focus of development in ITSs in the 1980s was Natural Language Processing (NLP; Burstein, 2009; Burton, 1976; Chowdhury, 2003). NLP research has helped create a more natural flow in communication between a human and machine, or student and tutor by gathering information about how humans use and understand language and then using it to interact with the user (Chowdhury, 2003). An early example of a system built on NLP is SOPHIE (Burton, 1976), which applied semantic grammar techniques in a well-defined domain. SOPHIE's expert domain was electronic troubleshooting, which simulated faults on which the student could apply their theoretical knowledge to form hypotheses and attempt experiments to troubleshoot the problem (Nwana, 1990).

Discovery worlds and simulations

Another product of ITS development throughout the 1980s was discovery worlds (Ahuja and Sille, 2013) or simulations. A simulation is a model of scientific (or social) phenomena that represents the domain-specific properties and conceptual representations of that phenomena. A simulation may emulate some features of real phenomena but not others. Originally referred to as microworlds by Papert as a "… subset[s] of reality or a constructed reality so … as to allow a human learner to *exercise* particular powerful ideas or intellectual skills" (Papert, 1980, p. 204). For example, digital simulations enable learners to change aspects of phenomena, i.e., speed up, slow down, change size, or simplify aspects of real-world phenomena. Learners can explore phenomena, which are not possible to explore in real life due to their size and or time scale (Gobert and Clement, 1999), or engage in due to safety issues.

The goal with simulations is for students to use a computer simulation environment to learn content and skills on their own (Shute and Psotka, 1994). The notion of student-led discovery, however, is at odds with an ITS since the goal of ITS is to provide feedback to guide students' learning. For example, White (1984) created a system that allowed students to explore Newton's laws of motion in a discovery world (a simulation) by controlling a spaceship and navigating it toward a target or through a maze. While these types of systems allow for more freedom on the part of the student to guide their own learning processes, it does demand some knowledge and skill to operate within the microworlds. Students must know what to manipulate and how to do so, i.e., design an experiment or support a hypothesis, etc. (Shute and Psotka, 1994).

Mental models

Related to discovery worlds and simulations is the psychological construct invoked to describe the form of the cognitive representations that are constructed when learners engage in deep learning (Johnson-Laird, 1983). More specifically, model-based learning assumes that understanding requires the construction of mental models of the phenomena under study, and that all subsequent problem-solving, inferencing, or reasoning are done by means of manipulating or "running" these mental models (Gobert and Buckley, 2000). Using the notion of mental models, the 1980s saw systems that leveraged progressively more complex causal models presented in ITSs to, in turn, support the development of students' mental models (Ahuja and Sille, 2013). These systems, like QUEST (Qualitative Understanding of Electrical System Troubleshooting; White and Frederiksen, 1990), point out errors in students' thinking and guide students through increasingly complex microworlds. They then allow students to test their ideas by applying the set of laws to see if it predicts real-world events (Shute and Psotka, 1994).

Virtual reality

Going beyond simulations are more deeply immersive-type environments called virtual reality, another innovation in ITSs during the 1990s. These were influenced by situated learning, which emphasized the importance of the physical (and social) context on students' learning processes (Brown et al., 1989). With virtual reality, users can experience a simulated 3-dimensional environment in which they can move about and manipulate aspects of the environment to learn about phenomena as well as new skills (Shute and Psotka, 1994). Virtual reality provides learning opportunities and experiences in a similar manner to simulations, i.e., experiences that would otherwise not be possible or feasible. For example, Perez-Valle and Sagasti (2012) developed a system that allowed students to visit 16th century Spain and tour buildings and temples that no longer exist.

Learner control

Moving into the 1990s, one major issue among researchers in ITS development centered on the topic of learner control. On one side of the issue are those who believe the learners should be in control of their learning processes, while others believe that structure and guidance are necessary components to a learner's success. When applied to the field of ITS, more learner control equates to a learning environment in which students can explore without many restraints imposed by the system. In a program called Smithtown (Shute and Psotka, 1994), which teaches basic microeconomics, students can explore the environment freely and then design their own experiments by choosing variables, testing a hypothesis, and analyzing their data. The system provides a notebook and table to collect and organize data, and graphing tools to allow students to plot their data. For example, students can select a market to explore, such as gasoline, and then adjust the price, and analyze the effects on the market by making a graph reflecting the supply and demand (Shute et al., 1988). In science, the degree of learner versus system "control" used to guide students' activities in learning environments, particularly those involving simulations and/or microworlds, is a topic that has been hotly debated (Hmelo-Silver, 2004; Kirschner et al., 2006). Some have encouraged open-ended exploration by students, and others have offered guidance or structure within the microworld to promote optimized learning. Later, we describe how Inq-ITS scaffolds students by both structuring inquiry into phases and using hints provided by a virtual tutor to help the student when the system detects that the student is struggling with a particular inquiry competency.

Collaborative learning

Further research in education and technology suggested collaborative learning can also enhance learning outcomes in the context of ITSs, which became another focus of ITS development in the 1990s (Alkhatlan and Kalita, 2018). There are certain characteristics that must be in place to achieve success in a collaborative learning group, such as participation and performance analysis (Soller, 2001). For example, in one such system for learning fractions, students could use audio chat to talk through the problem; students were also assigned roles such as helper or problem solver to guide them through the interaction and promote accountability (Olsen et al., 2014).

Tutoring for affective states and engagement

By the end of the 1990s, an emphasis on the emotional state of the learner became a new focus of ITS development. This was in part due to the hypothesis that engagement can influence learning (Gobert et al., 2015). Further, by being able to infer the emotional state of the learner, the system can then respond more appropriately to the needs of the student both cognitively and psychologically (Picard, 1997). For example, student engagement can be tracked via logfiles, which are fine-grained traces of students' interactions in a learning environment for science (Gobert et al., 2015). There also exist various methods to capture the affective state of the student including a computer's camera and microphones, specialized mouses, and neuro-headsets, which collectively provide feedback regarding a student's facial expressions, voice, and physiological measures such as heart-rate and skin-conductivity (Alkhatlan and Kalita, 2018; D'Mello and Graesser, 2009). For example, in Auto-Tutor, a natural language ITS that has been used in domains such as computer literacy and physics, three sensors monitored students' affective state using posture patterns, videos of their face, and audio of their speech. This data was used to detect and differentiate emotions such as boredom, delight, and frustration (D'Mello and Graesser, 2009), and in turn was used to devise a targeted response, such as presenting an engaging task if the system detected a student's boredom. For more information on engagement and affective tutoring, see Andres et al. (2021), Baker et al. (2021), Henderson et al. (2021), Lester et al. (2019), and Mott et al. (2021).

Game-based ITSs

One of the more recent developments regarding ITSs is that of game-based learning, which also presupposes that students will learn more effectively if they are more deeply engaged in learning (Alkhatlan and Kalita, 2018). Though, a more apt statement in our opinion is that engagement is a *necessary but not sufficient condition* for learning, since deep learning is influenced by a variety of factors (Gobert, 2022). Rai and Beck (2012) created a math tutoring system with game-like elements, such as naming the monkey character students interact with or receiving badges after completing a level of learning. Despite many benefits to game-based systems, such as maximization of student participation, there also exist potential drawbacks associated with these types of systems (Alkhatlan and Kalita, 2018). Games often demand high cognitive load of users, which can lead students to feel overwhelmed and, in turn, diminish long-term learning of the material (Rai and Beck, 2012); this focus may be better utilized on content rather than the dynamics of the game.

Student modeling

Earlier we highlighted the core components of ITSs, including the student model. It is important to note that the adaptability and personalization of an ITS ultimately relies on the construction of an effective student model. By this, we mean that the student model leads to an understanding of the user by drawing on relevant information including a student's performance, misconceptions, prior knowledge, and in some cases, their personality characteristics. In brief, there are a number of approaches to building a student model including the overlay model, perturbation model, machine learning, cognitive theories, fuzzy logic, and Bayesian networks, with most ITSs using some combination of these (Chrysafiadi and Virvou, 2013), though it is beyond the scope of this entry to articulate the differences between all of these development approaches.

Three approaches that are noteworthy due to their popularity in the development of ITSs are the overlay model, stereotype model, and perturbation model (Chrysafiadi and Virvou, 2013). The overlay model was developed by Stansfield et al. (1976) and views the student's knowledge as a subset, or overlay, of the expert's knowledge, as represented by the domain model. The focus of the ITS then, is to minimize the gap in knowledge between the expert's and the novice's knowledge, skills, etc. (Alkhatlan and Kalita, 2018). The perturbation model is often viewed as an extension of the overlay model because it in part is a subset of the expert's knowledge, but unlike the traditional overlay model, it incorporates misconceptions or incorrect beliefs into this knowledge gap (Alkhatlan and Kalita, 2018). The stereotype model categorizes students based on shared characteristics, or stereotypes in order to guide them through an appropriate task or level based on their current knowledge state. An ITS may classify a new user into a category based on their level of knowledge about the content and can then infer information about the user based on other shared characteristics of that particular stereotype (Chrysafiadi and Virvou, 2013).

There are certain drawbacks, however, to each of these approaches. The overlay model is limited in that it does not take into account the learner's behavior and personality characteristics or the way users integrate new information. The perturbation model has demanding computational requirements and is limited in that the predetermined "bugs" in the system may not apply to some students. The stereotype model is inflexible in that the stereotype classes are pre-set by the system designer, which is time-consuming, and the users must be "of a type" that allow for discrete categorization in order to apply the stereotypes (Chrysafiadi and Virvou, 2013).

Machine learning/data mining

Through the early 2000s, the overlay, stereotype, and perturbation models were the most popular student modeling approaches. However, due to the limitations described above, a newer more adaptive and automated approach, known as machine learning or data mining, gained momentum (Chrysafiadi and Virvou, 2013). Though the term dates back to the 1980s, by the early 90s it was viewed as a sub-process of Knowledge Discovery in Databases or KDD (Coenen, 2011). Machine learning fulfills the need to automate the processes by which a user's actions and behaviors can be observed (Chrysafiadi and Virvou, 2013; Luan et al., 2020). Machine learning analyzes large amounts of data in order to extract meaningful information (Baker and Yacef, 2009; Coenen, 2011), and it's techniques allow for better construction and greater adaptability of the student model (Ahuja and Sille, 2013). In the 1990s, machine learning saw a decline and then a resurgence because of the need for large data sets, labeled data, and computational complexity (Webb et al., 2001). Presently, machine learning is increasingly used for the explanation, prediction, and description of student behavior (Yang et al., 2021), due in part, to an increase in internet usage and information retrieval technologies (Webb et al., 2001) of student data.

Machine learning techniques have mainly been used in two areas of student modeling research. First, to induce a student model from many student behaviors, and second to create a bug library of student models. For instance, Baker (2007) constructed a machine learning model that was able to detect when a student was off-task. Baker et al. (2012) detected emotions of students based on log data of actions with an Algebra ITS system, rather than with physical sensors that other affect detection systems may use. Finally, Baker et al. (2006) used data mining techniques to detect students who were trying to game the system by getting to a bottom-out hint rather than engaging in deep learning; students were then given supplementary exercises based on the ones they attempted to bypass. Gobert et al. (2015) used machine learning on students' logfiles within a science inquiry assessment and learning environment to detect students' disengagement from learning; Gobert et al. (2012, 2013) also used machine-learning for assessment of students' inquiry competencies.

The move towards real time performance assessment & real time scaffolding of inquiry competencies

As addressed earlier in the entry, many ITSs have been designed for well-defined domains. By this, we mean, domains that have one correct answer, such as ITSs for math problems (Corbett and Anderson, 1995; Feng et al., 2009; Gertner and van Lehn, 2000) like the addition of $2 + 3$, or double-digit addition, etc. For domains in which there is one correct answer, the assessment of students' knowledge and skills is fairly straight-forward, and computational techniques including machine-learning, described above, are not typically necessary. In these cases, rather, knowledge-engineering (cf. Koedinger and MacLaren, 2002) is used for both assessing students and scaffolding them. Briefly, in knowledge engineering, a domain expert constructs by hand a cognitive model to capture what it means to demonstrate a sub-component based on the constructs of the domain and/or cognitive task analyses (Ericsson and Simon, 1980), then techniques such as model tracing are used to compare students' knowledge and skills to that of an expert, and intervene when the system detects a difference between the two. Additionally, there are science domains, such as Physics and Chemistry, which include mathematics in their underlying formulas, equations, etc., and they too can use knowledge-engineering for these mathematical aspects of the domain. Some examples here are VanLehn's et al. (2005) Physics Tutoring system for college level Physics problems, which helps students by giving them assistance as they set up free-body diagrams, write equations, and solve them, and Walsh et al.'s (2002) intelligent tutoring system for balancing chemical equations.

Science inquiry, which is the means by which a learner engages in a series of tasks related to forming a testable hypothesis, conducting an experiment to test a hypothesis, analyzing a resulting data, and then communicating findings, is an ill-defined domain (Kuhn, 2005). It is ill-defined because there are many ways in which students can complete a task both productively as well as unproductively. Thus, both scaffolding and the assessment of students' competencies that is required in order to scaffold students (as is the goal of ITSs), is much more complex. Described next in this entry is Inq-ITS Inquiry-Intelligent Tutoring System (inqits.com; Gobert et al., 2013) in order to exemplify an ITS for science inquiry, and how machine learning is used in an ITS of this type. We also use Inq-ITS to describe the core components described above (where applicable for Inq-ITS). Lastly, we provide an overview of Inq-ITS scaffolds and an overview of some results of our scaffolding research.

Inq-ITS is a computer-based system, i.e., an ITS, that provides both real time assessment and scaffolding of science inquiry practices as described in national and international science frameworks, including the Next Generation Science Standards (NGSS, 2013). In brief, using simulations and a series of inquiry tools, widgets, etc., students formulate hypotheses and questions, test them, interpret their data with graphs and other mathematical means, warrant their claims with evidence, and communicate their findings. As students conduct their investigations, the work products they create and processes they use by conducting their investigations are automatically assessed using integrated machine-learning as well as knowledge-engineering, where relevant (Gobert et al., 2013, 2016, 2017, 2019; Mislevy et al., 2020). In addition, Inq-ITS supports teachers' real time assessment and instruction of inquiry practices (NGSS, 2013) using real time reports and actionable alerts via a dashboard called Inq-Blotter (Dickler et al., 2021; Gobert, 2019; Gobert et al., 2019; Gobert and Sao Pedro, 2016; Sao Pedro, 2015, 2016), though these are not the focus of the present chapter (see Dickler et al., 2021 for details on Inq-Blotter).

Inq-ITS was explicitly designed to prioritize the assessment and scaffolding of inquiry rather than the learning of science, which many other curriculum-focused inquiry systems emphasize (cf., Linn and Hsi, 2000). Inq-ITS was informed by the assessment triangle, which includes cognition, observations, and interpretation (Pellegrino et al., 2001), and by Evidence-Centered Design (Mislevy et al., 2020) in which the system elicits and collects evidence of students' proficiencies at inquiry practices (Gobert and Sao Pedro, 2017), i.e., this is performance assessment. In brief Evidence-Centered Design (ECD; Mislevy et al., 2009, 2020) is a design framework that helps specify all the key elements of an assessment system and an intelligent tutoring system, including the domain model, the interface model, the student model, and the tutoring model. ECD also specifies the fine-grained details and the computational processes (critical to interpretation of student data) that are to be carried out in the assessment system in order to generate fine-grained, rigorous assessment data on students' competencies, and in turn, scaffolding. This approach is different from the assessment of rote knowledge of facts or formulas, which can be assessed via multiple choice items. Rather, performance assessment of science inquiry is done as students engage in rich, authentic inquiry tasks with simulations so that the system can act *intelligently* to support students' learning.

Inq-ITS builds on the inquiry and assessment research of others (some of which was reviewed earlier) and forges new ground for inquiry assessment and scaffolding of science with its application of computational analytics including educational data mining, knowledge-engineering, and natural language processing (Gobert et al., 2013, 2016, 2017, 2019). The machine-learned algorithms allow the system to better capture what kinds of difficulties students have with inquiry, and then based on students' difficulties, Rex, a cartoon dinosaur that acts as a virtual pedagogical agent, jumps in to help the student in real time (See Fig. 1 below). These algorithms were built on diverse samples and tested for generalizability, which ensures equity with respect to the new students who later use these systems (Sao Pedro et al., 2013b).

To illustrate how the system works, as well as how the architectural components of an ITS are represented, a small vignette is presented. An eighth grade teacher, Mrs. Donahue, is having her class conduct an inquiry with Inq-ITS' States of Matter Physical Science virtual lab (See Fig. 1, left). In this lab, students determine the effects of the independent variables (e.g., level of heat, amount of substance) on the dependent variables (e.g., time to melt, temperature when melted) by forming a hypothesis, collecting data to test their hypothesis using the simulation, analyzing their data, and communicating their findings.

At a high level, students engage in virtual inquiry with our widgets and a domain-based simulation of a science phenomenon; this includes, forming a question/hypothesis, collecting data with a simulation, interpreting data, warranting claims with data, using mathematics to develop deeper understandings of the phenomena under study and communicating findings about the investigation (Gobert et al., 2012, 2013, 2016, 2019). The simulation environment represents the *domain model* of the science phenomena under investigation, while the simulations, widgets and clickable options presented to students in Inq-ITS demonstrate the *user-interface model*. The Inq-ITS activities leverage the logging infrastructure to store all students' interactions (logfiles) at a fine-grained level, including students' interactions with widgets, representational tools, and simulations, and timestamps for those interactions (Mislevy et al., 2020; Gobert et al., 2012; Sao Pedro et al., 2013a); these are used to construct a student model of the students' competencies across the full range of inquiry practices, which informs the tutoring model.

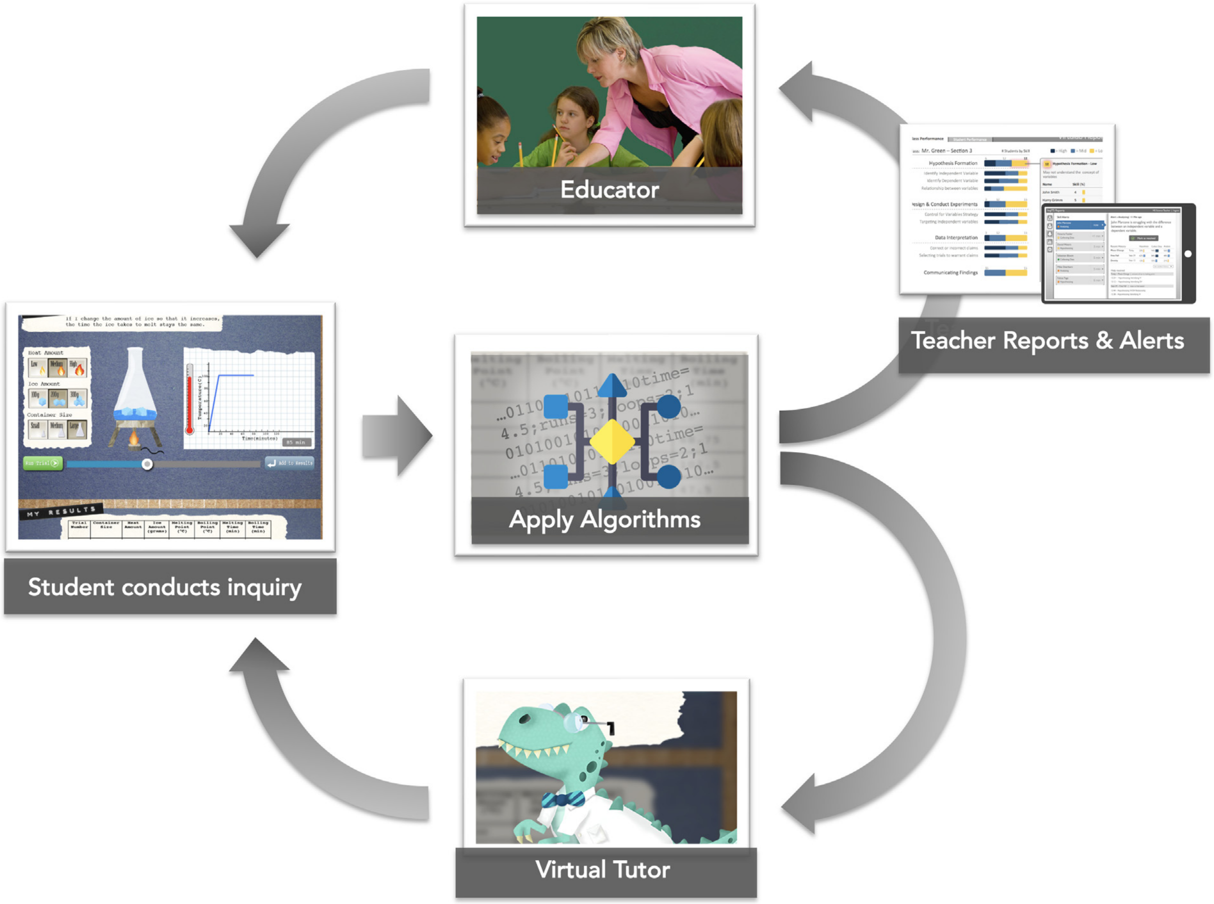


Fig. 1 As students conduct inquiry with Inq-ITS, their inquiry is automatically assessed by underlying algorithms. Rex reacts in real time to support students' learning in the moment. Real time reports and alerts are also generated in real time to support teachers' assessment and instructional needs.

Having a logging infrastructure that collects all students' interactions provides many benefits for intelligent tutoring systems. In terms of *research*, the system can effortlessly and accurately monitor and record every student action; this is important for testing the efficacy of scaffolds in an ITS. Regarding *assessment*, by seamlessly doing assessment in real time and at a fine-grained level, it can assess and remediate students' specific inquiry difficulties in the rich contexts in which they are learned (Mislevy et al., 2003). In terms of *instruction*, it can seamlessly integrate assessment with instruction so additional class time is not needed; that is, the tutor Rex can respond to individual students in real-time as they conduct inquiry, when it is most beneficial for learning; and it can provide formative data and actionable alerts to teachers about his/her students easily and automatically via our dashboard Inq-Blotter (Dickler et al., 2021; Gobert, 2019; Gobert et al., 2019; Gobert and Sao Pedro, 2016; Sao Pedro, 2015, 2016). With regard to *scalability*, because these materials are web-based, virtually any teacher with access to the web can use the ITS.

Furthermore, there are great advantages for analyzing rich logfiles of students' learning using machine-learning, as is done in Inq-ITS. These are often referred to as the 4Vs (Laney, 2001), namely: the large volume of log data, the velocity at which it is generated, the veracity in fine-grained logfiles, and the variability in students' logfiles since there are many ways in which inquiry can be done both productively and unproductively. However, theory is needed to both distill and aggregate data (Quellmalz et al., 2012), and to design categories a priori so that results are pedagogically meaningful (Gobert et al., 2013). In Inq-ITS, algorithms are used to derive rules based on cognitive task analysis (Ericsson and Simon, 1980) of student interaction data and/or human-classified labels of those data that were developed from the national inquiry frameworks (NRC, 1996; NGSS, 2013) as well as data (Gobert and Sao Pedro, 2017) and literature on students' difficulties with conducting inquiry (cf. Kuhn, 2005). Then, machine learning is leveraged to derive algorithms for process-oriented practices that can be exhibited in several different ways, e.g., collecting data (Sao Pedro et al., 2013a,b). The results are aggregated into performance indicators that provide evidence of students' proficiencies for each inquiry practice and their respective sub-skills, and the system is instrumented to jump in to provide real time feedback to students when the system detects that students are struggling on the inquiry competencies of interest.

Inq-ITS delivers scaffolds to students in text format via the pedagogical agent, Rex (see **Fig. 1**). This proactive scaffolding approach helps to support students in their inquiry processes (Schauble, 1990; deJong, 2006) by preventing students from engaging in unproductive behaviors (Buckley et al., 2006; Sao Pedro, 2013). This proactive approach is also important because students may not be aware that they need help (Aleven and Koedinger, 2000; Aleven et al., 2004).

Scaffolds in Inq-ITS

In Inq-ITS, the primary scaffolding strategies used by the pedagogical agent are explanations provided by Rex (see **Fig. 1**). Such scaffolding is based on principles from cognitive psychology (McKendree, 1990), and this approach to scaffolding has been shown to benefit learning in other domains (Beck and Mostow, 2008). The scaffolds involve multiple steps or levels (cf., Wood and Wood, 1999), similar to Cognitive Tutors (Anderson et al., 1995). In Inq-ITS, scaffolds are triggered by our assessment algorithms (cf., Gobert et al., 2013, 2016; Sao Pedro, 2013) when the systems detects students' difficulties with inquiry. An example set of the progression of scaffolds that is triggered when a student is not designing controlled experiments, a key skill in science inquiry (Kuhn, 2005), is presented below.

1. Orienting scaffolds are designed to orient the student to the current task within the inquiry cycle since students have difficulties monitoring their progress (de Jong, 2006). For this, Rex may state that in the experiment phase, data should be collected to help support or refute a stated hypothesis.
2. Conceptual scaffolds are designed to provide a high-level hint to the student, e.g., if a student has collected confounded trials (Sao Pedro, 2013), Rex may ask, "How will you know about the effects of x on y, given your trials?"
3. Procedural scaffolds are designed to provide information about the procedure to use on the current task, e.g., Rex may instruct the student to "construct a controlled trial, *relative* to your last trial run." Explicitly teaching this strategy has been shown to be effective (Klahr and Nigam, 2004; Sao Pedro et al., 2009, 2010).
4. Instrumental scaffolds are designed to tell the student exactly what to do on the current task, i.e., a "bottom-out hint" (cf. Aleven and Koedinger, 2000). For example, Rex may instruct the student to "conduct a set of trials in which you change only one variable while keeping the others the same."

Using scaffolds like these, Gobert and her team have demonstrated the effectiveness of our assessment and scaffolding approach at the middle school level for several NGSS practices, including forming questions, planning and carrying out investigations, analyzing and interpreting data, and warranting claims in which scaffolding helped students both acquire and transfer these practices to new Inq-ITS topics over long periods of time, tested up to 170 days (Li et al., 2018, 2019a,b; Gobert et al., 2018; Moussavi et al., 2016; Moussavi, 2018; Sao Pedro et al., 2013a,c).

Conclusions

In summary, the last half century has seen dramatic changes in ITS developments and innovations, enabling ITSs with new capacities, which greatly impacts learning. Important among them is fine-grained student modeling. This, in turn, allows for fine-grained assessment, diagnosis, and personalized tutoring.

This entry also highlights how another innovation, namely machine learning, can be used in an ITS, specifically Inq-ITS. Machine learning in Inq-ITS is used to elicit students' inquiry performances while they engage in inquiry, assess their competencies in real time, and trigger scaffolding to them in real time in order to help them learn the science competencies expected by national frameworks (NGSS, 2013). Furthermore, well-designed, targeted scaffolds that are based on fine-grained, rigorous assessment of students' competencies *can* lead to robust student learning even for ill-defined domains such as inquiry. In sum, this approach to assessment and scaffolding can and should inform the design of assessment and intelligent tutoring systems so that systems are instrumented to assess and support *all* students, regardless of zip code, ELL status, etc. Furthermore, it is imperative that technological resources be leveraged so that ITSs are scalable to provide learning benefits far beyond what an individual teacher can provide.

Acknowledgments

This research was funded by the National Science Foundation (NSF-IIS-1629045, NSF-DRL-1252477, NSF-IIS- 1902647) and the US Department of Education (R305A210432, R305A120778, ED-IES-15-C-0018, ED-IES-16-C-0014). Any opinions, findings, etc., expressed are those of the authors and do not necessarily reflect those of the funding agencies.

References

Ahuja, N.J., Sille, R., 2013. A critical review of development of intelligent tutoring systems: retrospect, present and prospect. Int. J. Comput. Sci. Issues 10 (4), 39.
Aleven, V., Koedinger, K., 2000. Limitations of student control: do students know when they need help? In: Gauthier, G., Frasson, C., VanLehn, K. (Eds.), Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000. Springer-Verlag, Berlin, pp. 292–303.
Aleven, V., McLaren, B., Roll, I., Koedinger, K., 2004. Toward tutoring help seeking. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (Eds.), Proceedings of the Seventh International Conference on Intelligent Tutoring Systems, ITS 2004. Springer-Verlag, Berlin, pp. 227–239.
Alkhatlan, A., Kalita, J., 2018. Intelligent Tutoring Systems: a Comprehensive Historical Survey With Recent Developments arXiv preprint arXiv:1812.09628.
Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R., 1995. Cognitive tutors: lessons learned. J. Learn. Sci. 4 (2), 167–207.
Andres, J.M., Hutt, S., Ocumpaugh, J., Baker, R.S., Nasiar, N., Porter, C., 2021. How anxiety affects affect: a quantitative ethnographic investigation using affect detectors and data-targeted interviews. In: International Conference on Quantitative Ethnography. Springer, Cham, pp. 268–283.
Baker, R.S., Yacef, K., 2009. The state of educational data mining in 2009: a review and future visions. J. Educ. Data Min. 1 (1), 3–17.
Baker, R.S., Corbett, A.T., Wagner, A.Z., 2006. Human classification of low-fidelity replays of student actions. In: Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems, vol. 2002, pp. 29–36.
Baker, R.S.J., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Aleven, V., et al., 2012. Towards sensor-free affect detection in cognitive tutor algebra. In: Yacef, K., Zaïane, O., Hershkovitz, H., Yudelson, M., Stamper, J. (Eds.), Proceedings of the 5th International Conference on Educational Data Mining, p. 126e133. Chania, Greece.
Baker, R.S., Nasiar, N., Ocumpaugh, J.L., Hutt, S., Andres, J.M., Slater, S., Schofield, M., Moore, A., Paquette, L., Munshi, A., Biswas, G., 2021. Affect-targeted interviews for understanding student frustration. In: International Conference on Artificial Intelligence in Education. Springer, Cham, pp. 52–63.
Baker, R.S., 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1059–1068.
Beck, J.E., Mostow, J., 2008. How who should practice: using learning decomposition to evaluate the efficacy of different types of practice for different types of students. In: Intelligent Tutoring Systems. Springer Berlin Heidelberg, pp. 353–362.
Bloom, B.S., 1984. The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. Educ. Res. 13 (6), 4–16.
Brown, J.S., Burton, R.R., 1978. Diagnostic models for procedural bugs in basic mathematical skills. Cognit. Sci. 2 (2), 155–192.
Brown, J.S., Collins, A., Duguid, P., 1989. Situated cognition and the culture of learning. Educ. Res. 18 (1), 32–42.
Buckley, B., Gobert, J., Horwitz, P., 2006. Using log files to track students' model-based inquiry. In: The Proceedings of the Seventh International Conference of the Learning Sciences (ICLS). Erlbaum, Mawah: NJ, pp. 57–63.
Burstein, J., 2009. Opportunities for natural language processing research in education. In: International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Berlin, Heidelberg, pp. 6–27.
Burton, R.R., 1976. Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems. BBN Report No. 3453. Bolt Beranek & Newman, Cambridge, MA.
Carbonell, J.R., 1970. AI in CAI: an artificial-intelligence approach to computer-assisted instruction. IEEE Trans. Man Mach. Syst. 11 (4), 190–202.
Chowdhury, G.G., 2003. Natural language processing. Annu. Rev. Inf. Sci. Technol. 37 (1), 51–89.
Chrysafiadi, K., Virvou, M., 2013. Student modeling approaches: a literature review for the last decade. Expert Syst. Appl. 40 (11), 4715–4729.
Coenen, F., 2011. Data mining: past, present and future. Knowl. Eng. Rev. 26 (1), 25–29.
Corbett, A., Anderson, J., 1995. Knowledge-tracing: modeling the acquisition of procedural knowledge. User Model. User-Adapted Interact. 4, 253–278.
Corbett, A.T., Koedinger, K.R., Anderson, J.R., 1997. Intelligent tutoring systems. In: Handbook of Human-Computer Interaction, pp. 849–874. North-Holland.
Crowder, N.A., 1959. Automatic tutoring by means of intrinsic programming. In: Automatic Teaching: The State of the Art. Wiley, New York, pp. 109–116.
de Jong, T., 2006. Computer simulations—technological advances in inquiry learning. Science 312, 532–533.
Dickler, R., Gobert, J., Sao Pedro, M., 2021. Using innovative methods to explore the potential of an alerting dashboard for science inquiry. J. Learn. Anal. 8 (2), 105–122. https://doi.org/10.18608/jla.2021.7153.
D'Mello, S., Graesser, A., 2009. Automatic detection of learner's affect from gross body language. Appl. Artif. Intell. 23 (2), 123–150.
Edwards, S.J., 1968. Development of a Self-Instruction Program on the Basic Principles of Fitting Clothing. Doctoral Dissertation. Texas Tech University.
Ericsson, K.A., Simon, H., 1980. Verbal reports as data. Psychol. Rev. 87, 215–251.
Feng, M., Beck, J., Heffernan, N., 2009. Using learning decomposition and bootstrapping with randomization to compare the impact of difference educational interventions on learning. In: Barnes, T., Desmarais, M., Romero, C., Ventura, S. (Eds.), Proceedings of the 2nd International Conference on Educational Data Mining, EDM 2009, pp. 51–60. Cordoba, Spain.
Gertner, A.S., VanLehn, K., 2000. Andes: a coached problem solving environment for physics. In: International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, pp. 133–142.
Gobert, J., Buckley, B., 2000. Special issue editorial: introduction to model-based teaching and learning. Int. J. Sci. Educ. 22 (9), 891–894.

Gobert, J., Clement, J.J., 1999. Effects of student-generated diagrams versus student-generated summaries on conceptual understanding of causal and dynamic knowledge in plate tectonics. J. Res. Sci. Teach. 36 (1), 39–53.

Gobert, J., Sao Pedro, M., 2016. A Real Time Alerting Tool to Transform Teachers' Assessment of Science Inquiry Practices, (NSF-IIS-1629045). Awarded by the. National Science Foundation.

Gobert, J.D., Sao Pedro, M.A., 2017. Digital assessment environments for scientific inquiry practices. In: Rupp, A.A., Leighton, J.P. (Eds.), The Wiley Handbook of Cognition and Assessment: Frameworks, Methodologies, and Applications, pp. 508–534. West Sussex, UK.

Gobert, J., Sao Pedro, M., Baker, R., Toto, E., Montalvo, O., 2012. Leveraging educational data mining for real time performance assessment of scientific inquiry skills within microworlds. J. Educ. Data Min. 4 (1), 111–143.

Gobert, J.D., Sao Pedro, M., Raziuddin, J., Baker, R.S., 2013. From log files to assessment metrics: measuring students' science inquiry skills using educational data mining. J. Learn. Sci. 22 (4), 521–563.

Gobert, J.D., Baker, R.S., Wixon, M.B., 2015. Operationalizing and detecting disengagement within online science microworlds. Educ. Psychol. 50 (1), 43–57.

Gobert, J.D., Baker, R.S., Sao Pedro, M.A., 2016. Inquiry Skills Tutoring System. US Patent no. 9,373,082 (issued).

Gobert, J., Sao Pedro, M., Betts, C., Baker, R.S., 2017. Inquiry Skills Tutoring System (Child Patent for Alerting System). US Patent no. 9,564,057 (issued).

Gobert, J.D., Moussavi, R., Li, H., Sao Pedro, M., Dickler, R., 2018. Real-time scaffolding of students' online data interpretation during inquiry with Inq-ITS using educational data mining. In: Cyber-physical Laboratories in Engineering and Science Education. Springer, Cham, pp. 191–217.

Gobert, J., Sao Pedro, M., Betts, C., Baker, R.S., 2019. Inquiry Skills Tutoring System (Child Patent for Additional Claims to Inq-ITS). US Patent no. 10,186,168 (issued).

Gobert, J., 2019. Inq-Blotter: Designing Supports for Teachers' Real Time Instruction (NSF-IIS- 1902647). Awarded by the. National Science Foundation.

Gobert, J., 2022. Engagement Alone Does Not Lead to Deep Science Learning. Inq-ITS. https://www.inqits.com/post/engagement-alone-does-not-lead-to-deep-science-learning.

Henderson, N., Min, W., Rowe, J., Lester, J., 2021. Enhancing multimodal affect recognition with multi-task affective dynamics modeling. In: 2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII). IEEE, pp. 1–8.

Hmelo-Silver, C.E., 2004. Problem-based learning: what and how do students learn? Educ. Psychol. Rev. 16, 235–266.

Johnson, W.L., Soloway, E., 1985. PROUST: knowledge-based program understanding. IEEE Trans. Software Eng. (3), 267–275.

Johnson-Laird, P., 1983. Mental Models: Toward a Cognitive Science of Language, Inference and Consciousness. Harvard University Press.

Kirschner, P.A., Sweller, J., Clark, R.E., 2006. Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educ. Psychol. 41 (2), 75–86.

Klahr, D., Nigam, M., 2004. The equivalence of learning paths in early science instruction effects of direct instruction and discovery learning. Psychol. Sci. 15 (10), 661–667.

Koedinger, K., MacLaren, B., 2002. Developing a Pedagogical Domain Theory of Early Algebra Problem Solving. CMU-HCII Tech Report, Pittsburgh, PA, pp. 02–100.

Kolodner, J.L., 1983. Reconstructive memory: a computer model. Cognit. Sci. 7 (4), 281–328.

Kolodner, J.L., 1992. An introduction to case-based reasoning. Artif. Intell. Rev. 6 (1), 3–34.

Kuhn, D., 2005. Education for Thinking. Harvard University Press, Cambridge, MA.

Kulik, J.A., Fletcher, J.D., 2016. Effectiveness of intelligent tutoring systems: a meta-analytic review. Rev. Educ. Res. 86 (1), 42–78.

Laney, D., 2001. 3D data management: controlling data volume, velocity and variety. META Group Res. Note 6, 70–73.

Lester, J.C., Boyer, K.E., Wiebe, E.N., Mott, B., Smith, A., 2019. Board 144: PRIME: engaging STEM undergraduates in computer science with intelligent tutoring systems. In: 2019 ASEE Annual Conference & Exposition.

Li, H., Gobert, J., Dickler, R., Moussavi, R., 2018. The impact of multiple real-time scaffolding experiences on science inquiry practices. In: Lecture Notes in Computer Science. Springer, pp. 99–109.

Li, H., Gobert, J., Dickler, R., 2019a. Scaffolding during science inquiry. In: Mitchell, J.C., Porayska-Pomsta, K., Joyner, D. (Eds.), Proceedings of the Sixth Annual ACM Conference on Learning at Scale. ACM, New York, NY, pp. 1–10.

Li, H., Gobert, J., Dickler, R., 2019b. Testing the robustness of inquiry practices once scaffolding is removed. In: Coy, A., Hayashi, Y., Chang, M. (Eds.), Proceedings of the Fifteenth International Conference on Intelligent Tutoring Systems. Springer, Cham, Switzerland, pp. 204–213.

Linn, M.C., Hsi, S., 2000. Computers, Teachers, Peers: Science Learning Partners. Erlbaum Associates, Mahwah, NJ.

Luan, H., Peter, G., Lai, H., Gobert, J., Yang Stephen, J.H., Ogata, H., Baltes, J., Rodrigo, G., Li, P., Chin-Chung, T., 2020. Challenges and future directions of big data and artificial intelligence in education. Front. Psychol. 11, 1–11.

Ma, W., Adesope, O.O., Nesbit, J.C., Liu, Q., 2014. Intelligent tutoring systems and learning outcomes: a meta-analysis. J. Educ. Psychol. 106 (4), 901.

McKendree, J., 1990. Effective feedback content for tutoring complex skills. Hum. Comput. Interact. 5, 381–413.

Merrill, D.C., Reiser, B.J., Ranney, M., Trafton, J.G., 1992. Effective tutoring techniques: a comparison of human tutors and intelligent tutoring systems. J. Learn. Sci. 2 (3), 277–305.

Mislevy, R., Chudowsky, N., Draney, K., Fried, R., Gaffney, T., Haertel, G., 2003. Design Patterns for Assessing Science Inquiry. SRI International, Menlo Park, CA.

Mislevy, R.J., Riconscente, M.M., Rutstein, D.W., 2009. Design Patterns for Assessing Model-Based Reasoning (Large-Scale Assessment Technical Report 6). SRI International, Menlo Park, CA. Downloaded May 31, 2011, from. http://ecd.sri.com/downloads/ECD_TR6_Model-Based_Reasoning.pdf.

Mislevy, R., Yan, D., Gobert, J., Sao Pedro, M., 2020. Automated scoring with intelligent tutoring systems. In: Yan, D., Rupp, A., Foltz, P. (Eds.), Handbook of Automated Scoring: Theory Into Practice. Chapman and Hall, London, UK.

Mott, M., Mott, B., Rowe, J., Ozer, E., Giovanelli, A., Berna, M., et al., 2021. "What's important to you, max?": the influence of goals on engagement in an interactive narrative for adolescent health behavior change. In: International Conference on Interactive Digital Storytelling. Springer, Cham, pp. 379–392.

Moussavi, R., Gobert, J., Sao Pedro, M., 2016. The effect of scaffolding on the immediate transfer of students' data interpretation skills within science topics. In: Proceedings of the 12th International Conference of the Learning Sciences. Singapore, pp. 1002–1006.

Moussavi, R., 2018. Design, Development, and Evaluation of Scaffolds for Data Interpretation Practices During Inquiry. Doctoral Dissertation. Worcester Polytechnic Institute, Worcester, MA.

National Research Council, 1996. National Science Education Standards. National Academies Press.

NGSS Lead States, 2013. Next Generation Science Standards: For States, by States. The National Academies Press, D.C.: National Academies Press, Washington, DC.

Nwana, H.S., 1990. Intelligent tutoring systems: an overview. Artif. Intell. Rev. 4 (4), 251–277.

Olsen, J.K., Belenky, D.M., Aleven, V., Rummel, N., 2014. Using an intelligent tutoring system to support collaborative as well as individual learning. In: International Conference on Intelligent Tutoring Systems. Springer, Cham, pp. 134–143.

Papert, S., 1980. Computer-based microworlds as incubators for powerful ideas. In: Taylor, R. (Ed.), The Computer in the School: Tutor, Tool, Tutee. Teacher's College Press, New York, NY, pp. 203–201.

Pellegrino, J., Chudowsky, N., Glaser, R., 2001. Knowing What Students Know: The Science and Design of Educational Assessment. National Academy Press, Washington, DC.

Perez-Valle, A., Sagasti, D., 2012. A novel approach for tourism and education through virtual Vitoria-Gasteiz in the 16th century. In: 2012 18th International Conference on Virtual Systems and Multimedia. IEEE, pp. 615–618.

Picard, R.W., 1997. Affective Computing. MIT Press, Boston, MA.

Quellmalz, E.S., Timins, M.J., Silberglitt, M.,D., Buckley, B.C., 2012. Science assessments for all: integrating science simulations into balanced state science assessment systems. J. Res. Sci. Teach. 49, 363–393.

Rai, D., Beck, J.E., 2012. Math learning environment with game-like elements: an incremental approach for enhancing student engagement and learning effectiveness. In: International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, pp. 90–100.

Rickel, J.W., 1989. Intelligent computer-aided instruction: a survey organized around system components. IEEE Trans. Syst. Man Cybern. 19 (1), 40–57.

Sao Pedro, M., Gobert, J., Heffernan, N., Beck, J., 2009. Comparing pedagogical approaches for teaching the control of variables strategy. In: Taatgen, N.A., vanRijn, H. (Eds.), Proceedings of the 31st Annual Meeting of the Cognitive Science Society. Cognitive Science Society, Austin, TX, pp. 1294–1299.

Sao Pedro, M.A., Gobert, J.D., Raziuddin, J., 2010. Comparing pedagogical approaches for the acquisition and long-term robustness of the control of variables strategy. In: Gomez, K., Lyons, L., Radinsky, J. (Eds.), Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences (ICLS 2010)—Volume 1, Full Papers. International Society of the Learning Sciences, Chicago, IL, pp. 1024–1031.

Sao Pedro, M.A., 2015. Inq-Blotter: Revolutionizing How Teachers Identify and Support Students Needing Help During Inquiry. Awarded by US DoE Institute of. Education Sciences (ED-IES-15-C-0018).

Sao Pedro, M.A., 2016. Inq-Blotter: Recognizing How Teachers Identify and Support Students Needing Help During Inquiry. Awarded by US DoE Institute of. Education Sciences (ED-IES-16-C-0014).

Sao Pedro, M., Baker, R., Gobert, J., 2013a. Incorporating scaffolding and tutor context into bayesian knowledge tracing to predict inquiry skill acquisition. In: Proceedings of the 6th International Conference on Educational Data Mining. Memphis, TN, pp. 185–192.

Sao Pedro, M., Baker, R., Gobert, J., 2013b. What different kinds of stratification can reveal about the generalizability of data-mined skill assessment models. In: Proceedings of the 3rd Conference on Learning Analytics and Knowledge, pp. 190–194. Leuven, Belgium.

Sao Pedro, M., Baker, R., Gobert, J., Montalvo, O., Nakama, A., 2013c. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. User Model. User-Adapted Interact. 23, 1–39.

Sao Pedro, M.A., 2013. Real-Time Assessment, Prediction, and Scaffolding of Middle School Students' Data Collection Skills Within Physical Science Simulations. In: Social Science and Policy Studies: Learning Sciences and Technologies Program Ph.D. Dissertation. Worcester Polytechnic Institute. Technical Report etd-042513-062949.

Schauble, L., 1990. Belief revision in children: the role of prior knowledge and strategies for generating evidence. J. Exp. Child Psychol. 49, 31–57.

Self, J.A., 1974. Student models in computer-aided instruction. Int. J. Man Mach. Stud. 6 (2), 261–276.

Shute, V.J., Psotka, J., 1994. Intelligent Tutoring Systems: Past, Present, and Future. Report No. AL-TP-94-0005. Air Force Materiel Command, Brooks Air Force Base, TX.

Shute, V., Glaser, R., Raghavan, K., 1988. Inference and Discovery in an Exploratory Laboratory. University of Pittsburgh Learning Research and Development Center.

Skinner, B.F., 1948. "Superstition" in the pigeon. J. Exp. Psychol. 38 (2), 168–172. https://doi.org/10.1037/h0055873.

Sleeman, D., 1985. UMFE: a user modelling front-end subsystem. Int. J. Man Mach. Stud. 23 (1), 71–88.

Soller, A., 2001. Supporting social interaction in an intelligent collaborative learning system. Int. J. Artif. Intell. Educ. 12 (1), 40–62.

Stansfield, J.L., Carr, B.P., Goldstein, I.P., 1976. Wumpus Advisor 1: A First Implementation Program That Tutors Logical and Probabilistic Reasoning Skills. AI Lab Memo 381. MIT, Cambridge, Massachusetts.

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M., 2005. The Andes physics tutoring system: lessons learned. Int. J. Artif. Intell. Educ. 15 (3), 147–204.

VanLehn, K., 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educ. Psychol. 46 (4), 197–221.

Walsh, M.B., Moss, C.M., Johnson, B.G., 2002. Quantitative impact of a cognitive modeling intelligent tutoring system on student performance in balancing chemical equations. Chem. Educ. 7, 379–383. https://doi.org/10.1007/s00897020620a.

Webb, G.I., Pazzani, M.J., Billsus, D., 2001. Machine learning for user modeling. User Model. User-Adapted Interact. 11 (1), 19–29.

White, B.Y., Frederiksen, J.R., 1990. Causal model progressions as a foundation for intelligent learning environments. Artif. Intell. 42 (1), 99–157.

White, B.Y., 1984. Designing computer games to help physics students understand Newton's laws of motion. Cognit. Instruct. 1 (1), 69–108.

Wood, H., Wood, D., 1999. Help seeking, learning and contingent tutoring. Comput. Educ. 33, 153–169.

Yang, C., Chiang, F.-K., Cheng, Q., Ji, J., 2021. Machine learning-based student modeling methodology for intelligent tutoring systems. J. Educ. Comput. Res. 59 (6), 1015–1035. https://doi.org/10.1177/0735633120986256.