Learning recommendation with formal concept analysis for intelligent tutoring system

Jirapon Muangprathub a,*, Veera Boonjing b, Kosin Chamnongthai c

a Faculty of Science and Industrial Technology, Prince of Songkla University, Surat Thani Campus, Surat Thani, 84000, Thailand
b Department of Computer Engineering, Faculty of Engineering, King Mongkut’s Institute of Technology Ladkrabang, Bangkok, 10520, Thailand
c Department of Electronic and Telecommunication Engineering, Faculty of Engineering, King Mongkut’s University of Technology Thonburi, Bangkok, Thailand

ARTICLE INFO

Keywords:
Computer Science
Learning recommendation
Formal concept analysis
Intelligent tutoring system
Adaptive learning

ABSTRACT

The aim of this research was to develop a learning recommendation component in an intelligent tutoring system (ITS) that dynamically predicts and adapts to a learner’s style. In order to develop a proper ITS, we present an improved knowledge base supporting adaptive learning, which can be achieved by a suitable knowledge construction. This process is illustrated by implementing a web-based online tutor system. In addition, our knowledge structure provides adaptive presentation and personalized learning with the proposed adaptive algorithm, to retrieve content according to individual learner characteristics. To demonstrate the proposed adaptive algorithm, pre-test and post-test were used to evaluate suggestion accuracy of the course in a class for adapting to a learner’s style. In addition, pre- and post-testing were also used with students in a real teaching/learning environment to evaluate the performance of the proposed model. The results show that the proposed system can be used to help students or learners achieve improved learning.

1. Introduction

E-learning systems are important in online education, using information and communication technology to help the teacher and improve the self-study of a learner. Computer-aided instruction (CAI) [1, 2] is one model that applies computers to education to support the self-study of a learner. However, CAI lacks the ability to adapt according to a student’s learning style. Hence, intelligent tutoring systems (ITS) have been developed for adaptive educational systems that employ artificial intelligence (AI) techniques to provide individualized instruction [3, 4].

ITSs are adaptive systems, which use intelligent technologies to personalize learning according to individual student characteristics, such as knowledge of the subject, mood and emotion, and learning style [4]. Many researchers have developed ITSs with various techniques to adapt the style of tutoring to match a student’s learning style and enhance learning, such as Protus [3], Oscar [4], ZOSMAT [5], etc. The traditional ITS architecture involves an expert model, instructional model, learner model, and user interface [6, 7]. To improve the value of ITSs, these models will use a knowledge structure [5, 9] to personalize learning [3, 10, 11]; and adaptive learning [4, 8] to improve the effectiveness of a learner’s experience. Thus, approaches to construct knowledge base and adaptive learning are key components in a successful ITS.

In order to develop a proper ITS, we require a knowledge base supporting adaptive learning, which can be achieved by a suitable knowledge construction, i.e., a knowledge representation with knowledge acquisition to personalize learning according to an individual learner’s characteristics. For this reason, this work provides formal concept analysis (FCA) [12, 13, 14, 15] for building the knowledge structure in ITS to support knowledge acquisition and adaptive learning. FCA is one approach for grouping documents into a hierarchical form that supports browsing. It automatically provides generalization and specialization relationships among the formal concepts of a document structure in a concept lattice [8]. Thus, this work applied FCA to adaptive learning for learners. Moreover, this method provides the content of related courses or group of documents that enable recommendations for a learning style. Adaptive learning consists of adaptive presentation and adaptive navigation [2, 6]. Adaptive presentation [5, 6, 14] refers to the education material while adaptive navigation [6, 16, 17, 18] refers to manipulation of the links and structure. This research focuses on adaptive presentation to give personalized advice on content to the learner by using FCA approach.

* Corresponding author.
E-mail addresses: jirapon.m@psu.ac.th (J. Muangprathub), kosin.cha@kmutt.ac.th (K. Chamnongthai).

https://doi.org/10.1016/j.heliyon.2020.e05227
Received 5 April 2020; Received in revised form 3 July 2020; Accepted 8 October 2020

2405-8440/© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
This article is organized as follows. In Section 2, we describe related works. In Section 3, we present our proposed learning recommendation component for ITS. Section 4 shows an example and discusses how our proposed framework can be beneficial. Finally, Section 5 concludes and envisions future works.

2. Related works

This section provides background and related work on ITS and FCA in subsections 2.1 and 2.2, respectively. Related works are reviewed for adaptive learning in ITS in subsection 2.3. In particular, the current research and proposed methods are discussed.

2.1. Intelligent tutoring systems

An Intelligent Tutoring System (ITS) [4, 5, 6, 7, 19, 20, 21] involves presenting learning materials to a learner as guided by artificial intelligence (AI) techniques, so as to improve the effectiveness of a learner’s experience. Generally, the classical ITS architecture is considered to have four major components: expert model, instructional model, learner model, and user interface as shown in Fig. 1. In Fig. 1, the first model is expert knowledge. The expert knowledge model [5, 6, 7] comprises the facts and rules of the particular domain to be conveyed to the learner, i.e. the knowledge of the experts. Knowledge elicitation and codification can be a very time-consuming task, especially for a complex domain with an enormous amount of knowledge and interrelationships of that knowledge. Thus, investigating how to encode knowledge and how to represent it in an ITS remains the central issue of creating an expert knowledge model. Also, in its function as a standard, the expert knowledge model can be used to assess the student’s overall progress. To achieve this requires establishing some criteria for comparing levels of knowledge. This type of comparison is possible only if the knowledge has been explicitly represented. Hence, ITSs differ considerably from traditional CAI programs in that the knowledge in the latter is implicitly represented within its code. Instructional model [5, 21, 22] refers to the teaching strategy or the pedagogic model. This model contains knowledge for making decisions about instructional tactics. It relies on the diagnostic processes of the learner model for making decisions about what information to present, and when and how, to a learner.

Learner model [5, 6, 23] is often viewed as a subset of the expert knowledge model, which changes over the course of tutoring. ITS is able to adapt to a specific student during the teaching process. The learner model can be evaluated in regard to several criteria of usefulness. These criteria include the following: (1) Learner model fit to data refers to how well the learner model can be used to simulate the quantitative and/or qualitative patterns of learning in real students [1], (2) Ease of understanding, (3) Generality/flexibility, (4) Cost of creation, (5) Multiple choices to pedagogical decisions at any level, (6) Time scale refers to the overall longevity of the learner model, and (7) Learning gains in practice. Finally, user interface module is the communicating component of the ITS, which controls interaction between the learner and the system. In addition to the conventional human-computer interface features, some recent systems have had natural language interaction [24, 25], speech recognition [26, 27], and the sensing of student emotions [27, 28].

Adaptive hypermedia [6, 7, 29] and ITS are used in E-learning systems to drive the connection and to personalize instruction on the basis of adaptation to the learner’s learning style. An ITS needs to be designed and implemented to support modification of lecture content, the decision rules and the fact base of the expert model, and the methods to measure performance of learning. There are two forms of adaptive learning, i.e. adaptive presentation and adaptive navigation, which are mentioned in subsection 2.3.

ITSs are adaptive systems that use intelligent technology to personalize learning according to an individual learner’s characteristics [3, 10, 11]. There are three main approaches to intelligent tutoring: (1) Curriculum sequencing that aims to improve the learning experience, (2) intelligent solution analysis, focusing on giving students detailed feedback on incomplete or erroneous solutions, helping them learn from their mistakes, and (3) Problem solving support, involving a constructivist approach, for students constructing their own knowledge, and thus encouraging a deeper understand of the topics. Furthermore, various ITSs were built for interactive learning with the learners surveyed [29, 30]. Table 1 summarizes reviewed prior studies of interest. It shows that mostly the previous works have applied machine learning with an expert model based on rule-based reasoning. However, the studies are limited by the fixed questions and content to suggest to users [37, 38, 39]. Fixed exercises are provided to learner with focus on learner model [32, 34, 35]. Interactive systems can be quite time consuming to develop [33]. The rules provided for application are not dynamic. These limitations are eliminated by using a dynamic knowledge base or an amendable knowledge base as provided by FCA. In this research, we focus on adaptive presentation as the proposed improved alternative type of knowledge base, by using FCA for ITS. We use intelligent solution analysis approach to improve self-study by students, evaluated by assessing changes from pre-test before study to next test time.

2.2. Formal concept analysis (FCA)

Formal concept analysis (FCA) [12, 13, 14, 15, 40], is a mathematical approach to data analysis based on lattice theory. It is widely used in information science [13] to describe attributes of information represented in a hierarchical model. FCA is not only a method for data analysis and knowledge representation, but also a formal formulation for concept formation and learning. It provides both generalization and specialization among concepts through the concept lattice [14, 15]. Practically, FCA starts with a formal context, which contains values 0 or 1 in an information system. Below, we introduce basic definitions and ideas of FCA following reference [12].

Definition 1. A formal context K := (G, M, J) consists of two sets G and M and a relation I between G and M. The elements of G are called the objects and the elements of M are called the attributes of the context. In order to express that an object g is in a relation I with an attribute m, we write glm or (g,m) ∈ I and read it as “the object g has the attribute m”.

Another form of formal context is called a cross table. It can be used to identify groups of cases with shared attributes in binary relation.
format. In knowledge representation of this study, the objects represent each chapter of each class, and are classified by the topics in a course. Attributes are based on sets of keywords from all the content in the class. This study introduces the definition of an association between object and its attributes as follows.

**Definition 2.** For a set \( A \subseteq G \) of objects, \( A' \) is defined as follows

\[ A' := m \in M | g I m \text{ for all } g \in A. \]

Correspondingly, for a set \( B \subseteq M \) of attributes, \( B' \) is defined as follows

\[ B' := g \in G | g I m \text{ for all } m \in B. \]

**Definition 3.** A formal concept of the formal context \((G, M, I)\) is a pair \((A, B)\) with \( A \subseteq G, B \subseteq M, A' = B \) and \( B' = A \). We call \( A \) the extent and \( B \) the intent of the formal concept \((A, B)\). \((G, M, I)\) denotes the set of all formal concepts of the formal context \((G, M, I)\).

**Definition 4.** If \((A_1, B_1)\) and \((A_2, B_2)\) are formal concepts of formal context \(K:=(G, M, I)\), \((A_1, B_1)\) is called a subconcept of \((A_2, B_2)\) or \((A_2, B_2)\) is a superconcept of \((A_1, B_1)\)), provided that \(A_1 \subseteq A_2\) (or \(B_2 \subseteq A_2\)) and

\[ K' := (G, M, I'), \text{ where } I' = (G, M, I), \text{ and } (A_1, B_1) \text{ is a subconcept of } (A_2, B_2). \]
and is denoted by \((A_1, B_1) \leq (A_2, B_2)\). The relation \(\leq\) is called the hierarchical order (or simply order) of the formal concepts. The set of all formal concepts of \((G, M, I)\) partially ordered in this way is denoted by \(B(G, M, I)\) and is called the concept lattice of the formal context \((G, M, I)\).

To extract knowledge for advising a learner, the implications between keywords from content are used to identify chapters that the learner should study after taking a post-test. The implication provides a group of chapters identified by keywords. Mostly, the previous studies with FCA applied to ITS used it for adaptive navigation in an e-learning system \([2, 6, 7]\). The authors applied FCA as a knowledge structure to webpages and their keywords \([20, 22]\). These keywords relate to learner search domains. Next, these search keywords are checked in formal concepts for match with the knowledge structure, to receive webpage suggestions for the learner. Similarly, in \([17]\), the authors demonstrated FCA to advice on topics in a Geometry course. In \([18]\), FCA is applied to obtain association rules among keywords in webpages, for Java language materials, to provide suitable content to learners.

The advantage of FCA in those prior studies in knowledge construction was based on it providing related keywords that can identify sets of data for adaptive navigation in an e-learning system. However, the adaptive presentation in the e-learning system should display educational materials according to a student’s weaknesses, as indicated by their responses to questions based on earlier educational material. Thus, this study employs a concept lattice in FCA as a knowledge structure for adaptive presentation in ITS.

2.3. Adaptive learning in ITS

Adaptive learning \([2, 6, 8, 41, 42, 43]\) is an educational method which uses computer as interactive teaching device. There are two forms, namely adaptive presentation and adaptive navigation. Adaptive presentation \([5, 6, 14]\) refers to the education material display according to a student’s weaknesses as indicated by their responses to questions based on earlier educational material. In other words, adaptive navigation \([6, 16, 17, 18]\) refers to the manipulation of links and structure. ITS can monitor the performance of a learner and personalize instruction on the basis of adaptation to the learner’s learning style, current knowledge level, and appropriate teaching strategies in an E-learning system. Personalized learning \([3, 8]\) designs educational experiences that fit the need, goals, talents, and interests of the learner. Researchers have recently begun to investigate various techniques to help teachers in E-learning systems to assess learning styles \([4, 43]\). Moreover, to adapt presentation and navigation according to learner, algorithms suggesting further learning can be found in prior literature \([8, 14, 17, 29]\).

In this research, we provide an improved knowledge base by using FCA to support adaptive presentation and personalized learning. This knowledge structure requires an adaptive algorithm to retrieve content according to an individual learner’s characteristics. Thus, we also propose an algorithm to suggest courses and classes to each learner.

3. Research methodology

3.1. System architecture and design

In this section, we present our proposed ITS system in Fig. 2. Generally, the classical ITS architecture is considered to have four major components: expert model, instructional model, learner model, and user interface, which are described as follows.

The expert model comprises the facts and rules of the particular domain to be conveyed to the learner, i.e. the knowledge of the experts. Referring to Fig. 2, after teacher inputs the details of a course, that course will be generated using FCA to represent the knowledge. This knowledge is about the relationship between learner profiles and details of course to guide content for the learner. To extract knowledge for advising a learner, the implications between keywords from content are used to identify chapters that learners should study after they take a post-test. The implication provides a group of chapters identified by keywords.

Instructional model is referred to for the teaching strategy or the pedagogic model. This model contains knowledge for making decisions about instructional tactics. It relies on the diagnostic processes of the learner model for making decisions about what, when and how to present information to a learner. Instructional model in Fig. 2 is the education module. This process derived from the knowledge base shows content for recommended learning. After the learners study, they will test themselves. The score will identify the level of learning and is stored in their profile for future.

From Fig. 2, the learner profile will be recorded for use in adaptive learning according to course selection. The data in this learner profile is related to the knowledge base by using FCA to suggest content to learner in the next learning step. In this work, we created user interfaces for learner and teacher with natural language interaction. We will show an example in section 4.

3.2. Learning recommendation with adaptive algorithm

We apply FCA to collaborative recommendations for each learner. Fig. 3 illustrates an overview of knowledge acquisition and adaptive learning for learners. Firstly, FCA is applied to achieve a suitable knowledge construction. The course details that include contents, objectives, exercises, and tests, will be prepared and transformed into independent hierarchical structure with keyword acquisition.
The following example illustrates knowledge based acquisition of the expert model in our ITS system. Suppose the initial course details are as shown in Table 2. This formal context form is transformed into formal concept and concept lattice according to the Definitions 2–4 as shown in Fig. 4. This figure represents knowledge base structure that shows the co-appearances and subconcept–superconcept relations in a concept lattice. This structure can provide a learning path to a learner. In Fig. 4, an example of a learning path is as follows:

1. k3 → k1
2. k4 → obj1
3. obj3 and k5 → k1 and k
4. k1 and k5 → obj3 and k3
5. k2 and k5 → obj2

From the above implication rules, the first and second rules imply that if the learners study topic k3 then they should study topic k1, and that if the learners study topic k4 then they will get to the first objective, respectively. These rules are used to recommend what to study to each learner.

Afterwards, the set of keywords is reformed into knowledge base structure by Algorithms 1 and 2 [8, 44, 45]. However, these algorithms in the previous work were proposed for a general case of FCA construction. Thus, this work proposes adapted algorithms for supporting ITS model. Next, when the learners study lessons and work through exercises and tests, they make mistakes in some questions, which informs about missed objectives. The keywords for mistaken exercises or tests will be matched with implication rules from knowledge base to identify relevant chapters or lessons. Consequently, our system will interact with the learners and recommend lessons to study again, and practice exercises and tests.

Algorithm 1 consists of two parts (i.e., extent and intent), where input data is a formal context (G, M, I). G represents keywords and M represents objectives of each lesson. The first part, FindExtentX(\(i\)) is a function to generate a family of extents, while the rest is used to find the intent of the formal concept. The result is a formal concept that identifies groups of keywords sharing common objectives of each lesson. To extract knowledge for advising the learner, the implications between keywords from content are used to identify chapters that learners should study after they take a test. The implications provide groups of keywords identified by objectives. Thus, Algorithm 2 is provided to build a concept lattice to prepare related knowledge of content to the learner. Algorithm 3 finds implication rules to recommend to learners.

**Table 2.** Document collection in formal context form.

| obj1 | obj2 | k1 | k2 | k3 | k4 | k5 |
|------|------|----|----|----|----|----|
| Q1   | 0    | 0  | 1  | 0  | 1  | 0  |
| Q2   | 1    | 0  | 0  | 0  | 1  | 0  |
| Q3   | 1    | 1  | 1  | 0  | 1  | 0  |
| Q4   | 1    | 0  | 1  | 0  | 1  | 0  |
| Q5   | 0    | 1  | 0  | 0  | 1  | 0  |
| Ex1  | 0    | 0  | 1  | 0  | 0  | 1  |
| Ex2  | 1    | 0  | 1  | 1  | 0  | 0  |
| Ex3  | 0    | 1  | 0  | 0  | 1  | 0  |

**Fig. 4.** Concept lattices generated from Table 2.

**Algorithm 1** Algorithm for constructing the set of formal concepts.

**Input** : The formal context (G, M, I).

**Output** : Set of the formal concepts \(W(G, M, I)\)

**Method** :

1. \(F_X = \text{FindExtentX}(i);\)
2. \(F_Y = R;\)
3. For \(i = 0\) to \(|M|\)
4. \{ //find intersection from extent \(X[i]\)
5. \(F_Y = F_Y \cup F_X[i];\)
6. \} //Return formal concept \((X, Y)\)

**function** FindExtentX(\(i\)) // input is formal context (G, M, I),

1. For \(i = 0\) to \(|M|\)
2. \{ 3. \(F_X[i] = \text{m}^\prime;\)
4. \(F_X = F_X[i];\)
5. \}
6. For \(j = 0\) to \(|F_X[i]|\)
7. For \(k = 0\) to \(|F_X[i]|\)
8. \{ 9. \(\text{IntersFx} = F_X[i][j] \cap F_X[i][j];\)
10. If \(\text{IntersFx} \neq \emptyset\)
11. \(F_X = F_X[i] \cup \text{IntersFx};\)
12. \} // Return a family of extent \(F_X\)
Algorithm 2 Algorithm for construction of concept lattice.

Input: Set of the formal concepts \( \mathfrak{B}(G, M, I) \).
Output: Concept lattice \( \mathfrak{B}(G, M, I) \).

Method:
1. \( Bottom = (G, \emptyset) \);
2. \( currentLevel = \emptyset \);
3. \( BufLevel = \text{FindSetExtLength}(Bottom) \) ;
   // SetLength is sorted by ascending
4. If \( BufLevel.\text{SetLength} = 1 \) then
5. \( \emptyset \rightarrow \text{Add link into Bottom} \);
6. \( currentLevel = BufLevel.\text{GroupSameLength}[0] ;\)
   // The first group level is 1
7. For \( i = 1 \) to \( n \)
   // \( n = \text{size of SetLength} = BufLevel.\text{SetLength} \)
8. For \( j = 0 \) to \( m \) // \( m = \text{size of currentLevel} \)
9. For \( k = 0 \) to \( p \)
   // \( p = \text{size of group level is} \)
   // \( i = \text{size of BufLevel.\text{GroupSameLength}[j]} \)
10. If \( X_i \in X_j \neq \emptyset \) then
11. \( \emptyset \rightarrow \text{Add link into Bottom} ;\)
   // Return concept lattice \( \emptyset \)

function \( \text{FindSetExtLength}() \) ;
// input is formal context \( (G, M, I) \),
1. \( \text{SetLength} = \emptyset ;\)
2. \( \text{GroupSameLength} = \emptyset ;\)
3. For \( i = 1 \) to \( |F| \)
   // Find length of extent in formal concept \( \text{SetLength} \)
5. If \( |X_i| \neq \text{SetLength} \)
6. \{ \)
   // add new length of extent \( X_i \) and \( X_j \)
   // into same length such that
7. \( |X_i| = \text{SetLength} ;\)
8. \( X_j = \text{GroupSameLength} ;\)
9. \} ;
10. \} // Return \( \text{SetLength} \) and \( \text{GroupSameLength} \)

4. Implementation

4.1. System overview

The aim of this research was to develop learning recommendations in ITS that dynamically predict and adapt to a learner's style, by using smartphone and web applications. This work implemented a C programming language tutor for general C programming teachers and learners, called Tutor C Programming. The proposed system is shown as an overview in Fig. 5. This figure shows an example of application. The application is divided into two parts, for front-end user and for back-end user. The front-end user is served to learner and teacher (or admin) based on web application, while the back-end is a hidden module using FCA for expert model in ITS, mentioned in section 4.2. The system is designed for three user groups, namely learners, teachers, and administrators, with interface structure designs in Fig. 5.

Referring to Fig. 5, the user can register to the system. Next, the learner user can manage his/her profile to update data and study lessons by themselves, with content provided by teacher. They can take the exercises and tests, after which they are assessed to generate recommendations by system. Moreover, they can view the study history and results. The teacher can add content to his/her course. The proposed system supports many classes. The teacher can manage a lesson, the objectives of a lesson, exercises, and tests to the students. Moreover, the teacher can view the scores to assess improvements of his/her students. The final user type, administrators, can manage all information in the system.

The web-based application is implemented to manage information of learners, teachers, and courses to recommend to learner, shown in Fig. 6. This shows an example web page to illustrate information management for learners, teachers, and administrators (more detail can be viewed on the web: https://www.parasystem.org/LearnC/pages/login.php). The learner can study provided content or can take an examination to get learning recommendations that are described in the next subsection.

4.2. System with recommendations implemented

This section presents the implemented learning recommendation for learners. The learner can study from provided content or can take an examination to get the learning recommendations. The examination provided is one of two types, i.e., multiple choice or programming test, shown in Fig. 7. In Fig. 7 (b), the learner will receive help from the system that recommends further study after an exercise or a test, in hint format. The hint is presented in three steps to support and to remind the learner.

Next, the learner gets the total score from both types of examinations, shown in Fig. 8(a). Afterwards, if the learner clicks to get learning recommendations, they can obtain the content that they need to improve with. The instructional model in Fig. 8(b) provides content for learner from teacher (or admin). They can manage the information of learner, course and examination to record, for adapting to a learner's style with use of FCA as in section 3.2. The learner score report is provided to teacher for monitoring learning improvement. This developed system is evaluated by testing if it improves a learner's learning performance.

5. Experimental results and discussion

The experimental process is demonstrated in Fig. 9. Initially, the participants were 70 students derived from 2 groups that enrolled in Principles of Programming Course. The first learner group (35 students) is using ITS (experimental group) and the second learner group (35 students) is not using ITS (control group). This first sample group is
Fig. 5. The interface structure design for the proposed system.

(a) The main page for login to system          (b) The page for user registration

(c) The page for administrator to manage user  (d) The page for teacher accepting student registration

(e) The student can select courses of interest  (f) The page for learner to view her/his score

Fig. 6. An example web page for information management.

(a) The page for multiple choice examination  (b) The page for programming in a test

Fig. 7. An example examination page for learner.
taken from the first-year students in Information Technology Program, while the second sample group is taken from the third-year students of Industrial Management Technology at the Prince of Songkla University, Thailand. The experimental group was studying using the proposed ITS system via the developed web application, while the control group was studying in the traditional lecture approach. Next, a pretest was administered to provide baseline for improvements after learning. Afterwards, a posttest was used to evaluate improvements by comparison with pretest. In this experiment, we used three chapters of programming to be tested.

The empirical study was performed to evaluate improvements in learning performance, by testing the following hypotheses:

1. The score of posttest is better than of pretest both when using our system and when not using the system. This hypothesis would show that the learners experience positive learning performance.
2. The scores of pretest and posttest when using our system are different from those without the system. The hypothesis is that using our system improves learning performance.

To test the first hypothesis, the pretest and posttest approach is used to count the number of correct answers given to tutorial questions during tutoring and then comparison between pretest score and posttest score. This work applied paired t-tests to pretest and posttest scores, for each chapter studied, as shown in Table 3. The results show that the posttest is better than pretest, with significant difference for each chapter.

To test the second hypothesis, it was necessary to compare the learning of experimental group with that of control group. A common measure of learning is learning gain. Learning gain could be measured in a number of ways, for example the number of tutorial questions a learner answers correctly or an improvement in test scores. Learning gain was measured by using a pre-test and post-test, where the same test for three chapters was taken before and after the tutoring. Test scores were then compared to establish whether there was any improvement.

learning gain = post-test score – pre-test score

We use the independent samples test to assess learning gain for each chapter, shown in Table 4. The results show that using our system improved learning performance from that achieved without ITS, with a significant difference for each chapter.

However, the above experiment can explicitly test only the application of ITS using FCA. The proposed knowledge base with lattice structure provided a dynamic knowledge base or an amendable knowledge base supporting adaptive learning to the users. Moreover, this structure can utilize both explicit and implicit knowledge. The explicit knowledge shows in the hierarchical structure while the implicit knowledge is discovery knowledge. These advantages should be tested in future studies.

6. Conclusion

ITSs are adaptive systems that use intelligent technologies to personalize learning according to individual student characteristics. This paper applied FCA as an intelligent method based on rule-based reasoning that dynamically predicted and adapted to a learner’s style to recommend topics to the learner. The FCA was used to improve the knowledge base supporting adaptive learning, which can be achieved by suitable knowledge construction. In addition, our knowledge structure provides adaptive presentation and personalized learning with the proposed adaptive algorithm to retrieve content according to individual learner characteristics. This work developed learning recommendations in ITS that dynamically predict and adapt to a learner’s style by using smartphone and web applications. It implemented a C programming language tutor for general C programming teachers and learners, called Tutor C Programming, based on the web application. To demonstrate the proposed adaptive algorithm, pretest and posttest were used to evaluate suggestion accuracy of course and class for adapting to a learner’s style. Moreover, this was also used with students in a real teaching/learning environment to evaluate the performance of the proposed teaching approach. The results show that the proposed system can be used to help students or learners to improve their learning.

Declarations

Author contribution statement

J. Muangprathub: Conceived and designed the analysis; Collected the data; Contributed data or analysis tools; Performed the analysis; Wrote the paper. V. Boonjing, K. Chamnongthai: Conceived and designed the experiments; Contributed data or analysis tools; Performed the analysis; Wrote the paper.
Table 3. The paired sample test results.

| Group          | Methods | Mean   | SD     | t-test | df   | p-value  |
|----------------|---------|--------|--------|--------|------|----------|
| Using ITS      | \(p_{\text{HCI}} = p_{\text{HCI}}\) | 10.629 | 1.087  | 57.848 | 34   | <.001*** |
| Non-using ITS  | \(p_{\text{HCI}} = p_{\text{HCI}}\) | 9.629  | 1.784  | 31.936 | 34   | <.001*** |
|                | \(p_{\text{HCI}} = p_{\text{HCI}}\) | 12.114 | 2.529  | 28.333 | 34   | <.001*** |
|                | \(p_{\text{HCI}} = p_{\text{CHI}}\) | 4.543  | 1.578  | 17.029 | 34   | <.001*** |
|                | \(p_{\text{HCI}} = p_{\text{CHI}}\) | 4.171  | 1.636  | 15.087 | 34   | <.001*** |

* \(p<0.05\), ** \(p<0.01\), *** \(p<0.001\).

Table 4. The independent samples test results.

| Lesson | Mean Non-using ITS | Mean Using ITS | SD Non-using ITS | SD Using ITS | t-test | df | p-value |
|--------|--------------------|----------------|------------------|--------------|--------|----|---------|
| CH1    | 10.63              | 5.91           | 1.087            | 2.092        | 11.832 | 51.12| <.001***|
| CH2    | 9.63               | 4.54           | 1.784            | 1.578        | 12.633 | 68  | <.001***|
| CH3    | 12.11              | 4.17           | 2.529            | 1.636        | 15.600 | 68  | <.001***|

CH1 = chapter 1, CH2 = chapter 2, CH3 = chapter 3.

Funding statement

This work was supported by the Thailand Research Fund (TRF) and Prince of Songkla University (PSU), under grant No. TRGS5880206 and SIT/5811255.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

Acknowledgements

The authors are deeply grateful the Thailand Research Fund (TRF) and Prince of Songkla University (PSU) grant, Thailand. This research was financially supported by the TRF and PSU, under grant No. TRGS5880206 and SIT/5811255. The authors also gratefully acknowledge the helpful check of English language by Assoc. Prof. Dr. Seppo Karrila.

References

[1] M. Goyal, D. Yadav, A. Choubey, E-learning: current state of art and future prospects, Int. J. Comput. Sci. Issues 9 (3) (2012) 409–499.
[2] Ma del Puerto Paul Ruiz, Ma Jestín Fernández Díaz, F. Ortiz, R.P. Juan, Adaptation in current e-learning systems, Comput. Stand. Interfaces 30 (2008) 62–70.
[3] A. Klajnić-Miličević, B. Vesin, Mirjana Ivanović, Z. Budimac, E-learning personalization based on hybrid recommendation strategy and learning style identification, Comput. Educ. 56 (2011) 885–899.
[4] A. Latham, K. Crockett, D. McLean, An adaptation algorithm for an intelligent natural language tutoring system, Comput. Educ. 71 (2014) 97–110.
[5] A. Keles, R. Oacak, A. Keles, A. Gulpil, Zoumat: Web-based intelligent tutoring system for teaching-learning process, Expert Syst. Appl. 36 (2009) 1229–1239.
[6] N. Kara, N. Sevim, Adaptive learning systems: beyond teaching machines, Contemp. Educ. Technol. 4 (2) (2013) 108–120.
[7] R. Koo, C.K. Hsu, M. Chang, J. Heh, A personalized webeapage reconstructor based on concept lattice and association rules, J. Internet Technol. 12 (2011) 1015–1024.
[8] J. Muangprathub, V. Boonjing, P. Pattarainatkorn, A new case-based classification using incremental concept lattice knowledge, Data Knowl. Eng. 83 (2013) 39–53.
[9] M. Nussbaum, R. Rosas, F. Feiran, F. Cardemas, Development of intelligent tutoring systems using knowledge structures, Comput. Educ. 36 (2001) 15–22.
[10] D. Litman, Speech and Language Processing for Adaptive Training, Cambridge University Press, 2013.
[11] T. Taran, S. Sirota, Knowledge learning technology for intelligent tutoring systems, Int. J. Inf. Theories Appl. 10 (2007) 153–158.
[12] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundation, Springer, Heidelberg, 1999.
[13] W.L. Johnson, A. Valente, Tactical language and culture training systems: using artificial intelligence to teach foreign languages and cultures, in: Proceedings of the 20th National Conference on Innovative Applications of Artificial Intelligence, 2008, pp. 1632–1639.
[14] J.R. van Seters, M.A. Ousevoort, J. Trampler, M. Goedhart, The influence of student characteristics on the use of adaptive e-learning material, Comput. Educ. 58 (2012) 942–952.
[15] S.K. D’Mello, A. Graesser, B. King, Toward spoken human-computer tutorial dialogues, Hum. Comput. Interact. 25 (2010) 289–323.
[16] Y. Akbulut, C.S. Cardak, Adaptive educational hypermedia accommodating learning styles: a content analysis of publications from 2000 to 2011, Comput. Educ. 58 (2008) 835–842.
[17] G.J. Hwang, A conceptual map model for developing intelligent tutoring systems, Comput. Educ. 40 (2003) 217–235.
[18] R. Wille, Formal concept analysis as mathematical theory of concepts and concept hierarchies, in: Lecture Notes Artificial Intelligent (LNAI-3626), 2010, pp. 1–35.
[19] H. Mohamed, T. Bensebaa, P. Trigano, Developing adaptive intelligent tutoring system based on item response theory and metrics, Int. J. Adv. Sci. Technol. 43 (2013) 1–14.
[20] U. Priss, Formal concept analysis in information science, Annu. Rev. Inf. Sci. Technol. 40 (1) (2006) 521–543.
[21] G. Zha, Z. Zhang, Knowledge unit discovery for programming tutoring based on formal concept analysis, in: 2010 International Conference on Educational and Information Technology (ICCIET 2010), 2010, pp. 473–479.
[22] H.M. Ghadiri, M. Rastgourpour, A web-based adaptive and intelligent tutor by expert systems, in: Advances in Computing and Information Technology, 2013, pp. 87–95.
[23] K. Kularbphetong, An adaptive web-based intelligent tutoring using mastery learning and logistic regression techniques, J. Theor. Appl. Inf. Technol. 66 (2014) 31–35.
[24] A.C. Graesser, S.K. D’Mello, X. Hu, Z. Cai, A. Oney, B. Morgan, Auto tutor, in: Applied Natural Language Processing: Identification, Investigation, and Resolution, 2012, pp. 169–187.
[25] X. Daomin, Multi-agent based e-learning intelligent tutoring system for supporting adaptive learning, in: 2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications, 2013, pp. 393–397.
[26] G. Beydoun, Formal concept analysis for an e-learning semantic web, Expert Syst. Appl. 36 (6) (2009) 10952–10961.
[27] S.Y. Cheng, C.S. Lin, H. Chen, J. Heh, Learning and diagnosis of individual and class conceptual perspectives: an intelligent systems approach using clustering techniques, Comput. Educ. 44 (2005) 257–283.
[28] C. Chen, Intelligent web-based learning system with personalized learning path guidance, Comput. Educ. 51 (2008) 787–814.
[29] Shaqun Yang, S. Ding, Zhiqiang Yao, The theory about cd-cat based on fca and its application, Int. J. Distance Edu. Technol. 7 (4) (2009) 61–62.
[30] P. Dalić, J. Dalić, B. Cvenković, V. Serifi, A review of intelligent tutoring systems in e-learning, Ann. Univ. Goerica 2016 (3) (2016) 85–89.
[31] J.M. Harley, F. Bouchet, M.S. Hussain, R. Azevedo, R. Calvo, A multi-componmental analysis of emotions during complex learning with an intelligent multi-agent system, Comput. Hum. Behav. 48 (2015) 615–626.
[32] B. Grauwemeijer, M. Mavrikis, W. Holmes, S. Gutierrez-Santos, M. Wiedmann, N. Rummel, Affecting off-task behaviour: how affect-aware feedback can improve student learning, Paper presented at the ACM International Conference Proceeding Series, 2016.
[33] F. Grivokostopoulou, I. Periokis, I. Hatzilygeroudis, An educational system for learning search algorithms and automatically assessing student performance, Int. J. Artif. Intell. Educ. 27 (1) (2017) 207–240.
[34] U. Kose, A. Arslan, Optimization of self-learning in computer engineering courses: an intelligent software system supported by artificial neural network and vortex optimization algorithm, Comput. Appl. Eng. Educ. 25 (1) (2017) 142–156.

[35] G.A. Khachatryan, A.V. Romashov, A.R. Khachatryan, S.J. Gaudino, J.M. Khachatryan, K.R. Guarian, N.V. Yufa, Reasoning mind genie 2: an intelligent tutoring system as a vehicle for international transfer of instructional methods in mathematics, Int. J. Artif. Intell. Educ. 24 (2) (2014) 162–188.

[36] D. Weragama, J. Reye, Analysing student programs in the php intelligent tutoring system, Int. J. Artif. Intell. Educ. 24 (2) (2014) 162–188.

[37] L.K. Soh, N. Khandaker, H. Jiang, I-minds: a multiagent system for intelligent computer-supported collaborative learning and classroom management, Int. J. Artif. Intell. Educ. 18 (2) (2008) 119–151.

[38] A. Latham, K. Crockett, D. McLean, B. Edmonds, A conversational intelligent tutoring system to automatically predict learning styles, Comput. Educ. 59 (1) (2012) 95–109.

[39] D. Wang, H. Han, Z. Zhan, J. Xu, Q. Liu, G. Ren, A problem solving oriented intelligent tutoring system to improve students’ acquisition of basic computer skills, Comput. Educ. 81 (2015) 102–112.

[40] B. Ganter, R. Wille, Applied Lattice Theory: Formal Concept Analysis, Institute for Algebra, 1997.

[41] J. Dabolini, Trends of the usage of adaptive learning in intelligent tutoring systems, in: Databases and Information Systems (Baltic DB&IS 2012): the 10th International Baltic Conference, 2012, pp. 191–196.

[42] M.C. Desmarais, R.S.J. Baker, A review of recent advances in learner and skill modeling in intelligent learning environments, User Model. User-Adapt. Interact. 22 (2012) 9–38.

[43] P. Phobun, J. Vicheanpanya, Adaptive intelligent tutoring systems for e-learning systems, Proc., Soc. Behav. Sci. 2 (2010) 4064–4069.

[44] C. Kumar, P. Singh, Knowledge representation using formal concept analysis: a study on concept generation, in: Global Trends in Intelligent Computing Research and Development, 2014, pp. 306–336.

[45] P.K. Singh, C. Kumar, A. Gani, A comprehensive survey on formal concept analysis, its research trends and applications, Int. J. Appl. Math. Comput. Sci. 26 (2) (2016) 495–516.