

**Research Article****DEVELOPMENT OF AUTHORING TOOLS AND PLAYERS FOR STEP-BY-STEP INTELLIGENT TUTORING THAT REPRODUCES THE CLASSROOM TEACHING STYLES OF HUMAN TEACHER**

¹Gwang-Hyok Kim, ^{2,*}Song-Hwan Kwon, ³Gyong-Jin Kim, ⁴Gang-Bom So, ²Sin-HyeJang, ²Ha-Gyong Kim, ²Jong-Nam Rim, ²Chung-Song Ko, ²Song-Il Pak, ²Chol-Jun Hong and ⁵Myong-Il Kim

¹Institute of Satellite Information Technology, State Academy of Sciences, Pyongyang, Democratic People's Republic of Korea

²Faculty of Information Science, University of Science, Pyongyang, Democratic People's Republic of Korea

³Unjong Science Region Postgraduate School, Pyongyang, Democratic People's Republic of Korea

⁴Information Department, Sinuiju Honoured Ex-servicemen College of Economics, Sinuiju, Democratic People's Republic of Korea

⁵Institute of Information Technology, University of Science, Pyongyang, Democratic People's Republic of Korea

Received 08th July 2026; **Accepted** 15th August 2026; **Published online** 30th September 2026

Abstract

This paper proposes an authoring tool and a player development approach for step-by-step tutoring that reproduces human teacher's classroom lecture. The step-by-step tutoring approach is one of the most frequently used methods in many intelligent tutoring systems, allowing authoring tool users to create an intelligent tutoring system for their subject without significant burden. This paper proposes a step-by-step tutoring principle that reproduces human teacher's classroom lecture styles, and proposes an authoring tool, a playback structure, and a workflow diagram. We also propose a related data structure, database, student model, domain model, and instructional model construction method. The tester (or player) structure and the action flow diagram proposed in this paper are the same as in the related platform. In this paper, active learning, scaffolding and proactive interventions were applied to the classroom tutoring styles. We are evaluated the degree to which the proposed system was preferred by the authors (teachers or experts in the field). We also quantitatively evaluated student achievement when intelligent tutoring contents created using the proposed authoring tool was used across the platform. The evaluation results confirmed that the authors' preferences for the proposed authoring tools were positive and that the results of the study using the system improved the academic performance of the students.

Keywords: Intelligent Tutoring Systems, Step-by-Step Tutoring Model, Authoring Tools, Checkers, Players, Platforms.

INTRODUCTION

Among the different models used in intelligent teaching systems are the step-by-step tutoring models. The step-by-step tutoring model is known as the method of focusing to pedagogical skills among pedagogical skills and tutoring contents. Pedagogical modes include proactive intervention, active learning, and scaffolding. We aim to propose designs to build by step tutoring models of intelligent tutoring systems that reproduces classroom lessons as they are. To do this, the principle of step-by-step pedagogical mode should be redefined to suit its purpose, and the overall flow of the system and the system structure should be designed. We also need to define data structures for system construction, build databases, and propose algorithms that support proactive intervention, active learning, and scaffolding learning. In Section 3, we review the previous work of each subsystem by dividing the intelligent tutoring system into authoring tools, checkers (or players), and platforms. Here, we analyze in detail the GIFT, a generalized intelligent tutoring framework in particular. In Section 4, we analyze previous studies on step-by-step tutoring modes in the intelligent teaching system for tutoring. In Section 5, we propose an approach of authoring and checking for step-by-step tutoring from the viewpoint of design submission. The paper presents the evaluation of the proposed system construction approach by means of the degree to which the authoring tool is preferred by the tool users.

We also present results of a quantitative evaluation of learners' performance gains using the results created by the proposed authoring tool.

Intelligent Tutoring Systems (ITSs)

The paper will address, review and summarize the following research questions about intelligent tutoring systems.

RQ 1: What are the Intelligent Tutoring Systems?

Intelligent tutoring systems are computer-based learning environments that provide adaptive, personalized tutoring and immediate feedback, and also mimic one-to-one human tutors. Intelligent tutoring systems have a research history of about 50 years, starting from the study of the SCHOLAR system (Carbonell [18]) in the 1970s, and similar systems have been continuously studied [19-22]. In general, intelligent tutoring systems consist of four components: a domain model (expert model), a student model (user profile) that tracks learners' learning and cognition, a pedagogical model (tutoring model) that relates to pedagogical strategies and path guidance and a user interface [62-76]. In the following figure, we use the term module instead of model in the sense that we implement the model. The domain model is a model that includes knowledge, concepts, and rules about the current learning topic, and the student model is a model that tracks the learner's cognitive state, proficiency, and incorrect views by analyzing the interaction and progress between the student and the system.

***Corresponding Author:** Song-Hwan Kwon,

Faculty of Information Science, University of Science, Pyongyang, Democratic People's Republic of Korea.

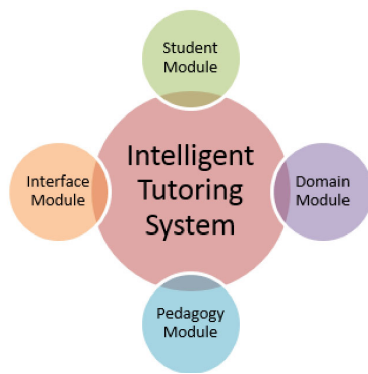


Figure 1. The Components of ITSs [63]

The pedagogy model determines the instructional strategy when to give a hint, give an explanation, inform a new one, and present challenging problems, and guides the learning path. The user interface provides a friendly environment for the student and teacher and allows them to interact [49].

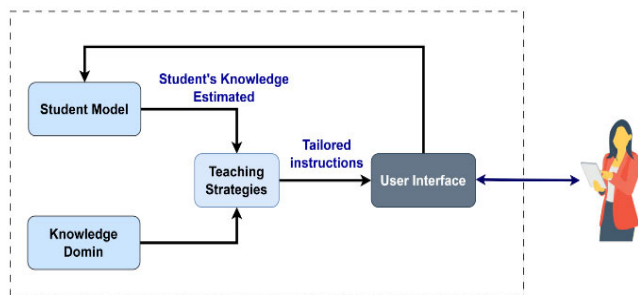


Figure 2. The General Flow Diagram of ITSs [50]

RQ 2: What are the types of intelligent tutoring systems and what are the typical examples of such types?

The first intelligent tutoring system in the 20th century was the Intelligent Tutoring Tools (ITTs) developed by the Byzantium research project, which included six universities. This intelligent tutoring system has received positive appraisal from many research institutions that use it as a general tutoring system builder [6]. Using this constructor, you can create an Intelligent Tutoring Applet (ITA). Using this builder, many teachers created ITAs on different topics [6, 23].

Conversational Chatbots/Dialogue Tutors

These systems use natural language processing to assist students in their learning through pedagogical conversation, as human tutors and guide students' learning through suggestion and motivational language, rather than providing immediate answers to students. The Sghaier [17] reports that the large language model is promoting new developments in higher education, especially in personalized and customized education.

AutorTutor: Through a Socratic style of conversation, it helps learners learn about a topic.

Khanmigo (by Khan Academy): It helps students to learn mathematics and science.

Mizou: It enables teachers to create tailored chatbots that support a specific curriculum.

Cognitive Tutors (Model-Tracing Tutors)

These systems model the cognitive processes involved in problem solving. These systems model the learner's step-by-step learning approach to a problem and implement immediate feedback in case of errors.

MATHia (by Carnegie Learning): Proceeding structured and personalized mathematics.

Andes Physics Tutor: used in education to help solve physical problems.

ESC101-ITS: This system was developed by IIT Kanpur to assist in introductory programming.

Adaptive Learning Platforms (Knowledge-Tracing Tutors)

These systems model the student's knowledge state[58]. This allows the learner to track the knowledge concept that he has learned perfectly successfully, dynamically fine-tune the complexity of learning, and adjust the learning path to suit the student.

DreamBox Learning: Assists in student mathematics learning in real time.

There are also ASSISTments, which provide immediate feedback in solving mathematical problems, and ActiveMath, which is an adaptive mathematics tutoring based on the Web.

Intelligent Game-based and Simulation Tutors

These systems use a simulated or game environment for interactive training. It is mainly used in training or program development with technical difficulties.

ST Math: Use visual quiz problems for mathematics instruction.

Flight Simulators: It is used to support training in flight simulation environments.

Multimodal/Specialized Tutors

These systems use audio, handwriting, and visual signals rather than keyboards for student performance [50].

Amira Learning: It can hear students' reading and provides real-time guidance.

ELSA Speak: We use the system with AI technology to improve the pronunciation of students.

Squirrel AI: This system uses student-centered dialogue data when performing a speech dialogue.

Collaborative Intelligent Tutoring Systems (CITS)

Collaborative Intelligent Tutoring Systems (CITS) have evolved from ITS. A CITS provides a collaborative learning environment, allowing students to interact not only with the computerized system but also with other students for mutual understanding and explanation.

RQ 3: What are the major intelligent tutoring systems and platforms?

The main intelligent tutoring systems and platforms are as follows.

The system Carnegie Learning is known to use cognitive science to conduct high-level mathematics instructing.

McGraw Hill's Assessment and Learning in Knowledge Spaces (ALEKS) is an online tutoring system and evaluation program that includes mathematics subject, chemistry, statistics, and commercial subject-related course resources [3]. ALEKS is a web-based platform that uses adaptive queries to determine the knowledge status of a student in particular.

Technological Pedagogical Content Knowledge (TPACK) is an AI-driven intelligent tutoring system that supports personalized mathematics learning based on the content knowledge framework. TPACK uses reinforcement learning and natural language processing (NLP) to provide context-sensitive feedback, interactive simulation environment [4].

DreamBox Learning is an adaptive math tutoring platform that fine-tunes the query in real time.

ST Math uses spatiotemporal and visual quizzes problems without resorting to text to teach mathematical concepts.

ASSISTments is a platform for helping teachers in a way that provides appropriate feedback to students' mathematics learning progress.

ASSISTments Dataset 2009-2010 includes student interaction logs of students who were exposed to the ASSISTments online teaching platform during 2009-2010 [5]. This dataset is the most widely used benchmarks dataset in Educational Data Mining and the Intelligent Tutoring Community, which provides a very rich foundation for student learning history modeling and performance prediction. Each row of this dataset contains detailed interactive attribute information such as student identifiers, problem numbers (IDs), specific skills (knowledge components) names, success rates (hits/misses), and response sequences over a period of time. The features of this dataset are structured to support complex modeling tasks such as Bayesian Knowledge Tracing (BKT), Deep Knowledge Tracing (DKT), and Dynamic Bayesian Networks (DBN). Researches uses these data to develop and verify adaptive learning algorithms, analyze the impact of multiple technique matching, and explore the knowledge perceived over time.

Squirrel AI is known as a system that focuses on advanced algorithms for adaptive learning, especially in large-scale environments.

Duolingo uses AI-driven language learning exercises that provide personalized instruction and instant feedback. The Duolingo Intelligent Tutoring System uses artificial intelligence and machine learning algorithms to model human tutors and adapt to students' speed and language levels. Duolingo as an intelligent tutoring system has the following functions: adaptive learning, modeling, and instant feedback. Adaptive learning functions analyze learners' learning performance in real time, such as fine-tuning the complexity of the lesson and discovering the vulnerability. In the modeling function, we create an individual model for each student based on each individual's history to identify the exercises needed for each learner. Duolingo supports student learning through AI-

driven dialogue and interaction, and provides immediate feedback when learning errors occur.

Cognitive Tutor Authoring Tools (CTAT) is known as a basic framework that uses example-tracing and cognitive modeling for tutoring[40]. This system works based on a model of how students acquire and apply knowledge. These systems use a step-by-step instructional principle and provide feedback control to the learner. Cognitive Tutor (CTAT) allows pedagogical researchers with limited programming experience to create (cognitive) tutors [16].

SmartTutor is a system developed by the University of Hong Kong University (HKU), which integrates Internet technology, pedagogical research and artificial intelligence to support student learning.

AutoTutor simulates the teaching style and teaching strategy of human teachers to assist college students in learning computer hardware, operating systems, and Internet subjects. AutoTutor understands the learner's input through the keyboard and then performs interactive actions such as feedback, guidance, feedforward, and hints.

In particular, the intelligent authoring tools include ASPIRE, the Cognitive Tutor Authoring Tools (CTAT), GIFT, ASSISTments Builder, and AutoTutor tools [14].

There are also intelligent tutoring systems such as Algebra Tutor PAT, SQL-Tutor, EER-Tutor, COLLECT-UML, StoichTutor, Mathematics Tutor, eTeacher, ZOSMAT, REALP, CIRCSIM-Tutor, Why2-Atals, ActiveMath, ESC101-ITS, and ErAdaptrEx [15].

ACTIVEMATH is a generic web-based learning system that dynamically generates interactive (mathematical) courses adapted to the student's goals, preferences, capabilities, and knowledge [26].

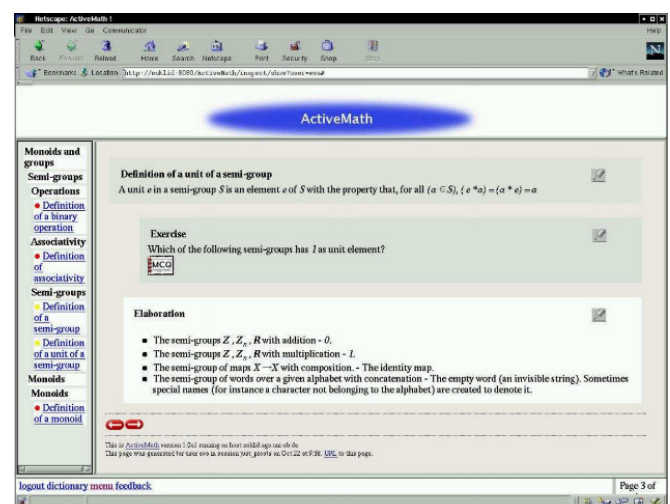


Figure 3.A Screenshot of an ActiveMath Session [27]

RQ 4: What are the main characteristics of typical intelligent tutoring systems?

The main characteristics of typical intelligent tutoring systems are as follows.

As personalized learning, each student applies different teaching strategies, examples, and practical exercises. There is

student modeling, which has detailed history of student expertise, gaps in knowledge, and misunderstandings [34]. There is an instant feedback function, which provides real-time feedback, an error description, scaffolding for problem-solving and resource presentation. Active learning is not motivated by information obtained passively but by motivating students to solve problems in a more active and active stance. Intelligent tutoring systems have an expert model of the subject to guide and assist students' learning efficiently, which is called domain knowledge.

RQ 5: What do MOOCs look particularly like?

MOOCs (Massive Open Online Courses) are free, low-cost, web-based, and educational-related curriculum datasets, with a very large number of participants, sometimes involving many universities, companies, and organizations, to create curricula. The characteristics of MOOCs include lecture videos, interactive quiz games, and conference results.

RQ 6: What is the general architecture of the intelligent tutoring system?

The general architecture of intelligent tutoring system is as follows. By a consensus of many researchers (Nwana, Freedman, Nkambou et al.) the basic components of the intelligent tutoring system were identified as follows [8-10]: Domain models, student models, pedagogical models, and user interface.

The domain model consists of expert knowledge, facts, rules, and problem-solving strategies for specific topics. The domain model is divided into two domains. one is the content domain and the other is the academic performance evaluation domain. The content domain includes the curriculum, subject knowledge and structured knowledge. The academic performance assessment domain should have models that are ready to solve the problems that the system faces, to detect and diagnose errors in the student's learning process in time, and to evaluate the learner's performance. The domain model plays the role of "what" in the intelligent tutoring system. A domain model, also known as cognitive model or expert knowledge model, can be constructed using either of the two theory of learning. One learning theory is ACT-R, which considers all possible steps required to solve the problem. The model built on the basis of this theory includes concepts, rules, and problem-solving strategies of the domain that we want to learn [10]. This model is a source of expert knowledge, acts as a standard for evaluating student achievement and performs error detection[10]. Another theory is Stellan Ohlsson's theory, known as constraint-based modeling (CBM), in which the domain model is represented as a set of constraints for exact solutions [11, 12].

The student model is used to assess how much improved the cognitive status and academic achievement by the student's learning process. The student model includes diagnostic information and learning progress tracking. Diagnostic information includes time spent on task execution, types of errors, patterns associated with assistance, and knowledge that judges that a concept is completely mastered. In learning progress tracking, the system can be used to update in real time information about student achievement, inferring the advantages and disadvantages of student learning. The student model plays the role of "who" in the intelligent tutoring system. In intelligent tutoring systems, the process of student

modeling is called model tracking. In constraint-based intelligent tutoring systems, negative feedback (error-based feedback) or positive feedback is performed based on the results of the student model.

The main components of the tutoring activity include the introductory, the representation of the tutoring content, the presentation of practical problems, and the application. The pedagogical model receives the results from the domain model and the student model and decides which instructional strategy or action to take. The pedagogical model plays a "how" role in the intelligent tutoring system. The pedagogical model includes instructional strategies, learning path selection functions, and adaptability. The instructional strategies include suggestions, feedback, explanations, and worked examples. Learning path selection determines a sequence of topics and learning contents that meet the student's needs and the learning objectives. Adaptive ability selects the optimal instructional strategy from a library that allows the student to succeed. In the problem-solving process-based intelligent tutoring systems, learners hope to guide what to learn next in their current learning situation. Also, intelligent tutoring systems provide feedback to enable the learner to reach the learning goal at a reasonable time if he/she makes mistakes [13]. Using knowledge tracking techniques, intelligent tutoring systems track the student's learning process and preserve its history. For example, the cognitive tutoring system developed by Carnegie Mellon University represents knowledge-tracking information called skillometer, which represents the degree of learner's cognitive success for each expertise involved in solving algebraic problems by visual graphs. When a student requests assistance such as a hint or a mistake occurs, the knowledge tracking data and skillometer are updated automatically and in real time [5-7].

User interface models enable interaction and communication with students and intelligent tutoring systems. An interactive and a visual environment are necessary. In interactive environments, there are presentation of training data, query presentation, and receipt of learner input.

Tommaso [24] proposed an easy and fast way to develop an interface for intelligent tutoring authoring tools using Large Language Models and prompt engines. This tool allows you to develop an authoring tool without programming.

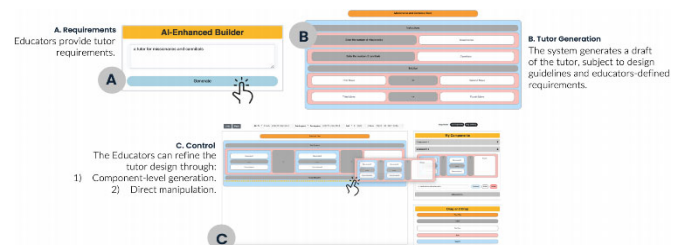


Figure 4. Illustration of Our AI-Enhanced Tutor Builder System: Educators Provide Input Requirements (A) which Inform the Automated Generation of a Draft Tutor Interface (B), Followed by the Educator's Hands-on Refinement through Component Generation and Direct Manipulation (C), Visualizing the Integration of Generative Design and Educator-Driven Customization[24]

In the paper from Priynka [25], a method for designing intelligent tutoring systems that helps students in online programming is proposed. In the following, the proposed methods are presented through diagrams and algorithms.

RQ 7: What are the methods of designing and developing intelligent tutoring systems?

Corbett [7] summarized that the design and development of an intelligent tutoring system is done by repeatedly applying the following four steps: Needs assessment, cognitive task analysis, development of the initial version of tutoring systems, and evaluation.

The first step, the needs assessment, is the instructional design process related to software development. This step includes learner analysis and consultation with experts in the field. This step involves designing the student domain and the knowledge domain. This design step should consider the probability that a student will solve a problem, the time to reach the desired benchmark, and the ability to address the probability that a student will apply that expertise in practice in the future. Next, the interface should be well designed to enhance the effect of interaction [13].

The next step, the cognitive task analysis, is to conduct consultation with domain experts and to analyze and design how to observe the tutoring and learning activities.

Next, there are development step for a first version of system and evaluation step for the constructed system.

RQ 8: What are the types of UI design of intelligent tutoring systems?

The UI design styles of intelligent tutoring systems are quite diverse. These interfaces are designed to suit the instructional strategy used, such as from constraint-based rules to dialogue interaction strategies. The main design styles are as follows. There is dialogue interface, adaptive and dynamic interface, gamified interface, model-tracking (problem-solving) interface, multi-modal interface, and data-driven interface.

Conversational/dialogue-based interface

This approach uses natural language processing based on a powerful Large Language Model to perform human teacher-like activities in a dialogic manner. Important features include Chatbot interface, tailored response, and context understanding.

Adaptive/dynamic interface

In this form, content representation and their placement vary according to the student's behavior, learning speed and cognitive status. The main features are personalized navigation, adaptation of content to the level of expertise, and real-time customization of the interface environment. This type of design aims to minimize the burden and unnecessary consumption of learning by displaying the necessary information regarding the tendency of the individual learner's learning history.

Gamified interface

This format integrates game-style design elements to enable more active learning activities with a higher interest in learning. The main features are points, leaderboards, achievements, and visual effects (e.g., graphics or maps representation of learning progress) such as games.

Model-tracing/problem-solving interface

These interfaces allow for design, considering all the critical instructional strategies, the learner's mental learning activities against the expert model. The main features are step-by-step guidance, instant feedback, and structured problem-solving domain. An example is Cognitive Tutor. It is now considered appropriate to apply such a STEM (mathematics, physics, computer science) education.

Multimodal/immersive interface

This format is known as a very advanced technology that goes beyond the effects of traditional graphical user interfaces (GUIs) by including such interactions with 3D environments or audio. The main features are virtual reality (VR), augmented reality (AR), and voice user interfaces (VUI). It is suitable for medical practice, simulation-based combat training, hand-held technical upgrading, etc. [50].

Minimalist and data-driven interfaces

This format uses data visualization to show the student's performance. The main features are grid-based structure, hierarchical visual data, etc.

RQ 9: What relationship are the knowledge graphs and the intelligent tutoring system?

The knowledge graphs and intelligent tutoring systems are integrated to develop intelligent, more individualized and context-oriented educational platform[1, 2]. While the intelligent tutoring system performs individualized learning guidance, diagnoses learning situations and errors, and makes appropriate teaching, the knowledge graph plays the role of a structured knowledge base that provides the necessary context, semantic relationships, and educational structure to the AI. A knowledge graph acts as a domain model in intelligent tutoring system [29]. Researchers with the same view as Alan Turing have the view that knowledge graphs play a bridge between human reasoning and machine processing. The relationship between these two in tutoring is discussed in detail as follows.

a. Structuring the learning domain

The knowledge graphs organize educational-related data (concepts, expertise, topics) into a network of nodes (entities) and edges (relationships). Such relationships can be either "prerequisite" or "part of" or so on.

b. Providing personalized learning paths

The knowledge graphs allow an ITS to match the learning progress of individual learners to the knowledge graph and perform tailored learning. As such an example, there is the identification of learning errors (or weakness identification). That is, when an ITS identifies that a student has a very difficult learning of a concept (e.g., informed search), it can perform a backward search to find the essential prior concept (e.g., uninformed search) of that concept through a knowledge graph. Another example of tailored learning is adaptive recommendation. Such an ITS can recommend exercises that include a set of knowledge and high-density information designed to maximize the knowledge gap of a particular student [28].

c. Increasing the interpretability and precision of AI-driven systems

In modern intelligent tutoring systems, the knowledge graph is integrated with Large Language Models, which can help overcome errors or illogical explanations of LLMs. There are a precise retrieval and Explainable tutoring [31].

Precise retrieval: When a student asked a question, the intelligent tutoring system uses the knowledge graph to retrieve the precise and domain-specific information. This approach is called knowledge graph-enhanced retrieval-augmented generation (KG-RAG) [30].

Explainable tutoring: An intelligent tutoring system using a knowledge graph can explain why this particular topic was recommended, not acting like a black box. Intelligent tutoring systems guide the learning path that is suitable for the student based on the knowledge graph.

PREVIOUS RESEARCH ON ITSs AUTHORING TOOLS, CHECKERS (OR PLAYERS), AND PLATFORMS DEVELOPMENT

Generally, ITS authoring tools are needed for the creation of contents of intelligent tutoring, players for delivery, checkers for evaluation, and platforms allow the intelligent tutoring systems to experience in an interactive way in the cloud environment. This paper does not address previous work on platforms because platforms and players are generally in the same mode of operation. Authoring tools must include checkers or players. Thus, in Section 1, which is concerned with the authoring tool, we summarize and analyze the typical authoring tools and checkers (including players).

Authoring Tools

These tools aim to enable any professional or teacher to develop an intelligent tutoring system for their subjects without developing programs and without knowledge of AI. The following are the tools for authoring ITSs.

CTAT (Cognitive Tutor Authoring Tools) is one of the most popular tools, which supports the so-called “example-tracing” and “model-tracing” [36]. SimStudent is a machine learning agent developed to help beginner creators build intelligent tutoring systems (ITS) without the need for program development [38].

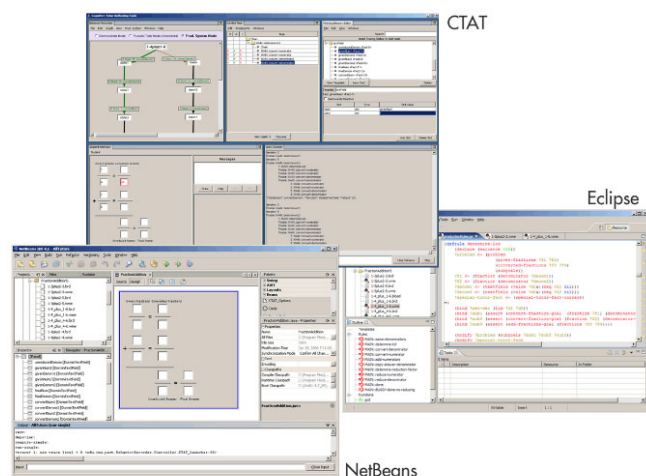


Figure 11. The Cognitive Tutor Authoring Tools[40].

xPST (Extensible Problem Specific tutor) is an authoring tool used to develop an ITS for geometry and statistics [32]. The GIFT (Generalized Intelligent Framework for Tutoring) is a framework developed to enable rapid and standard creation of adaptive training [37]. XAIDA & REDEEM is tools that enable the development of intelligent tutoring systems such as manufacturing and classroom instruction [32].

ACTS (AI-Based Classroom Teaching Simulator) is Conversational ITSs developed by the University of Virginia that simulates the classroom teaching of human teachers.

DomainBuilder is the name of the authoring tool included in the TutorBuilder system, which allows for test and evaluation of creations associated with medical education [39]. The functions are DomainBuilder and TutorBuilder. DomainBuilder has visualized test functions, and TutorBuilder has a playback function. By comparing their solution and a constraint checklist set, it creates a tutor based on constraints that automatically evaluate students. This software verifies the domain model to ensure that the constraints are defining the right solutions [41].

Checkers (or Players)

The checkers (or players) check and evaluate whether the production artifacts actually improve student learning and whether there is no error in student interaction [33]. These tools have the following functions. It functions as a storage and interpretation of interaction information with the student. For example, the Tutorshop tool keeps the information by tracking the actions that the student interacts with the system, and researchers use this information to test instructional models or try to improve their performance [36]. There is also a testing function for natural language. For example, ConceptGrid checks the correctness of the question response by the natural language that the student is given and received, comparing with the results of human student interaction [35]. Test methods of checkers are as follows [39]. A simulation-based approach using a tool such as SimStudent is presented, which creates a synthetic student to test the expert model. A tool like TutorBuilder can play or preview the results created by direct interaction. DomainBuilder checks the consistency and integrity of the domain content created using the validity engine [42-45].

GIFT (Generalized Intelligent Framework for Tutoring)

Our research and development focused and investigated GIFT specifically for the design and development of intelligent tutoring systems. The results are as follows.

The Generalized Intelligent Framework for Tutoring (GIFT) is an open source AI-driven platform developed by the United States DEVCOM Soldier Center to create, manage, and evaluate an adaptive and personalized tutoring system. Also, GIFT is a service-oriented modular framework based on the experience of tools, methods and standards that facilitate the development of adaptive tutoring systems (AIS) [47, 59].

The goal of GIFT is to develop an authoring tool that can develop new ITS, ITS components (sensor model, instructional model, user interface, sensor interface), tools and methods, and creation standards that can be used or reused in an external training environment. It also provides instructional managers

that include the best teaching principles, strategies and tactics for ITS. It also provides an experimental tool to analyze the effects of components, tools and methods of ITS [48]. The GIFT has three functions: Authoring, Instructional Management, and Experimentation and Evaluation [48]. Authoring provides a convenient tool (learner model, teaching model, user interface, sensor interface) for development of ITSs in the field by subject or topic experts who is not aware of information technology, computer science, programming technology, artificial intelligence principles and techniques. Thus, authoring tools play the role of instructional designers through graphical user interfaces, such as routing, evaluation, establishment of adaptation rules, etc., including CAT (Course Authoring Tool), DAT (Domain Knowledge Tool), and SAS (Survey Authoring System) [51, 59].

The instructional management function provides an instructional engine (eMAP) to adaptively manage the instructional flow based on real-time data for the learner, and GIFT uses real-time data such as learner learning performance, biometric features, and survey response to adaptively use the learner's cognitive state in the instructor [52]. Experimentation and Evaluation Function: ITSs Researchers and developers work as test tools to evaluate the effectiveness of various tutoring methods. We use this function to evaluate new instructional strategies, test cognitive models, and compare learning outcomes through the analysis of a set of learners [53].

The four components of the above-mentioned intelligent tutoring system are as follows from the viewpoint of the configuration module in GIFT [54]. Learner Module tracks and evaluates current learning states, such as student ability, temperament, and learning performance [55]. Pedagogical Module determines the best instructional strategy (e.g., feedback or hint) based on the learner state [56]. Domain Module converts instructional strategies within a given subject into specific learning actions [57]. Sensor Module integrates external data such as physiological sensors, emotions and pose recognizers to complete the learner model [60].

GIFT has completed, published and distributed its design documents annually through international academic seminars, which are as follows [61-68].

Learner Modeling documentation was completed in 2012 and the document was released in 2013 as Volume 1: ISBN 978-0-9893923-0-3.

The instructional management was completed in 2013 and the document was published in 2014 as Volume 2: ISBN 978-0-9893923-2-7.

The authoring tools documentation was completed in 2014 and the document was distributed in 2015 as Volume 3: ISBN 978-0-9893923-6-5.

The authoring tools documentation was completed in 2014 and the document was distributed in 2015 as Volume 3: ISBN 978-0-9893923-6-5.

The Assessment Methods document was completed in 2016 and was published in 2017 as Volume 5: ISBN 978-0-9977257-2-8.

The team tutoring document was completed in 2017 and the document was distributed as Volume 6: ISBN 978-0-9977257-4-2 in 2018.

The self-improvement systems (self-improvement systems) document of ITS was completed in 2018 and the document was released in 2019 as Volume 7, International Standard Book Number: 978-0-9977257-7-3.

The operation flow diagram of each module of GIFT is as follows [62].

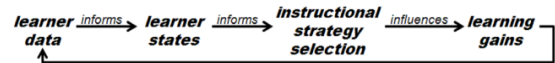


Figure 12. The Overall Flow Diagram of GIFT Platform [63, 74]

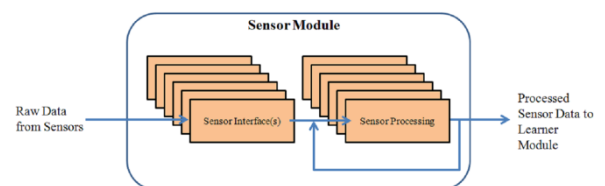


Figure13. The Flow Diagram of Sensor Modules in GIFT Platform [64].

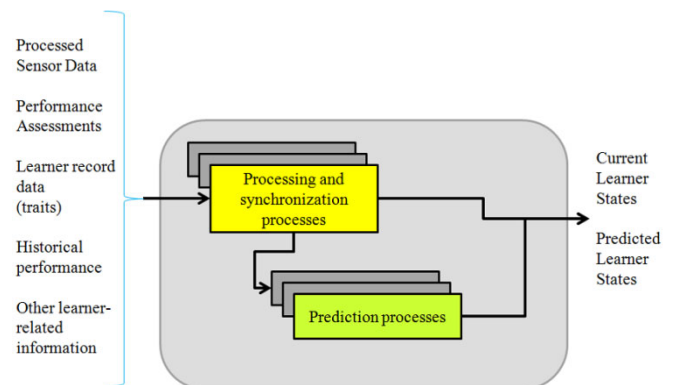


Figure14. The Flow Diagram of Learner Modules in GIFT Platform[67].

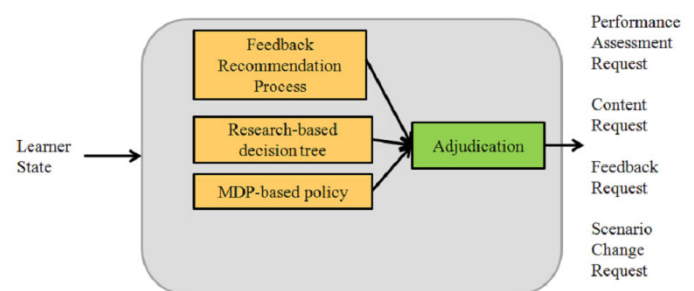


Figure15. The Flow Diagram of Instructional Modules in GIFT Platform [66]

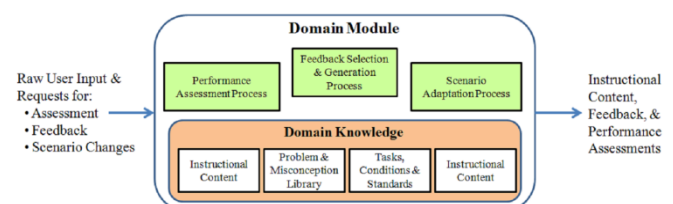


Figure 16. The Flow Diagram of Domain Modules in GIFT Platform [65]

Approaches to GIFT and the development and authoring modes are as follows [66-68, 75, 76].

GIFT cloud: A way to access a cloud that is already built online on the Internet without GIFT on its own computer, and to enable instructional design or course design using the latest updated version.

GIFT Local: This is the way you download an offline version and develop it, providing local service management in Windows, etc. operating systems.

Integration: A collaborative approach to traditional media such as external simulators, games, and PowerPoint.

Domain knowledge file (DKF): In the GIFT design, authors of courses decide which instructional strategies should be implemented based on transitions of learner states, which are authored in the Domain Knowledge File (DKF).

Gateway module: The gateway module provides a connection between the external training application (such as VBS2, PowerPoint, TC3 Sim/Vmedic) and the GIFT in add-in (plug-in) mode.

For example, GIFT 4.0 provides a set of authoring tools such as the author's curriculum (CAT, Course Authoring Tool), the author's adaptive feedback and instructional strategies (DAT, Domain Knowledge File Authoring Tool), and the author's survey (SAS, Survey Authoring System).

The creation of the curriculum of GIFT is as follows [68].

.xml Editor: All the authoring tools of GIFT use the.xml editor or visualized interface.

CAT (Course Authoring Tool): The author uses CAT to design the required course flow.

Node in CAT: Added and extended within CAT to add a transition between concepts, surveys, guidance provided to external training applications and learners.

CAT's link function: The author links training applications in the course to the specific DKF files they authored.

Export Tool: After the course is completed, it is possible to play on a computer without the installation of the full GIFT version.

The following figure illustrates the authoring process in GIFT [68].

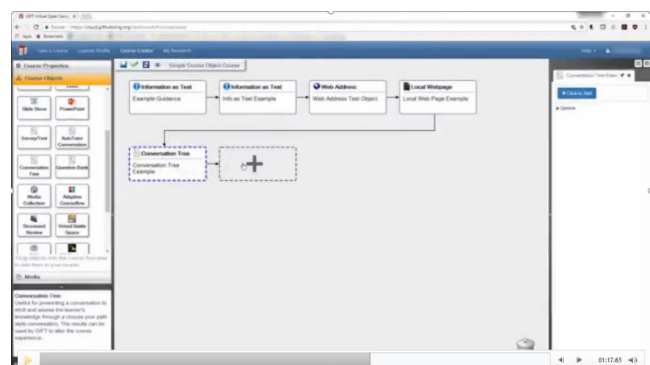


Figure17. An Example of an Instructional Model Authoring Flow Diagram in the GIFT Framework (1)[68]

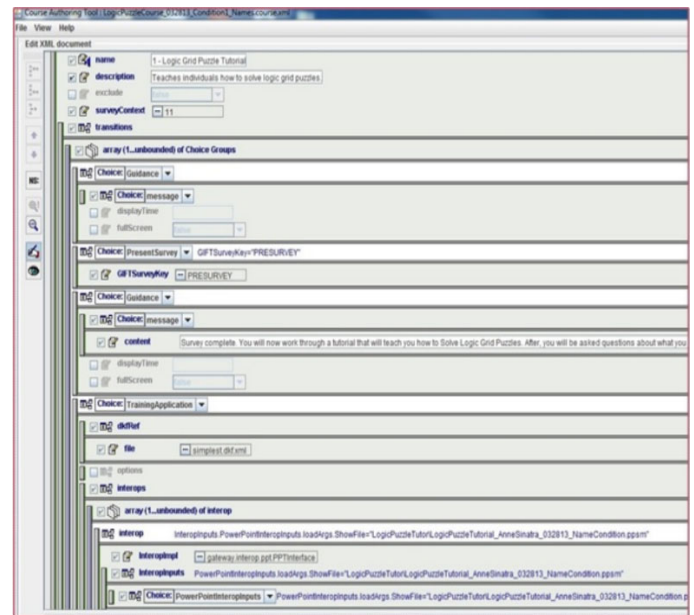


Figure18. An Example of an Instructional Model Authoring Flow Diagram in the GIFT Framework (2)[68]

The architecture and operation flowchart of GIFT is as follows [62].

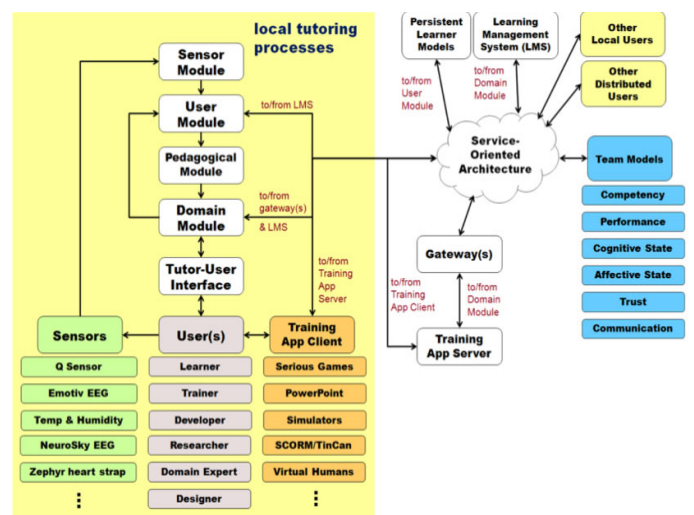


Figure19. The Architecture and Operation Flow Diagram of GIFT Platform [65]

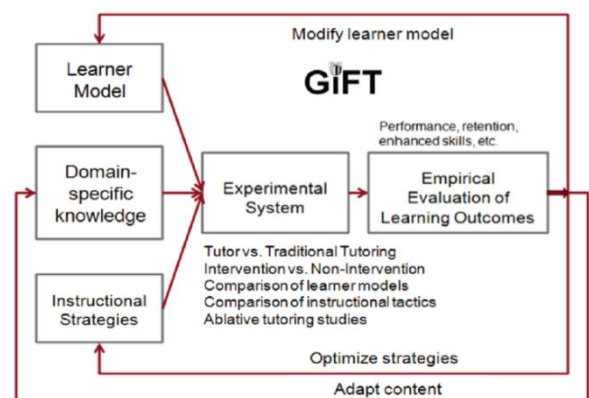


Figure20. The Architecture and Operation Flow Diagram of GIFT Platform (Overview)[65]

In GIFT, the interaction between learner, tutor and environment takes place as follows [67].

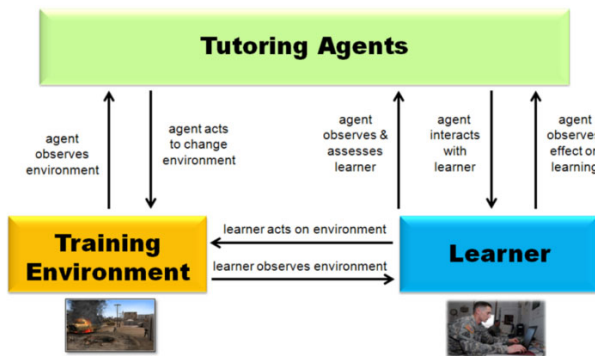


Figure 21. Interaction between Learner, Tutor and Environment (from an Agent Perspective) on the GIFT Platform [66]

In GIFT, the interaction between the elements of the system, the individual learner and the external environment is done as follows.

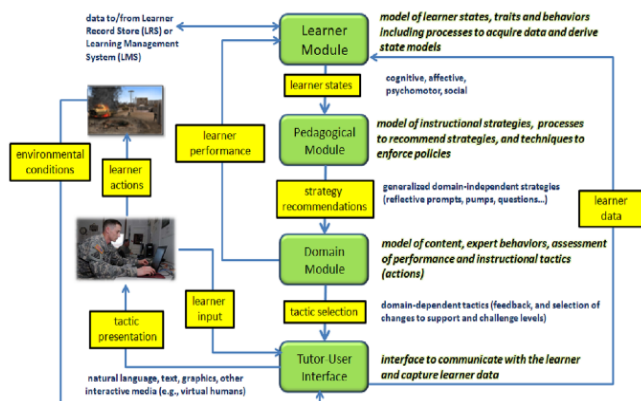


Figure 22. Interaction between System Components, Individual Learner and External environment on GIFT Platform[66].

In GIFT, the Learning Effect Model (LEM) for individual learners is as follows [67].

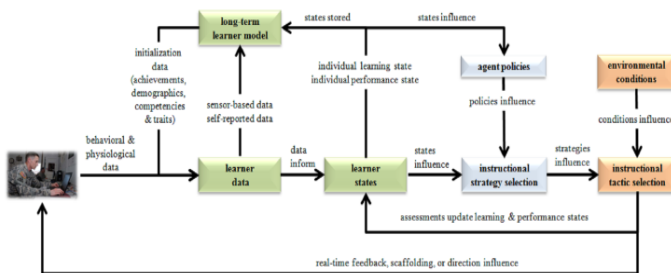


Figure 23. Learning Effect Model (LEM) for Individual Learners on the GIFT Platform[67]

The Learning Effectiveness Model (LEM) for Group Learning in GIFT is as follows [68].

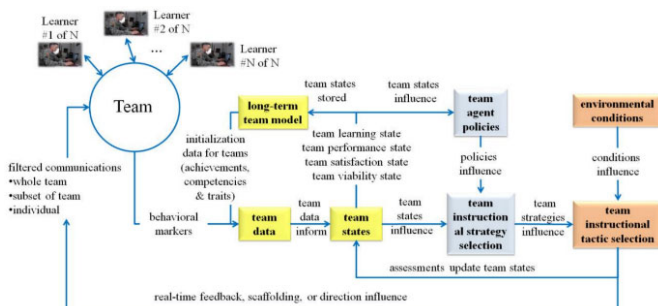
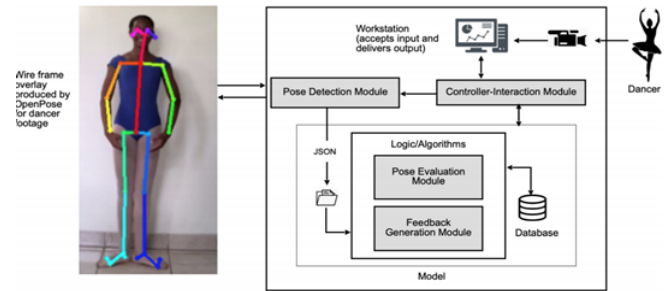


Figure 24. Learning Effect Model for Group Learning on GIFT Platform (LEM)[68]

GIFT is used for multi-uses and multi-purposes, and is also connected to Kinect, Game, and PowerPoint.



High level sketch of the overall architecture of DanceTutor.

Figure 25. The Connection between GIFT and Kinect[68].

STEP-BY-STEP MODE IN ITS FOR TUTORING

RQ 1: What is a step-by-step tutoring mode?

Step-by-step mode in intelligent tutoring systems is a pedagogical approach designed to guide the learners by decomposing tasks into small and manageable steps and solving complex problems. Unlike systems that only give the final answer at once, this approach promotes 'active learning' by scaffolding the learning process. Step-by-step mode realizes immediate feedback for each action of the student and gives hints whenever necessary [77-81].

RQ 2: What are the key features of step-by-step intelligent tutoring system?

a. Scaffolding & Decomposition

Large tasks are decomposed into small tasks, which reduces the cognitive burden and allows the learner to solve a subproblem at one time [77].

b. Immediate Feedback (Inner Loop)

The system does not wait for the student to complete the large task, but evaluates each step so that the defect can be corrected and completed immediately [78].

c. Adaptive Hints

If a student fails at a certain learning stage, the ITS provides hints from general reminders to explicit instructions based on the students' level [79].

d. Proactive Intervention

The systems can intervene with the student's learning if a student's learning is "stuck". This provides direct training data related to the early step of the learning step that are currently facing difficulties or presents examples [80].

RQ 3: How is the step-by-step mode implemented in modern ITSs?

In the following, we illustrate various systems in which a step-by-step tutoring approach is implemented.

a. AI Tutoring Platforms

Platforms such as "Study Mode" in ChatGPT or Gemini do not allow direct answers of users. Instead, they activate a guidance mode that uses questions that decompose problems [81].

b. Educational Software

Step-by-step tutoring systems such as Hawkes Learning use interactive steps to guide students in subjects such as mathematics or microeconomics [80].

c. Programming Supporting

These systems (e.g., Stitch) analyze the student's code, compare it with the goal solution, and provide a step-by-step guide to correct errors.

d. Example-Based Feedback

For example, a system like Step Tutor. Students see example codes or case problems before and after their activities, compare their results or reflect on their results.

RQ 4: (Pedagogical Advantages) What advantages do the step-by-step tutoring mode have over other methods from the pedagogical point of view?

a. It increases the effect of positive direction of learning [79].

The step-by-step tutoring mode has a greater effect on student learning than the traditional computer-based approach. This is possible only when the student is actively involved in the learning activity, not simply providing an answer.

b. This mode is very effective in correcting errors [78].

This mode often uses positive reinforcement for successful learning steps and constructive correction for incorrect steps.

c. Adjusting individual learning pace

Students are taught with their own learning speed. More skilled learners receive very little or very subtle hints, but relatively less skilled learners are more directed [77].

d. When using the step-by-step mode, conceptual understanding is further facilitated.

By focusing on the problem-solving process, students can better understand the underlying principles.

From Question 1 to Question 4, it can be seen that the step-by-step tutoring mode has some features compared to the other tutoring methods [77-81]. One is the difference in the "answer" mode. The traditional approaches are to provide immediate final answers, while the step-by-step approach is to encourage "student to do the work". The other is the difference between the step-by-step and computer-aided instructions. While in Computer-Aided Instruction (CAI) feedback is simply implemented, the step-by-step provides more precise and adaptive inner-loop learning guidance.

AUTHORING AND CHECKING (PLAYING) FOR STEP-BY-STEP TUTORING

The above classifies intelligent tutoring systems as frameworks and platforms. This intelligent tutoring system also consists of student models, tutoring models, domain models, and interfaces. The tutoring model can be viewed pedagogically, systematically, and AI.

The tutoring model is considered only from a pedagogical point of view as follows. The tutoring model plays the role of "how" in the system. The meaning of this 'how' is interpreted in two ways. One is the pedagogical mode, and the other are the instructional contents that is necessarily necessary or is considered to be a student's solution. In relation to the pedagogical approach, the terms Active Learning, Scaffolding, and Proactive Intervention are used. The term "active learning" does not mean that the student immediately gives the entire answer to the problem, but that the student himself will be able to elicit the answer by himself through the process of solving the problem. The term "scaffolding" is used to divide a problem that must be solved into several subproblems, meaning that each subproblem is solved by a student's effort completely through an internal iterative loop. For this reason, when using the term scaffolding, the term "Scaffolding for Problem-Solving" is often used. The term scaffolding also means that only when each subproblem is completely solved, does the solution to the other subproblem at its upper level. The system can interfere with the student's learning if the student's learning is challenged. This provides direct learning data related to the early state of the learning state that are currently facing difficulties or presents examples. This is a "proactive intervention".

This section presents the methods of intelligent tutoring authoring and checking using only step-by-step tutoring. The step-by-step tutoring model mainly focuses on the application of pedagogical modes, which embody the method and principle of "active learning", "scaffolding", and "proactive intervention". Thus, as the pedagogical modes, hints, scaffolding for problem-solving, and proactive intervention are used.

Step-by-Step Tutoring Principle

In the past, the research and development of intelligent tutoring systems has proposed step-by-step in the process of developing a research to implement the teaching methods of human teachers in machines (Vande Pol et al. [46]). This approach is a way of implementing a framework called KCDs (Knowledge Construction Dialogues), which guides all students according to a pre-planned "guide direction of inference", taking into account that students' abilities are different, KCD applies how to exit from the main course of the course and return to the main path before it, when the student has a wrong answer, for the sub-session for "therapeutic" teaching. This approach requires repeated learning for a long time and does not take into account the individual student's level of knowledge, which can be disappointing and inefficient for some students.

Our research and development of intelligent tutoring systems seeks to implement the step-by-step intelligent tutoring function that implements this step-by-step tutoring principle. We assume that the intelligent teacher using the step-by-step pedagogical approach performs the following instructional activities. The teacher constructs few static student models manually using his or her teaching experience and academic knowledge, apply a step-by-step approach to teaching, and return to the lesson when judged to have failed to understand the content of the current teaching. That is, the lecture on the subject that is not understood is repeated, we revisit the supplementary lecture on the wrong problem and then the previous question was re-presented and the 10 points were passed to continue the next topic. In this way, the teacher's

unilateral teaching flow prevails because he is not in the mind of the student, the teacher is motivated to take education and the student is in a passive state. It is also a unilateral teaching mode that does not take into account each individual student, i.e., each student will have different learning efficiencies and different learning outcomes.

Step-by-Step Tutoring Contents Data Structures

The subject teacher edits the necessary data in text, creates it using the provided authoring tool, and then imports the created step-by-step tutoring contents into the ITS Checker (or Player) to test or evaluate it.

First, the system designer must submit authoring specifications for the users of the authoring tools. The authoring specifications include data structure formulations and problem formulations for users of authoring tools. Next, the author creates data structures for his subject by editing it on a computer or writing it on a paper. Then, the author uses this data structures and the authoring tools to create step-by-step tutoring contents. The process flow from the presentation of system designer's authoring specifications to the creation of the tutoring contents is shown in Figure 26.



Figure 26. The Process Flow from Submitting Authoring Specifications to Creating Tutoring Contents

This data structures consist of domain model data, basic data of student model, and basic data of pedagogical model. In the data structure shown below, the labeled rows 1 refer to domain model data, the labeled rows 2 refer to basic data of student model, and the labeled rows 3 refer to basic data of pedagogical model.

Developers (programmers who develop authoring tools) refer to a database that is a product of authoring, in addition to data structures formulations for authoring tool users, and problem formulations for authoring tool users. Data structures formulations, problem formulations, and example data structures are as follows.

a. Data Structures Formulations

{[Topic 1(///Domain model data///)], [Question and correct Answer pair in topic 1 (QA pairs)(/// basic data of student model ///), Supplementary lecture material for the question and answer pair in topic 1 (///basic data of pedagogical model ///)] (///There may or may not be a pair of question and correct answers on the topic ///)}, ... , ..., {[Topic n(///Domain model data///)], [Question and correct Answer pair in topic n (QA pairs)(/// basic data of student model ///), Supplementary lecture material for the question and answer pair in topic n (///basic data of pedagogical model ///)] (///There may or may not be a pair of question and correct answers on the topic ///)}, The question and answer pair (QA pair) in the section (///basic data of student model ///)}

b. Problem Formulations

The form of the problem is as follows.

Type 1: Single-choice mode

Type 2: Multiple selection

Type 3: Blank filling

Form 4: Combination of Combo-box and Blank Filling

Type 5: Response mode with natural language

c. Data Structure Examples

[Chapter 1] Basics

1: [Section 1] Concepts

///// a .ppt (or a picture), necessary videos and comment texts on subject 1. Several .ppt (or pictures), several videos, and several comment texts are available. There must be at least one .ppt for the topic i. /////

/*---It's a ppt (or a picture) of topic 1 and domain model data. ---*/

1: (topic 1 - ppt 1) Slide1.png

/*--This is the comment text on the first ppt of topic 1. --*/

1: (Explanation 1) I will study the subject from this time.

/*--This is the second ppt of topic 1. --*/

1: (topic 1 - ppt 2) Slide5.png

/*---This is the comment text corresponding to the second ppt of topic 1. ---*/

1: (Explanation 1) The course of this subject is as follows.

/*---This is a video on topic 1. At this point, only one video is allowed. ---*/

1: (video 2)

/*---At this point, only one video is allowed. This is QA pairs in Subject 1. ---*/

2: (Question and correct answer pair in topic 1)

/*---Several problems can be presented. The form of the problem is of various forms. ---*/

/*---The following is a multi-choice problem. ---*/

2: [Question 1, 2] The English source of AI is artificial intelligence. What is the meaning of English artificial?

①imitation

②spurious

③human

[answer] ① 5 ② 5 ③ 0

/*---The following is a multi-choice problem. ---*/

[답] ① 4 ② 3 ③ 3

2: [Question 2, 2] What is the meaning of English intelligence?

① the ability to acquire and apply knowledge

② the ability to think and reason

③ ability to understand and benefit from experience

[answer] ① 4 ② 3 ③ 3

/*---The following is a multi-choice problem. ---*/

2: [Question 3, 2] What are there in human intelligence?

① Human intelligence includes thinking, reasoning, oral speaking, ear listening, language, visual understanding, nose smell, and skin feeling.

② Intelligence of man involves intelligent behavior and action.

[answer] ① 5 ② 5

/*---The following is a multi-choice problem. ---*/

2: [Question 4, 2] Why do AI technologies and products need?

①To replace what man does

②To enhance human intelligence.

③To make an autonomous being like a human being.

[answer] ① 3 ② 3 ③ 4

/*---The following is a combination of COMBOX and Blank Filling. ---*/

2: [Question 5, 4] Theory and development of ②? systems that perform tasks of increasing ① such as perception, classification, learning, abstraction, reasoning, and behavior

[answer] ① human' ability, machine intelligence ② computer

/*---The following is a natural language-based response model. ---*/

2: [Question 6, 5] What is AI defined in a broad and general sense?

/*---This are the supplementary lecture materials, a basic data for tutoring model, corresponding to the QA pairs in topic 1. The format is the same as the domain model data type.---*/

3: (ppt 2-1) Slide6.png

3: (Explanation 2-1) Artificial intelligence means counterintelligence.

3: (video 2-1) 1-1.mp4

3: (ppt 2-2) Slide7.png

3: (Explanation 2-2) Human intelligence includes perception, classification, learning, abstraction, reasoning, behavior, and so on.

3: (video 2-2) 1-1.mp4

3: (ppt 2-3) Slide7.png

3: (Explanation 2-3) Artificial intelligence technology and products began to develop from the human needs to make machines that replace human intelligence and function or implement them at a higher level, and then, as technology and computing resources rapidly evolve, they are moving towards the realization of the desire to develop fully autonomous mechanical agents.

3: (video 2-3)

3: (ppt 2-4) Slide7.png

3: (Explanation 2-4) AI is about the theory and development of computer systems that, to a narrow extent, perform tasks that augment human intelligence such as perception, classification, learning, abstraction, reasoning, and behavior, and to develop fully autonomous agents in a wide range.

3: (video 2-4)

/*--- It's a ppt (or a picture) of topic 2. ---*/

1: (topic 2 - ppt 1) Slide8.png

1: (Explanation 1)

1: (video 1)

/*---This is the final test entry for the section. It is basic data for tutoring model.---*/

2: (Chapter 1. Section 1. Final test)

Step-by-Step Tutoring Contents Database for Authoring Tools Developers

The Step-by-Step Tutoring Contents data structures used by the authoring tool and the checker is implemented using the database, which is as follows.

Step-by-Step Tutoring Contents Checking (Playing) Algorithm for Authoring Tools Developers

The step-by-step tutoring contents checking (playing) algorithm works in units of LS_t , each time a lecture topic.

[Algorithm 2. Step-by-Step Tutoring Contents Checking (Playing) Algorithm]

```

1: function Step-by-Step_Tutoring_Contents_Checking
   (GT_MDB)
2: {
3:   for(i=0; i<n_max; i++) // n_max=max{n_s, n_e, n_v, n_q}
4:   {
5:     ith slide display;
6:     ith movie playback;
7:     for(j=0; j<n_q; j++)
8:     {
9:       jth question;
10:      jth answer;
11:      while (result mark < threshold
   mark)
12:      {
13:        Additional lecture on the
14:        jth query presentation ;
15:      }
16:    }
17:  }
18: }

```

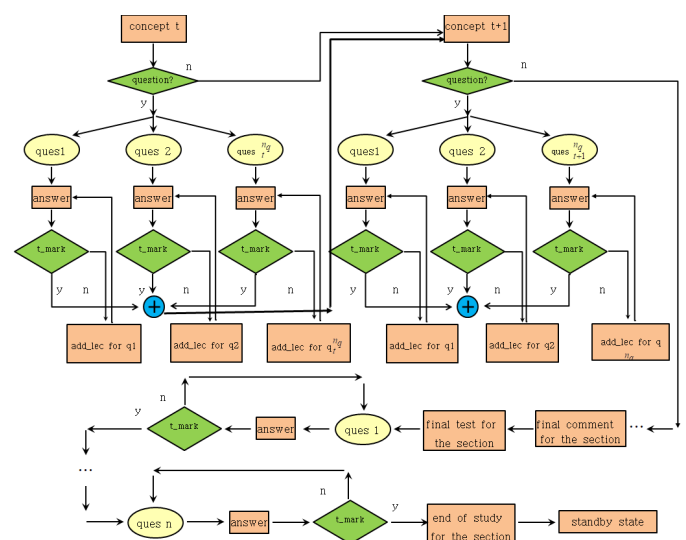


Figure 27. Step-by-Step Tutoring Contents Flow Diagram.

EVALUATION ANS ANALYSIS

This paper proposes a design to construct a step-by-step teaching model that reproduces the classroom teaching style of human teachers. We evaluated the degree to which the proposed authoring tool was preferred by the teachers and analyzed its causes. We also evaluated and analyzed the student achievement using the tutoring model created using the proposed authoring tool(step-by-step).

Evaluation of Preferences for the Authoring Tool

Teachers were selected as participants for evaluation. These volunteer 60 different field experts (teachers) were provided with the Intelligent Tutoring System authoring tools. The proposed authoring tools are a step-by-step tutoring contents authoring tool, a tool using only the knowledge graph, and a tool using partially observable Markov Decision Process (POMDP).

Table 1. Step-by-Step Tutoring Contents Database

Record name	Data type	Main key	Meanings
Chapter table			
Chapter ID	INTEGER	√	Chapter number
Chapter Title	TEXT		Chapter title
Section table			
ID	INTEGER	√	unique identifier of section
Chapter ID	INTEGER		the number of the chapter in which the section belongs
Section ID	INTEGER		the number of the section in the chapter
Section Title	TEXT		section title
Lecture table			
comment: Each slide is represented as a line of this table and is separated by the value of the SectionID field.			
In order to obtain the slide information for Section 1.2, we obtain the unique identifier(ID) of the row corresponding to Section 1.2 in the table and return only the rows with the same value and Section ID.			
ID	INTEGER	√	unique identifier of slide
Section ID	INTEGER		Unique number of a section containing a slide, in what section is the slide?
Slide Name	TEXT		"Subject name/group teaching/slide"
			Name of the picture file in the path (including the extension)
Comment	TEXT		comment
Video Name	TEXT		"Subject name/group teaching/slide"
			Name of a movie file in the path (including an extension)
Show Order	INTEGER		Numerical values for slide display order
has Test	INTEGER		If there are questions for slide, then 1, 0, if no.
Final Test table			
comment:			
In group teaching, there is a final question at the end of each section.			
The information is a table. As in this table, in all test-related tables, the test data is entered in XML format. The test editor that prints the test problem in XML has been developed and is therefore used. Each row represents one test problem. Thus, the way to obtain the necessary test data is as in the table above, so that Parent Section only obtains the same rows as the unique number of the section.			
ID	INTEGER	√	unique identifier
XML Data	TEXT		XML data for testing
Parent Section	INTEGER		corresponding section number
Table for Slide Test			
comment: In the slide of the Lecture table, if "has Test" is 1, there is a slide. Then, we get the same lines of ID value of the slide line and the same value of Parent Slide in the Slide Test table and then ask.			
ID	INTEGER	√	unique identifier
XML Data	TEXT		XML data for testing
Parent Slide	INTEGER		unique number of the slide, The ID of slide in the Lecture table.
Slide Name	TEXT		The name of the picture file (including extension) in the path of "the subject name/group teaching/slide"
Comment	TEXT		comment
VideoName	TEXT		The name of the movie file (including extension) in the path of "the subject name/group teaching/slide"

The teachers were asked to examine various authoring tools and use a most desirable tool to create the content of their subjects' intelligent tutoring. The experts in each field included 10 experts in information science, 9 experts in artificial intelligence, 20 mathematical experts, 15 experts in physics, and 6 biologists. As a result, the percentage of the authoring tools that teachers chose was as follows (Table 2).

Table 2. selection rate of authoring tools (%)

	selection rate
step-by-step	80 (%)
knowledge graphs	15 (%)
POMDP	5 (%)

As a result, the proportion of selecting the step-by-step tutoring model authoring tool was 80%, the proportion of selecting tool using only the knowledge graph was 15%, and the proportion of tool using POMDP was 5%. The highest proportion of teachers selected the step-by-step model authoring tool. The reason for this is that teachers could easily understand the principles of creation of the authoring tool and make it easier to create them.

Assessment of Learners' Performance Gains using a Step-by-Step Tutoring Model

One hundred students responded to collaboration for evaluation. The students were divided into 50 students each.

One group was asked to self-study on the subject studied alone, and the other group was asked to use a step-by-step tutoring model to self-study. After completing the study of the three sections, the tests were conducted and the results are shown in the table below.

Table 3. Assessment of Student Achievement using a Step-by-Step Tutoring Model

	Mean value
Learners' performance gains (self-study)	6.8
Learners' performance gains (using step-by-step)	8.4

As a result, we were able to ascertain that students' academic performance using the step-by-step tutoring model was on average 1.6 higher.

CONCLUSION

This paper proposes a step-by-step tutoring contents authoring tool design specifications. We have proposed a pedagogical principle for step-by-step tutoring authoring tools and checkers, data structure, database, and tutoring content's guidance algorithm. The experiments confirmed that the creation principle of the proposed authoring tool was easy to understand by the general teachers, thus enhancing the preference of teachers. It also confirmed that the learners' performance gains using the tutoring model created using this authoring tool was on average 1.6 higher than that of students who had studied on their own.

Abbreviations

ITSs	Intelligent Tutoring Systems
POMDP	Partially Observable Markov Decision Process
LMSs	Learning Management Systems
KG	Knowledge Graph
QA	Question and Answer

Author Contributions

Gwang-Hyok Kim: Conceptualization, Methodology

Song-Hwan Kwon: Writing-original draft preparation, Project administration

Gyong-Jin Kim: Investigation, Software

Gang-Bom So: Software

Sin-Hye Jang: Verification

Ha-Gyong Kim: Resources, Validation

Jong-Nam Rim: Investigation, Software

Chung-Song Ko: Software, Formal analysis

Song-Il Pak: Writing-review & editing

Chol-Jun Hong: Software, Data curation

Myong-Il Kim: Software

Conflict of Interest

The authors hereby declare no conflict of interest.

REFERENCES

- Xiaohan Z. (2020). A Survey on Application of Knowledge Graph. IOP Conf. Series: Journal of Physics: Conf. Series 1487 (2020) 012016, <https://doi.org/10.1088/1742-6596/1487/1/012016>.
- Chen, P., Lu, Y., Zheng, V.W., Chen, X., Yang, B. Knowedu (2018): A system to construct knowledge graph for education. IEEE Access 6, 31553–31563 (2018).
- Khazanchi, R. (2021). Impact on Intelligent Tutoring System (ITS) on Mathematics Achievement Using ALEKS, SPRINGER NATURE Link, 2021, pp 471-475.
- V, T, K. A. (2021). Intelligent Tutoring Systems for Personalized Mathematics Learning: A TPACK Perspective, SPRINGER NATURE Link, 2021, pp 206-220.
- Van K. L. (2026). ASSISTments Dataset 2009-2010, IEEE DataPort, DOI:10.21227/xgb2-tb53.
- Kinshuk (1996). Computer aided learning for entry level Accountancy students. PhD Thesis, De Montfort University, England, July 1996.
- Corbett A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent Tutoring Systems. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), Handbook of human-computer interaction (pp. 849-874). Amsterdam: Elsevier.
- Nwaha, H. S. (1990). "Intelligent tutoring systems: An overview". Artificial Intelligence Review 4(4): 251-277. doi: 10.1007/bf00168958. <https://dx.doi.org/10.1007/2Fbf00168958>.
- Freedman, R. (2000). What is an intelligent tutoring system? Intelligence 11(3): 15-16. doi: 10.1145/350752.350756. <https://dx.doi.org/10.1145/2F350752.350756>.
- Nkambou, R., Mizoguchi, R., & Bourdeau, J. (2010). Advances in intelligent tutoring systems. Heidelberg: Springer.
- Mitrovic, A., Ohlsson, S. (2006) Constraint-Based Knowledge Representation for Individualized Instruction. Computer Science and Information Systems, 3(1), 1-22.
- Ohlsson, S., Mitrovic, A. (2007) Fidelity and Efficiency of Knowledge representations for intelligent tutoring systems. Technology, Instruction, Cognition and Learning, 5(2), 101-132.
- Anderson, H., Koedinger, M. (1997). intelligent tutoring goes to school in the Big City. International Journal of Artificial Intelligence in Education, 60(1), 264-272.
- Sottolare, R. (2012). Considerations in the development of an ontology for a generalized intelligent framework for tutoring. In 12M defense and homeland security simulation Conference (DHSS 2012).
- McLaren, B. M., DeLeeuw, K. E., & Mayer, R. E. (2011). Polite web-based intelligent tutors: Can they improve learning in classrooms? Computers & Education, 56(3), 574-584. doi: 10.1016/j.compedu.2010.09.019.
- ZerKouk, M. et al. (2025). A comprehensive Review of AI-based Intelligent Tutoring Systems Applications and Challenges, arXiv: 2507.18882v1.
- Sghaier, G. et al. (2025). A systematic literature review to implement large language model in higher education: issues and solutions, Discover Education (2025) 4:35 | <https://doi.org/10.1007/s44217-025-00424-7>.
- Carbonell, J. R. (1970). Mixed-initiative man-computer instructional dialogues (Ph.D. thesis), Massachusetts, USA: Massachusetts Institute of Technology. Dept. of Electrical Engineering.
- Brown, J. S., Burton, R. R., & Bell, A. G. (1975). SOPHIE: A step toward creating a reactive learning environment. International Journal of Man-Machine Studies, 7(5), 675–696.
- Clancey, W. J. (1979). Tutoring rules for guiding a case method dialogue. International Journal of Man-Machine Studies, 11(1), 25–49.
- Stevens, A. L., & Collins, A. (1977). The goal structure of a socratic tutor. In Proceedings of the 1977 annual conference (pp. 256–263). New York, NY, USA: Association for Computing Machinery.
- Pablo, A. G. et al. (2025). A framework for adapting conversational intelligent tutoring systems to enable collaborative learning, Expert Systems With Applications 271 (2025) 126663, <https://doi.org/10.1016/j.eswa.2025.126663>.
- Kinshuk, and Ashok, P. (1997) A Conceptual Framework for Internet Based Intelligent Tutoring Systems. Knowledge Transfer, II, 117-24.
- Tommaso, C. et al. (2024). Towards Educator-Driven Tutor Authoring: Generative AI Approaches for Creating Intelligent Tutor Interfaces, arXiv: 2405.1471v1, <https://doi.org/10.1145/3657604.3664694>.
- Priynka, S., Mayuri, H. (2022). Designing an intelligent tutoring system for computer programming in the Pacific, Education and Information Technologies (2022) 27:6197–6209, <https://doi.org/10.1007/s10639-021-10882-9>.
- Erica, M. et al. (2001). ACTIVEMATH: A Generic and Adaptive Web-Based Learning Environment, Carsten Ullrich DFKI Saarbrücken, D-66123 Saarbrücken, Germany November 20, 2001.
- MUHAMMAD, A. et al. (2020). The transition from intelligent to affective tutoring system: A review and open issues, doi 10.1109/access.2020.3036990, IEEE Access.
- Linqing, L. et al. (2023). Knowledge Graph Enhanced Intelligent Tutoring System Based on Exercise Representativeness and Informativeness, arXiv: 2307.15076v1.
- Guixia, W., Zehui, Z., Shouyuan, Q. (2025). Synergizing Knowledge Graphs and LLMs: An Intelligent Tutoring

- Model for Self-Directed Learning, *Educ. Sci.* 2025, 15, 1102 <https://doi.org/10.3390/educsci15091102>.
30. Chenxi, D. et al. (2023). How to Build an Adaptive AI Tutor for Any Course Using Knowledge Graph-Enhanced Retrieval Augmented Generation (KG-RAG), *arXiv*: 2311.17696.
 31. Mei-Ling, C. et al. (2026). Knowledge Graph-Enhanced Retrieval for Personalized AI Tutoring, *arXiv*:2611.17696.
 32. Tom, M. (1999). Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art, *International Journal of Artificial Intelligence in Education* (1999), 10, 98-129.
 33. Vincent, A. et al. (2025). An Integrated Platform for Studying Learning with Intelligent Tutoring Systems: CTAT+TutorShop, *arXiv*:2502.10395.
 34. Kenneth H. et al. (2018). Opening Up an Intelligent Tutoring System Development Environment for Extensible Student Modeling, *AIED 2018, LNAI 10947*, pp 169-183, 2018, <https://doi.org/10.1007/978-3-319-93843-1-13>.
 35. Abu Naser, S. (2016). ITSB: An Intelligent Tutoring System Authoring Tool, *Journal of Scientific and Engineering Research*, 2016, 3(5):63-71.
 36. Alevan, V., Sewall, J., McLaren, B. M. & Koedinger, K. R. (2006). Rapid authoring of intelligent tutors for realworld and experimental use. In Kinshuk, R. Koper, P. Kommers, P. Kirschner, D. G. Sampson & W. Ditteren (Eds.), *Proceedings of the 6th IEEE international conference on advanced learning technologies (ICALT 2006)* (pp. 847-851). Los Alamitos, CA: IEEE Computer Society.
 37. Robert A. S. (2014). Examining Opportunities to Reduce the Time and Skill for Authoring Adaptive Intelligent Tutoring Systems, *Proceedings of the 2nd Annual GIFT Users Symposium (GIFTSym2)*, 44-48.
 38. MacLellan, C. J., Wiese, E. S., Matsuda, N. & Koedinger, K. R. (2014). SimStudent: Authoring Expert Models by Tutoring. *GIFTSym2*, Pittsburgh, PA.
 39. Diego, D. et al. (2018). Authoring Tools for Designing Intelligent Tutoring Systems: A Systematic Review of the Literature, *Int J Artif Intell Educ* (2018) 28:336-384, <https://doi.org/10.1007/s40593-017-0157-9>.
 40. Alevan, V., McLaren, B. M., Sewall, J., & Koedinger, K. (2006). The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains. In M. Ikeda, K. D. Ashley, & T. W. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*, (pp. 61 -70).
 41. Keith, B. (2015). *Authoring Tools for Adaptive Training – An Overview of Systems and Taxonomy for Classification*, Springer International Publishing Switzerland 2015, D.D. Schmorow and C.M. Fidopiastis (Eds.): AC 2015, LNAI 9183, pp. 562-569, 2015. DOI: 10.1007/978-3-319-20816-9_54.
 42. Robert, S. et al. (2017). Defining Complexity in the Authoring Process for Adaptive Instruction, Springer International Publishing AG 2017, D.D. Schmorow and C.M. Fidopiastis (Eds.): AC 2017, Part II, LNAI 10285, pp. 237-249, 2017. DOI: 10.1007/978-3-319-58625-0_17.
 43. Benjamin, D. et al. (2014). AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring, *Int J Artif Intell Educ* (2014) 24:427-469, DOI 10.1007/s40593-014-0029-5.
 44. Tom, M. (2016). Coordinating the Complexity of Tools, Tasks, and Users: On Theory-based Approaches to Authoring Tool Usability, *Int J Artif Intell Educ* (2016) 26:37-71, DOI 10.1007/s40593-015-0076-6.
 45. Fletcher, J. D. et al. (2018). Shared Mental Models in Support of Adaptive Instruction for Teams Using the GIFT Tutoring Architecture, *Int J Artif Intell Educ* (2018) 28:265-285, DOI 10.1007/s40593-017-0147-y.
 46. Vande Pol, J., Volman, M., Beishuizen, J. (2010). Scaffolding in teacher-student interaction: a decade of research. *Educ. Psychol. Rev.* 22, 271-296.
 47. Robert A. et al. (2011). Using Learner Data to Influence Performance during Adaptive Tutoring Experiences, *adfa*, p. 1, 2011. © Springer-Verlag Berlin Heidelberg 2011.
 48. Tiffany, P., Robert, S. (2011). Research to Practice: Adaptive Technologies and Approaches to Performance Improvement, *adfa*, p. 1, 2011. © Springer-Verlag Berlin Heidelberg 2011.
 49. Qwaider, S. R., Abu-Naser, S. S. (2018). Excel Intelligent Tutoring System. In: *Proceedings* (2018).
 50. Caleb, V. et al. (2022). Automated Assessment of Team Performance Using Multimodal Bayesian Learning Analytics, 2022 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), I/ITSEC 2022 Paper No. 22258 pp 1-14.
 51. James, R. S. et al. (2015). Designing Representations and Support for Metacognition in the Generalized Intelligent Framework for Tutoring, D.D. Schmorow and C.M. Fidopiastis (Eds.): AC 2015, LNAI 9183, pp. 663-674, 2015. DOI: 10.1007/978-3-319-20816-9_63.
 52. Benjamin, G. (2015). Augmenting Instructional Practice in GIFT Using the Engine for Management of Adaptive Pedagogy (EMAP), D.D. Schmorow and C.M. Fidopiastis (Eds.): AC 2015, LNAI 9183, pp. 595-604, 2015. DOI: 10.1007/978-3-319-20816-9_57.
 53. Benjamin, G. (), Feedback Source Modality Effects in Game-Based Training: A Trade-Off Analysis, *Proceedings of the Human Factors and Ergonomics Society 59th Annual Meeting – 2015*.
 54. James. R. S. et al. Demonstration: Using GIFT to Support Students' Understanding of the UrbanSim Counter Insurgency Simulation, *Seventeenth International Conference on Artificial Intelligence in Education (AIED 2015)*, pp 17-23.
 55. Robert A. S. (2011). Modeling Shared States for Adaptive Instruction, *adfa*, p. 1, 2011. © Springer-Verlag Berlin Heidelberg 2011.
 56. Benjamin, G. et al. (2012). Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies, *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012*.
 57. Keith, B. et al. (2012). Recommendations for Modern Tools to Author Tutoring Systems, *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012*.
 58. Robert, A. S. et al. A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS), *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012*.
 59. Colin, R., Stephen, G. (). Bringing Authoring Tools for Intelligent Tutoring Systems and Serious Games Closer Together: Integrating GIFT with the Unity Game Engine, *Recommendations for Authoring, Instructional Strategies and Analysis for Intelligent Tutoring Systems (ITS): Towards the Development of a Generalized Intelligent framework for Tutoring (GIFT)*, *AIED 2013 Workshops Proceedings Volume 7*, pp37-40.
 60. Robert, S., Benjamin, G. (2012). Designing Adaptive Computer-Based Tutoring Systems to Accelerate Learning and Facilitate Retention, *Cognitive technology*, volume 17, issue 1, 2012.

61. Arthur, G. (2013). A Guide to Understanding Learner Models, Design Recommendations for Intelligent Tutoring Systems - Volume 1.
62. Goldberg, B., Sinatra, A., Sottolare, R., Moss, J., & Graesser, A. (2015). Instructional Management for Adaptive Training and Education in Support of the US Army Learning Model-Research Outline (No. ARL-SR-0345). Army research lab aberdeen proving ground md human research and engineering directorate.
63. Robert A. S. (2017). Challenges in Authoring, Instructional Management, and Evaluation Methods for Adaptive Instructional Systems, or publication in the Proceedings of the American Education Research Association (AERA) Conference Technology, Instruction, Cognition & Learning Special Interest Group (AERA TICL SIG) Symposium on Intelligent Tutoring Systems, BIG DATA–Learning Analytics, and Automated Humanlike Tutoring: Similarities and Differences in San Antonio, Texas, 28 April 2017.
64. Robert A. S. (2017). Adaptive instructional methods to accelerate learning and enhance learning capacity, Proceedings of the International Defense and Homeland Security Simulation Workshop, 2017, ISBN 978-88-97999-90-4; Bruzzone, Cayirci and Sottolare Eds.
65. Robert A. S. et al. (2018). Special Issue on the Generalized Intelligent Framework for Tutoring (GIFT): Creating a Stable and Flexible Platform for Innovations in AIED Research, *Int J Artif Intell Educ* (2018) 28:139–151, DOI 10.1007/s40593-017-0149-9.
66. Scott O., Robert A. S., Keith, B. Task allocation and automation level in the design of intelligent tutoring system authoring tools, Proceedings of the International Defense and Homeland Security Simulation Workshop, 2017, ISBN 978-88-97999-90-4; Bruzzone, Cayirci and Sottolare Eds.
67. Benjamin S. G. et al. (2011). Enhancing Performance through Pedagogy and Feedback: Domain Considerations for ITSS, Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2011, No. 11010, pp 3-12.
68. Robert A. S. (2018). A Comprehensive Review of Design Goals and Emerging Solutions for Adaptive Instructional Systems, *Tech., Inst., Cognition and Learning*, Vol. 11, pp. 5–38, © 2018 Old City Publishing, Inc.
69. Michael, A. B. et al. (2011). A New Approach to Incremental Cycle Detection and Related Problems, arXiv: 1112.0784v1.
70. Edith, C. et al. (2013). A Labeling Approach to Incremental Cycle Detection, arXiv: 1310.8381v1.
71. Armaël, G. et al. (2019). Formal Proof and Analysis of an Incremental Cycle Detection Algorithm, 10th International Conference on Interactive Theorem Proving (ITP 2019). Article No. 18; pp. 18:1–18:20, Leibniz International Proceedings in Informatics.
72. Wang, F. (2015). Handling exponential state space in a POMDP-based intelligent tutoring system, in Proc. 6th International Conference on E-Service and Knowledge Management (IIAI ESKM 2015), pp. 67-72, 2015.
73. Folsom-Kovarik, J. T., Sukthankar, G., and Schatz, S. (2013). Tractable POMDP representations for intelligent tutoring systems. *ACM Trans. Intell. Syst. Technol.* 4, 2, Article 29 (March 2013), 22 pages. DOI = 10.1145/2438653.2438664
<http://doi.acm.org/10.1145/2438653.2438664>.
74. Sottolare, R.A., Holden, H.K.: Motivations for a Generalized Intelligent Framework for Tutoring (GIFT) for authoring, instruction and analysis. In: AIED 2013 Workshops Proceedings, vol. 7, pp. 1–9 (2013).
75. Sottolare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED), 2012.
76. Sottolare, R.A., Brawner, K. W., Goldberg, B., & Holden, H. (2013). The Generalized Intelligent Framework for Tutoring (GIFT). In C. Best, G. Galanis, J. Kerry & R. Sottolare (Eds.), *Fundamental Issues in Defense Training and Simulation* (pp. 223-234). Burlington, VT: Ashgate Publishing Company, 2013.
77. Yuan, S. et. al. (2025). Stitch: Step-by-step LLM Guide d Tutoring for Scratch, arXiv: 2510.266334v1.
78. Wengran, W. et. al. (2020). Step Tutor: Supporting Students through Step-by-Step Example-Based Feedback, ITiCSE'20, June 15–19, 2020, Trondheim, Norway, <https://doi.org/10.1145/3341525.3387411>.
79. Anohina, A. (2007). Advances in Intelligent Tutoring Systems: Problem-solving Modes and Model of Hints, *International Journal of Computers, Communications & Control* Vol. II (2007), No. 1, pp. 48-55, <https://doi.org/10.15837/ijccc.2007.1.2336>.
80. VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence and Education*, 16, 227–265.
81. [81] Ranganathan, R. et. al. (2014). What do students do when using a step-based tutoring system?, *Research and Practice in Technology Enhanced Learning* Vol. 9, No. 2 (2014) 323 - 347, Ó Asia-Pacific Society for Computers in Education
