Gameful Intelligent Tutoring System: Using BITS and Data-Driven Models to Design an Animated User Interface for Hint Generation

Proposal by: Calvin G. Goah

COMS 3998 W, Section 28

Fall 2017

Table of Contents

1 Introduction

This project aims to improve upon the intelligent hinting system already emplaced with SAGE. The intelligent hinting system is part of the SAGE Intelligent Tutoring System (ITS). An ITS must be able to accurately diagnose a student's knowledge level using principles rather than pre-programmed responses, decide what to do next and adapt instruction accordingly and, finally, provide feedback [2]. Early attempts at creating an ITS resulted in numerous Computer Assisted Instructional systems, which were essentially designed to have a rigid tree structure that, though effective in helping learners, did not consider the variances in student knowledge [2]. The current hinting system model used by SAGE auto-generates hints at various project states, indicating that the current system is incapable of generating hints at every project state, especially in the case of first time users [5]. Furthermore, it is also lacking the granularity necessary for handling variances in student knowledge bases.

With this in mind, the motivation for this project thus founts from the desire to granularize the hinting system such that hint generation is based on individual user data gathered using a combination of the BITS and data-driven models, and to implement an on-demand hinting option for users. Once the hint itself is generated, the focus of this project then shifts towards determining how best to present said information to the User via the Scratch Analyzer UI.

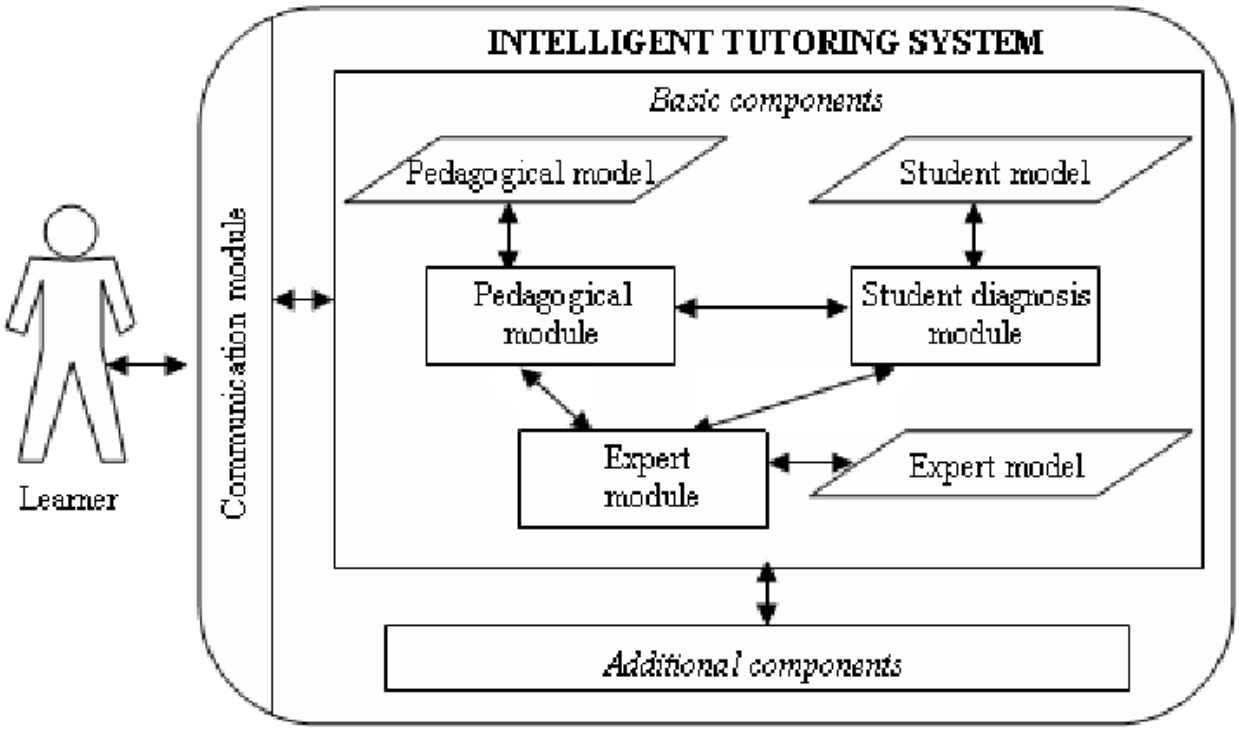
As there is another research team working on the same topic, specifically with regards to data-driven models for generating hints, if it so happens that they are able to implement all the hinting features detailed above before the completion of the strategy detailed in this paper, then this project will use their implementation as a foundation for developing the Scratch Analyzer hinting UI.

2 Related Work

2.1 "Advances in Intelligent Tutoring Systems: Problem-solving Modes and Model of Hints" [1]

This paper defines the concept and the architectural parts of an intelligent tutoring system, mainly that it is an intelligent system because it uses principles and methods of artificial

intelligence such as knowledge representation, inference mechanisms and machine learning in its structure and operation. An intelligent tutoring system is an adaptive system as it alters aspects of its structure, functionality or interface for the particular user and his/her changing needs over



time.

In trying to create a hinting system that encourages problem solving and that is adaptive in providing support, Anohina describes a developed approach based on two modes of problem-solving and a two layer model of hints. These hints are described to range from the most general to the most specific.

Additionally, this paper also discusses the ITS for the Minimax algorithm in which the proposed approach is being implemented.

2.2 “A Web-based Bayesian Intelligent Tutoring System for Computer Programming” [2]

This paper discusses how Bayesian Networks are used in constructing a hinting system. Uses the Bayesian Network as a means of documenting each individual user’s understanding of the material. Goes in depth on how these directed acyclic graphs are constructed from conditional probabilities and how these probabilities are used to determine if a student is ready for new material given that she understands a certain percentage of the prerequisite material.

In addition, the paper also describes BITS adaptive guidance feature which uses Bayesian Networks to generate tailored pedagogical options that support the individual student. In tandem with this support structure is an interactive user interface that fetches relevant material from the preloaded knowledge base for struggling students.

The major drawbacks to BITS is that, for one, having to ask teachers using various curricula to create these Bayesian Networks would be rather cumbersome, and, secondly, BITS isn’t suited for problem-solving, rather it is more of an adaptive process for providing curated help for individual students.

2.3 “New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization” [3]

This paper describes various means by which data-driven ITS can be used to provide hints which aid in the process of problem-solving. It describes the construction of cognitive models that better match student performance. By using SimStudent, a data-driven application for developing intelligent tutoring systems, to collect and analyze student data, it produces the cognitive model component of an ITS that is used for all of its functions: evaluate, suggest, update, and select.

This paper also touches on Hint Factory--a method for automatically generating context specific hints by using previously collected student data--, modeling engagement and affect, the Markov Decision Process, and various data-driven applications for producing better cognitive models.

2.4 “Automatic Generation of Programming Feedback: A Data-Driven Approach” [4]

This paper describes the solution space representation to solving a task. It involves generating a graph representation of all the possible paths a student could take in order to get from the problem statement to a correct answer, where the nodes are candidate solutions and the edges are the actions used to move from one solution state to another. Abstract Syntax Trees (AST) are used to reduce the solution space by getting rid of extraneous elements of written code, such as whitespace and comments, in order to actively track each student’s progression through a specific task.

As in [3], this paper also utilizes the Markov Decision Process to decide which next state to direct the student towards, optimizing for the fastest path to the solution. It also draws upon the Hint Factory construct, but modifies it slightly in order to adjust for the ill-defined nature of solving programming problems, which specifies that different solutions can solve the same problem.

3 Proposal

3.1 Granularize The Hinting Process

One of the goals of this project is to be able to design a hinting system that takes into account the variances in user knowledge. This is going to be accomplished by combining elements of BITS architecture, such as:

- A Personalized Learning feature for assessing student knowledge of a given task
- An Adaptive Guidance feature for offering tailored pedagogical options that support the individual student
- A Bayesian Network for housing student data
- A template DAG for teachers to easily implement 6-8th grade curricula

This project will also strive to incorporate elements of a data-driven approaches, such as:

- Using Hint Factory logic tutor for feedback generation from a solution space
- Utilizing abstract syntax tree edits for determining the additions, deletions, and changes required to turn one tree into another, this finding the differences between the student's response path versus a path that leads to the final answer.
- Utilizing a Markov Decision Process
- Providing hints ranging from very general to very specific
- Utilizing cognitive models (i.e. suggest, update, and select)

If successful in combining these two approaches to SAGE, the hints and pedagogical learning sequences generated could greatly improve the effectiveness of the hints generated through SAGE. Optimizing the data-driven approach to seek out the shortest path to a solution will be instrumental in making this project function properly.

3.2 Create a User Interface for Hints

After the hinting system itself is functional, the next step would be determining how best to offer these hints to the user. At this moment the UI for Scratch supports shaking blocks for hints. With that in mind, these are the various methods that will be used in trying to improve the UI:

- Animating the blocks in Scratch Analyzer such that prerequisite concepts for fixing the current issue are made to migrate/move near one another before going back to their original positions.

- Creating a side window that fetches relevant material from the knowledge base for students to quickly review.
- Creating a button/option for on-demand hinting that generates hints ranging from very general to very specific using our BITS/ Data-Driven model to hinting described in section 3.1.

These three specifications, in tandem with the hinting system created in 3.1, will hopefully be able to contextualize a problem for the user in a gameful manner that promotes fun, but at the same time promotes the building of intuition with regards to computational thinking.

4 Timeline

Milestones	print
Environment Setup	Sprint 0

Implementing BITS/ Data-Driven Architecture	Sprint 1
Combining BITS and Data-Driven Approach	Sprint 2
midterm	-
Creating Scratch UI Animation	Sprint 3
On-Demand Hinting/ Fetching Relevant Information	Sprint 4
Final	-

5 Future Work

- Affect Detection
  - System for detecting boredom and responding in a positive manner
  - Could keep mouse tracking data
  - Could use computer camera to track eyes and other facial features
- Determining if feedback given ends up helping students
  - It would be good to know how the hints provided aide the user in understanding a concept previously unknown to her
- Try to incorporate a Bayesian Network template for teachers to use in 6-8 CS curricula (A strong attempt will be made to get incorporate this into SAGE if time permits)

References

[1]Alla Anohina 2007, Advances in Intelligent Tutoring Systems: Problem-solving Modes and Model of Hints, *International Journal of Computers, Communications & Control Vol. II, No. 1*, pp. 48-55

[2]C.J. Butz, S. Hua, R.B. Maguire 2006, A Web-based Bayesian Intelligent Tutoring System for Computer Programming, *Journal of Web Intelligence and Agent Systems Vol. IV, No. 2*, pp. 77-97

[3]Kenneth R. Koedinger, Emma Brunskill et al. 2013, New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization, *AI Magazine*

[4]Kelly Rivers and Kenneth R. Koedinger, Automatic Generation of Programming Feedback: A Data-Driven Approach

[5]Sambhav Anand and Allison Sawyer 2017, Enhanced Data Collection, Student Progress Modeling, and Intelligent Hinting in SAGE COMS 3998 Sec. 14