Flowchart-based Bayesian Intelligent Tutoring System for Computer Programming

Danial Hooshyar*, 1, Rodina Binti Ahmad1, Moein Fathi1, Moslem Yousefi2, Maral Hooshyar3
1Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
2Center of Systems and Machines Intelligence, College of Engineering, Universiti Tenaga Nasional, 43300 Kajang, Malaysia
3Department of Industrial Engineering, Islamic Azad University Nour Branch, Iran
E-mail: Danial.hooshyar@gmail.com, Rodina@um.edu.my, Moein.fathi@gmail.com, moslem_yousefi@yahoo.com, maral8116@gmail.com

Abstract— There is a misconception of what programming is at the early stages of learning programming for Computer Science (CS) minors. More researches in this field have revealed that the lack of problem-solving skills, which is considered as one of the prominent shortcomings that novices deal with, is exacerbated by language syntax that the novices employ. A Flowchart-based Intelligent Tutoring System (FITS) is proposed in this research aimed at introducing the early stages of learning programming (CS1) to put the record straight. The students who have no prior knowledge of programming are the target audience of this research. In order to support novice programmers in beginning of programming, Bayesian network approach is applied mainly for decision making and to handle uncertainties in knowledge level of students. How to use Bayesian network to take full advantage of it as an inference engine for providing users with various guidance is described in this paper. Therefore, our proposed system provides users with dynamic guidance such as recommend learning goals, recommend options for flowchart development, and generate appropriate reading sequences. Additionally, our proposed system’s architecture and its components are elaborated. Our future work is to evaluate the FITS by conducting an experimental study using novices.

Keywords- Flowchart-base; Intelligent Tutoring System; Problem-Solving skills; Novice Programmers

I. INTRODUCTION

Web courseware environments offer users a personal environment to control their own pace of study with all necessary learning materials so as not to be depended on an instructor’s presence. These environments are even greater than web-based learning systems [1]. Web-based tutoring systems are capable of adding hyperlink which can be clicked on by users for referring to other sites. Moreover, they benefit from multi-media such as video and audio to convey their point in learning process. Though, the majority of these environments are not dynamic, so not only they are not interactive but also they are not adaptive. It means, all users with different knowledge background are displayed the same static learning materials. Thus, the aforementioned type of systems proved to be unable in satisfying the target users [2].

As indicated by Yao and Yao [3], a system should be capable of dealing with different type of users. Additionally, development of system with behavior adaption to its task, interests, and other features to satisfy its users are the only way of fixing the problem of traditional approaches, as stated by Brusilovsky and Maybury [2]. Regarding web-based tutoring system, a smart system was developed by Liu et al. [6], involving decision making based on uncertain information. As mentioned earlier, Bayesian network is one of the frameworks of uncertainty management using probability theory [4, 5]. Bayesian network has already been used by researchers in the field of web intelligence in many tasks such as multi-agent systems [7, 8] and e-commerce [9, 10]. The main aim of this research is to propose a flowchart-based Bayesian intelligent tutoring system for computer programming, to support novice programmers in problem-solving and solution designing skills. Bayesian network handles the decision making process in our proposed system. Using FITS, alike [11, 6], students are assisted in navigation through the online learning materials. However, learning goals, flowchart development, and appropriate reading sequences are recommended by FITS, unlike [11, 6]. For instance, to learn a specific concept without knowing all previous concepts and materials, FITS defines the least prerequisite knowledge required to understand the specific concept along with correct learning sequence. By doing so, the navigation problem of users is fixed and they are all refocused on their study objective by making the tutoring system more adaptable. We believe, the proposed system not only help novice programmers to overcome their difficulties in learning programming, but also aid instructors in teaching programming [12]. In addition, we are convinced that FITS can be very helpful in different field of studies, as it applies a flowchart-based Bayesian ITS along with using an automatic text-to-flowchart approach in its knowledge base module. It is noteworthy to mention that we discovered there has yet to be
any study dealing with this specific issue, flowchart-based Bayesian network ITS in computer programming.

II. RELATED WORKS

Several efforts have been made to develop ITS for programming, namely Lisp-Tutor [13] aimed at teaching lisp programming using rule-based approach, SQL-Tutor [14] aimed at SQL programming benefiting from constraint-based modelling approach, JITs [15] aimed at Java programming using decision tree approach and etc. Although these systems have shown positive feedbacks, the users are not provided with navigation option to navigate through online learning materials while they are given individualized recommendations from the system. Problem-solving improvement with the help of error feedback and updating the student model is the main aim of ITSs. However, FITS is aimed and proposed to cover both problem-solving skills and learning material navigation. Applying Bayesian network in ITS is suggested first by Villano [16]. Though, some researchers like Martin and Vanlehn [17] said the assessment made by Villano cannot reach to this conclusion that what exactly users know and what they do not know. In our proposed system, three color lights show whether the user knows the concept, does not know and not ready to learn, or does not know and ready to learn. Martin and Vanlehn [17] proposed the ITS which only shows what users know, but FITS assesses both known and unknown concepts along with navigation options through unknown concepts. Many systems in this area are proposed and developed for applying Bayesian network in ITS systems [18, 19, 20, 21, 22, 23]. Several applicable techniques were reviewed by Jameson [24] to be applied on ITS systems such as Dempster-Shafer theory of evidence, fuzzy logic, and Bayesian network. Bayesian networks were found to be the best among them all, proved by Pearl [25], and as a result of that this algorithm were chosen to be used in FITS.

III. FITS’S ARCHITECTURE

In this section, we introduce FITS for computer programming. The major components of the proposed system are outlined at this section and the way of their interaction with each other is described as well. In addition, how to apply Bayesian network in FITS for modeling and inference in order to design pedagogical options for each single user is demonstrated. The main components of the proposed system are shown in figure 1.

In overall, FITS benefits from several artificial intelligence techniques to assist users in computer programming such as Natural Language Processing (NLP), Bayesian Network, knowledge base, and knowledge expansion. There is also a multi-agent system improvised at the back side of FITS, locating at knowledge base, comprising NLP, knowledge expansion and knowledge based systems, and a Crawler for online searching of the relevant websites so as to improve the knowledge base without human interference. The multi-agent system has benefited from a novel approach for text-to-flowchart conversion in an automatic manner; this is used to involve the users in flowchart development and problem-solving activities. Moreover, the flowchart-based multi agent system has benefited from a novel automatic text-to-flowchart approach.

A. Bayesian Networks in FITS

In order to offer personalized environments in systems, users’ interaction with the system should be observed to provide a profile for individual user for the system so as to update the Bayesian network. Two different approaches are used in FITS to obtain the profile from users: I) in case the user knows about a specific concept, he/she responds to the query generated by the system, II) in case the user does not know about the concept, a quiz will be generated by the system for that specific concept with the aim of defining whether the user has comprehended the concept or not. As soon as users reply to the queries generated by the system, the Bayesian Network algorithm gets updated and as a result of that some relevant lecturer note or reading materials will be given to the user. Afterwards, the user is expected to provide the system with feedbacks so as to update the algorithm stepwise. The user can simply choose one of the options in the system such as 1) I understand, 2) I do not understand. Regarding approach (I), the algorithm will be updated instantly and the knowledge level of the user will be reflected. In approach (II), a proper quiz will be retrieved from the system database and will be displayed to the user. For example, a user is unsure concerning “if statement”, so a sample quiz regarding “if statement” will
be generated. After the quiz is responded by the user, the system extracts the stored answer to that quiz from its database and compares it with the user’s answer. Then, an instant feedback will be shown to the user regarding the correctness of his/her answer. In case of the answers are all correct, the Bayesian network gets updated and Navigation menu, refer to sub-section “navigation support”, is presented again while that specific concept is marked as known. In case of most responses are wrong, the users will be recommended to review the learning material regarding that concept again along with displaying them the correct responses. The Bayesian network also gets updated accordingly. Therefore, the Bayesian Network algorithm gets updated frequently and based on the students’ knowledge, it recommends them some relevant lecturer note or recommends them sub-flowcharts and asks them to complete it by providing them with guidance, workspace and editor, system-chat, instant feedback, and visualization notations. For example, the sub-flowchart generated for Factorial Exercise by the system might be different from user to user and some users automatically will be given more hints to complete the sub-flowcharts and some of them less based on the knowledge level of the user.

B. Knowledge Base

The knowledge base is divided into two different sections. Class lecturer notes in web page form and a repository of sample quizzes and questions along with their key solutions which are stored at the first part while the second section includes a flowchart-based multi agent system. The lecturer notes are presented when the user learns a new concept, while quizzes and questions are represented to define whether the user comprehends a specific concept or not. These learning materials containing lecture notes and quizzes are stored by specific concept in programming which refer to its relevant node in the Bayesian network. This separation strategy not only enables instructors to add, edit, and compose the learning materials independently, but also the student level of knowledge can be estimated by the inference mechanism of Bayesian network algorithm applied in the proposed system. As each single node in Bayesian network is referred to a specific concept, the result of quizzes and the feedbacks taken from lecturer note are used to update the belief of the system. Moreover, in order to make any changes in learning materials, they only need to be re-indexed and not reformulate the whole Bayesian network. The system also can recommend some learning sequences based on the user level of knowledge which is the result of this separation mechanism.

a) Flowchart Development

Before users start interacting with this section, they are required to enter their desired programming problem to the system. Afterwards, the multi-agent system recommends users two different options for flowchart development, toolbar and guidance. At the toolbar option, users will be given an editor, the flowchart template, workspace to complete the sub-flowchart, a sub-flowchart of the entered programming exercise, and a system chat. After dragging and dropping and completing the flowchart, the multi-agent system provides them with brief feedbacks next to each shape in flowchart, as brief feedbacks are classical way of making users think after their failure. At the guidance option, users are provided with template of the flowchart having some shapes or sub-flowcharts placed in the right positions, unlike the first option, users are needed to complete the flowchart using the editor. Users are also not allowed to drag and drop the shape in the wrong position as the system could identify it. Instant feedback for each drag and drop is provided by the system chat. Once the users are done dragging and dropping, the system offers them a full flowchart of the programming exercise extracted from the Internet so as to compare it with developed flowchart by users. During flowchart completion using toolbar and guidance options, if any error occurs, it will be detected and stored by error detection component. Afterwards, it will be passed to the Web Crawler in order to find the relevant, additional information and definitions for improving the database automatically without human intervention for the next users. It is also should be mentioned that worst case scenario also predicted by the multi-agent system. In case no keyword is found in the databases, the system refers each main word and keyword to its corresponding shape and develops the sub-flowchart as well as providing users with a workspace for users to complete the sub-flowchart, online system chat to guide users stepwise, and the flowchart template. It is noteworthy to mention that based on the profile taken and knowledge assessment by Bayesian network for each concept, FITS could surprisingly manipulate the guidance and sub-flowcharts so as to provide users with an intelligent, personalized, and dynamic environment for problem solving improvement. This feature, taking full advantage of Bayesian network with the aid of Markup languages, can be considered as the main novelty of this study as there is no system developed benefiting from such interesting feature to make problem solving activities easy, interesting, and appealing subject to students. Based on the estimation of the probability knowing the knowledge concept, FITS recommends three different flowchart development options. For instance, the guidance given to the user A for a programming problem, i.e. Fibonacci Series, is different from user B based on his/her profile. The general architecture of developed flowchart-based multi agent system is shown in figure 2:
C. User Interface Module

Using the interface component, users are able to interact with FITS which is divided into two sub-sections: input and output. The former one is aimed from users to FITS and to update the Bayesian network according to evidence obtained from users, described in section 3.1; and the latter from FITS to users which is shown in a web browser. The recommendations or guidance provided by FITS are either learning techniques or adaptive guidance, discusses in the next section, or animated flowchart development multi-agent, discussed in flowchart development section. All aforementioned learning materials are displayed for users using output module of FITS.

D. Adaptive Guidance

FITS can provide users with designed options for supporting a specific user individually by using Bayesian network and estimating the knowledge level of the user. Three various sorts of adaptive guidance are offered to users by FITS: navigation menu, prerequisite recommendations and flowchart development for problem solving skills, discusses in section 3.2.1.

a) Navigation Support

In order to navigate through the concepts which are shown by proper light, each one indicating the knowledge level of the users in that specific concept such as red indicates the concept is unknown for the specific user, the navigation menu is used. Therefore, each concept is marked with a color (yellow, already known the concept, red, the concept is unknown and not ready to learn, and green, unknown but ready to learn) showing the knowledge level of the user in that specific concept. It is noteworthy to mention that the Bayesian network compute these lights in an automatic manner from the initial profile, quizzes, and feedbacks taken from users.

b) Prerequisite Recommendations

In the case that users choose their level of understanding after they are exposed to learning materials and guidance, FITS also offers students learning materials to prerequisite concepts of this topic such as link of lecturer notes regarding the parent set of specific concept in the Bayesian network. For example, offering “variable” concept for “assignment” since variable is a parent node of assignment concept in the graph. This method is used to avoid repeating the problem concept and to enable users for visiting the prerequisite concepts in order to make sure they have understood the specific concept. By doing so, the user who believe that a prerequisite concept is comprehended, while it is not, can confirm that prerequisites are understood and they do not influence his/her learning process.
There is a misconception of what programming is at the early stages of learning programming for CS minors. A flowchart-based Bayesian network intelligent tutoring system is proposed aimed at computer programming. A system proposed in this paper has benefited from a novel flowchart-based multi agent system at its knowledge base component which applies an automatic text-to-flowchart conversion approach for involving users in flowchart development with the aim of problem-solving improvement. How Bayesian network is applied in the proposed system for modeling and inference purposes, the architecture of FITS, and its components for computer programming are described in this paper. Presently, the majority of web-based learning environments are static, suffering either from lack of interactivity or lack of adaptivity. Various learning materials including lecturer notes, quizzes, guidance, error, feedbacks, flowchart development, and etc. are provided dynamically by FITS for users. Users are guided by navigation through online learning materials with applying traffic lights. This mechanism used in FITS offers users several recommendations regarding learning materials based on the knowledge level of each specific user while some specific concepts are not comprehended. Moreover, prerequisite knowledge required for understanding the specific concept can be given to users by FITS, when users wish to learn a specific concept without learning its previous concepts. To the best of our knowledge, FITS is the first flowchart-based intelligent tutoring system in computer programming. As already indicated, FITS has benefited from a flowchart-based multi agent system at its knowledge base module which applies a novel approach for improving problem solving skills of CS minors. This multi-agent system aimed at getting users involved in flowchart development with the purpose of improving their problem solving skills in a form of solution designing activities. The potential of applying FITS into different computer programming courses highlights it from other related works in the area of programming. We believe, FITS is a useful intelligent tutoring system for both learners and instructors in computer programming. Our future work is the implementation of our proposed FITS as well as its evaluation by conducting an experimental study using novices.

REFERENCES


