

Cloud-based Context-aware Mobile Intelligent Tutoring System of Technical Computer Skills

<https://doi.org/10.3991/ijim.v11i4.6852>

Hanan Elazhary

King Abdulaziz University, Jeddah, Saudi Arabia;

Electronics Research Institute, Cairo, Egypt

helazhary@kau.edu.sa; hananelazhary@eri.sci.eg

Abstract—In spite of the ultimate importance of procedural technical skills for most disciplines, there is shortage in the number of corresponding Intelligent Tutoring Systems (ITSs). This paper proposes a cloud-based context-aware mobile ITS of Technical computer Skills (TS-ITS). TS-ITS is designed as a mobile application to be a valuable learning tool that can be ubiquitously used. Due to the nature of technical skills, they typically require special types of ITSs including, for example resource-intensive demonstration videos. Accordingly, TS-ITS is designed based on mobile cloud computing to benefit from the cloud resources. This also facilitates serving and maintaining progress information of an enormous number of students. TS-ITS is context-aware and can provide a student with a set of relevant questions and possibly demonstration videos according to the assessed knowledge level. Scaffolding is also employed, where an error in a question corresponding to a skill can trigger videos of prerequisite ones since skills are sorted topologically. This is integrated with self-paced learning capability, where a student can efficiently select a topic or skill(s) to learn or search via keywords for a limited-scope demonstration video according to the skill the student is trying to acquire or the task the student is trying to accomplish. TS-ITS can be adapted to other domains and extended to include advanced types of training material as they become available powered by the theoretically infinite cloud resources. Empirical evaluation of a prototype implementation of TS-ITS shows its tutoring effectiveness.

Keywords— computer skills, context awareness, intelligent tutoring systems, mobile applications, mobile cloud computing, scaffolding, self-paced learning, technical skills, topological sorting

1 Introduction

Intelligent tutoring systems (ITSs) are intended to tutor the students and improve their skills [1]. ITSs are developed for many reasons such as the absence or scarceness of human tutors in rural areas, to help students learn at their own pace or to provide a supplementary learning tool to students. Intelligent tutoring systems can provide different and various forms of *intelligent* help for an effective learning process. For example, they can generate problems and examine the student answers to spot errors and

try to figure out their causes to provide relevant feedback and help. Context-aware ITSs adapt the problem generation and the provided help to each individual student based on the assessed level of knowledge in order to address weakness areas in the respective domain [2]. ITSs have been developed over the year for many disciplines. In biology, for example a system has been developed to tutor the students to solve problems regarding genetics concepts [3]. In mathematics, for example a system has been developed to tutor students to solve algebra problems [4]. Solving problems in language learning has also been considered. For example, a system has been developed to tutor students to extract the roots of Arabic words [5]. The computer domain has also gained considerable attention. For example, systems have been developed to tutor students to solve problems regarding the SQL database language [6], object-oriented analysis and design [7] and programming skills [8].

Cognitive knowledge can be broadly classified into declarative knowledge and procedural knowledge [9]. Declarative knowledge is knowledge about facts and concepts such as general knowledge in biology and geography. In Bloom's cognitive taxonomy, declarative knowledge is explicitly classified into factual and conceptual knowledge. Procedural knowledge, on the other hand is knowledge of how to do things such as how to solve problems in algebra. In Bloom's revised cognitive taxonomy, a more advanced form of knowledge is metacognitive knowledge or self-knowledge of cognitive tasks and when each should be applied [10]. It is obvious that most ITSs in the literature are mainly concerned with procedural knowledge, because it is more challenging in comparison to declarative knowledge.

However, a very important type of procedural knowledge, that has not gained needed attention is procedural technical skills needed to accomplish tasks that are not textual in nature. A research study has shown that many technical skills are portable across multiple occupations [11] indicating that such tutoring systems are of ultimate importance for a wide-range of disciplines. Another research study has shown that acquiring such skills is challenging for technical institute graduates due to many reasons such as the scarcity of learning materials and the relatively large class sizes [12]. Those findings motivate exerting more effort to develop ITSs for such skills.

Two possible approaches for realizing such ITSs are developing immersive Virtual Reality (VR) environments and interactive ITSs and developing simpler questions such as short-answer, multiple-choice and true and false questions. Immersive VR learning environments have been proposed in the literature for ITSs [13], but after about ten years, some authors are still suggesting frameworks for such ITSs [14] due to their complexity. Such environments have been developed for very few applications such as foreign language learning [15] and cellular biology concepts teaching [16]. On the other hand, relatively few simple questions-based ITSs have been developed. For example, Hospers et al. [17] proposed an agent-based ITS for tutoring nurses technical skills such as the angle at which a syringe should be injected in a given situation based on short-answer questions. The system included merely simple questions, but the authors proposed extending the system with a VR environment.

Mobile applications have the advantage of availability anywhere and at any time. Context-aware mobile applications have the advantage that they can change in behavior in different situations to suit the mobile user context [18]. Cloud computing can

handle enormous amounts of data efficiently [19] and mobile cloud computing has the advantage of benefiting from the cloud, for example by exploiting the theoretically infinite cloud resources to make up for the limited mobile resources [20]. This paper proposes the cloud-based context-aware mobile ITS of Technical computer Skills (TS-ITS) to be used ubiquitously, to adapt question generation and provided help to each individual student and to benefit from the cloud resources. The idea of being cloud-based is to exploit the cloud as a central repository where questions of different types can reside. Those questions can range from simple (true and false, short-answer and multiple-choice) questions to resource-intensive interactive and VR-based exercises. This is in addition to demonstration videos that can be provided to the students as needed. While VR environments need considerable effort to be developed, short-answer, multiple-choice and true and false questions-based ITSs are promising easier to develop alternatives and so they are adopted in the current version of TS-ITS. Nevertheless, being cloud-based, TS-ITS can be easily extended by resource-intensive interactive and VR-based exercises as they become available. Such a central repository can also serve an enormous number of students maintaining their progress information to be reported to the teachers upon request.

In TS-ITS, the students are provided with a set of relevant questions and possibly demonstration videos in a context-aware fashion according to the assessed knowledge level. Scaffolding [5] is also employed, where an error in a question corresponding to a given skill can trigger videos of prerequisite skills one by one since skills are sorted topologically [21]. Additionally, a student can efficiently select a specific topic or skill(s) to learn or search for relevant limited-scope demonstration videos using keywords (regarding the *topic* the student is studying, the *skill* the student is trying to acquire or the *task* the student is trying to accomplish) for self-paced learning. As previously noted, such a tutoring system is useful for a wide range of disciplines and is needed to make up for the relatively short technical skills learning tools [11, 12]. The contributions of the paper can be summarized as follows:

- Developing TS-ITS, a tutoring system of technical computer skills, which are among the most desirable skills in many disciplines.
- The design of TS-ITS allows convenient ubiquitous learning and allows incorporating and adding resource-intensive questions and videos in addition to serving an enormous number of students maintaining their progress information.
- TS-ITS integrates context-awareness, self-paced learning and scaffolding for an effective learning process.
- TS-ITS can be adapted to other similar domains.

The following sections of the paper are ordered as follows: Section 2 is concerned with related work. Section 3 describes the proposed system, TS-ITS. Section 4 describes the prototype implementation of TS-ITS. Section 5 provides the results of the evaluation of the prototype. Finally, Section 6 provides the conclusion of the paper and possible future research and enrichments of TS-ITS.

2 Related Work

This section discusses related research in the literature. Specifically, it discusses ITSs of computer concepts and skills and mobile ITSs.

2.1 Computer concepts and skills ITSs

Several ITSs have been developed in the literature for tutoring students computer concepts and skills such as SQL database language [6], object-oriented analysis and design [7] and programming skills [8]. Operating systems in particular have gained considerable attention. For example, the Nachos [22] is an ITS developed to simulate and tutor university students operating system concepts. It includes five major components, each focusing on a single assignment allowing students to write and test corresponding code. RCOS [23] is another simulated operating system with animations. It allows students to write, compile and execute corresponding programs in Java. WebgeneOS [24] was developed as a Web-based operating systems concepts tutoring system that allows students to submit programs using Web forms to be executed remotely and receive results and comments. Another ITS has been developed to tutor students modern operating systems concepts through visualization [25].

Computer programming has also been considered in ITSs. One of the earliest systems is C-Tutor, which has been developed by Hahn et al. for tutoring students basic C language concepts [26]. In this system, program descriptions are extracted using reverse engineering and the students programs are analyzed using a debugger. The CPP-Tutor was later developed as a pilot project for tutoring students C++ programming [27]. Java programming has also been considered [28].

While the above declarative and procedural knowledge have been extensively addressed, technical computer skills have not gained similar attention. Wang et al. [29] recently developed iTutor for tutoring students basic computer skills. This is the closest system to ours. Unfortunately, it was developed in Chinese and its details are vague. Besides, it has several drawbacks that are addressed in TS-ITS. Unlike TS-ITS, skills are not fine-tuned and there is no strategy to facilitate locating a relevant limited-scope video for self-paced learning. Besides skills are not arranged in order of prerequisites and so skills cannot be mastered gradually and scaffolding is not possible. Additionally, context-awareness, for example by generating additional questions for a student with an un-mastered skill is not obvious. Another major drawback is that iTutor is not designed to be easily extended to other disciplines. Finally, it is a Web-based application rather than a mobile application and so cannot be used ubiquitously.

2.2 Mobile ITSs

Mobile ITSs have been proposed several years ago in order to benefit from the mobility that is not affected by limitations of time and location [30]. Very few research studies considered context-aware mobile ITSs. For example, Gomez et al. [31] developed a mobile application for delivering material relevant for language learning according to context specified by the student such as oral presentation or oral com-

munication. Mobile cloud computing has also been exploited to benefit from the cloud resources. For example, Khemaja and Taamallah [32] developed a language tutoring system that delivers relevant tutoring material from the cloud to children with special needs according to context including the disability and the situation. Procedural technical skills have been considered in the mobile ITS developed by Wu et al. [33] to tutor nurses required assessments for different diseases on an artificial body with sensors using the mobile as a medical tool. In this paper, we integrate context-awareness in question generation and help with scaffolding and self-paced learning in a cloud-based mobile technical skills tutoring system for an enhanced tutoring process.

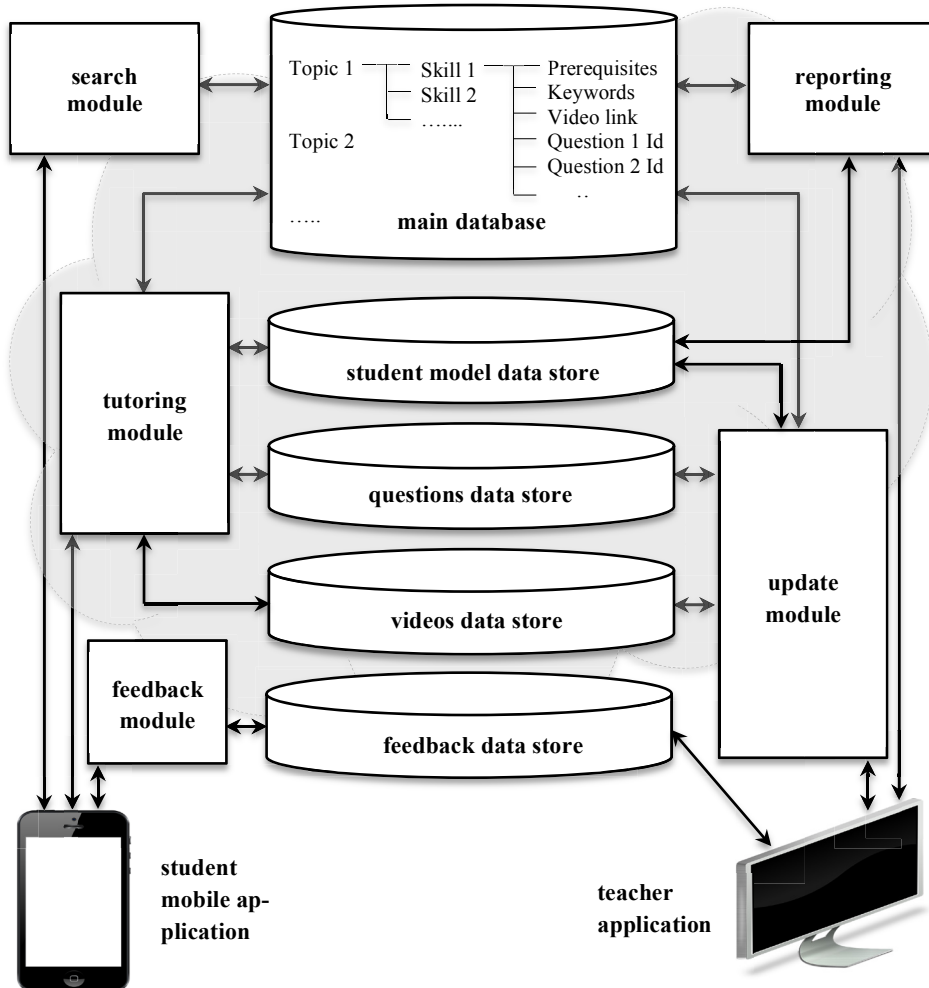


Fig. 1. Block diagram of TS-ITS (icons from pixabay.com).

3 The Proposed System

Figure 1 shows a block diagram of TS-ITS. As shown in the figure, it is formed of cloud-based main database, four data stores (videos, questions, feedback and student model) and five modules (search, tutoring, reporting, feedback and update). This is in addition to the teacher application and the student mobile application. These are explained in more details in the following sub-sections.

3.1 The main database

The main database stores the details of TS-ITS architecture. In TS-ITS, the tutoring material is classified into coarse-grained topics (and possibly sub-topics). Each topic is, in turn classified into fine-grained skills. Each skill is accompanied by a number of similar questions on different data, in different situations or in different formats. For example, suppose that the topic is *Microsoft Word*, the sub-topic is *text editing* and the skill is *formatting a word bold*. Each of the questions corresponding to this skill would address steps of formatting a given word bold, but each question involves a different word. Similarly, suppose that the sub-topic is *referencing* and the skill is *inserting a reference*. One question might address steps of inserting a journal reference formatted according to IEEE style, while another would address steps of inserting a conference reference formatted according to APA style. The database maintains the topics, sub-topics and skills hierarchy in addition to the Ids of the corresponding questions.

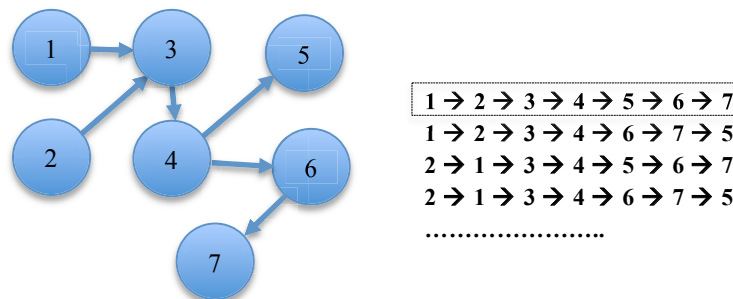


Fig. 2. An example directed graph of skill prerequisites and some possible corresponding topological sorts.

Each skill is also accompanied by a video demonstrating the steps of the skill. The database stores the video links. Additionally, each skill is accompanied by keywords and corresponding synonyms and close matches. This is in order to facilitate locating the skill and the corresponding video for self-paced learning. Normally, skills are not independent. In other words, mastering a skill may require mastering another prerequisite skill. In the above examples, *formatting a word bold* implies the ability of *selecting a word* (to be able to format it). This is why each skill is accompanied by a list

of its prerequisites. A data structure of prerequisites is automatically built in the form of a directed graph as shown in Figure 2. In this figure, skills 1 and 2 are prerequisites of skill 3, which is a prerequisite of skill 4. Skill 4, in turn is a prerequisite of both skills 5 and 6 and finally, skill 6 is a prerequisite of skill 7.

3.2 The data stores

TS-ITS includes four data stores. One data store includes the questions corresponding to the different skills. Questions can be of different formats provided that those corresponding to each skill address its steps as previously noted. A multiple-choice question is composed of a question statement and a set of choices. A true and false question or a short answer question, on the other hand is composed of a single question statement (true and false choices can be automatically inserted in the former case and a textbox can be automatically inserted in the latter). Each question may be accompanied by a figure illustrating its statement. Additionally, each question is accompanied by its correct answer and possibly explanation of the correct answer. As previously noted, other question types such as interactive and VR-based exercises can be added as they become available.

Another data store includes the videos corresponding to the different skills. Each video should be as short as possible so that the student is not distracted and can concentrate on a skill at a time. In other words, in case the student is seeking a specific skill, he/she does not have to watch a long video and extract the required skill demonstration so as not to waste time or be discouraged to watch the video. The third data store stores data required for student modeling. It records the Ids of the questions that have been correctly/incorrectly solved by each student and the timestamp of each skill trial. The fourth data store stores feedback from the students. Students can provide feedback regarding, for example the clarity and/or difficulty of the questions or the clarity and quality of the demonstration videos in addition to any suggestions regarding the database hierarchy and keywords.

3.3 The search module

The function of the search module is to help the student search for a relevant video for self-paced learning. Of course, the student can browse through the videos to select the relevant one according to the skill he/she is trying to acquire. The problem is that as the number of topics, sub-topics and corresponding skills increases, browsing through the videos would be extremely cumbersome. This problem is partly addressed, as previously noted by fine tuning the hierarchy of topics and sub-topics such that each video has a limited scope to facilitate the task of locating the relevant one. This is enhanced by providing the student with the ability to search using a keyword or a description of the task the student is trying to accomplish. Single keywords would be matched directly with the existing ones in the main database. On the other hand, descriptions provided by the student are processed to extract possible keywords (after excluding articles and conjunctions) and match them with the database keywords.

3.4 The tutoring module

The tutoring module considers skills in the order of prerequisites as indicated by the corresponding directed graph. This is achieved via topological sorting. In the *source-removal topological sort algorithm* [21], nodes with no prerequisites are pruned first. This continues until all nodes are pruned or else an error is reported. Accordingly, it determines the order of skills such that no skill is considered unless all its prerequisites have been considered. Figure 2 shows some possible topological sorts based on the shown directed graph. The first sort is selected, because it follows the default ordering of the skills (increasing order of skill Ids) as much as possible.

Whenever a skill is considered, the student is provided with a corresponding question randomly. Questions are retrieved from the questions data store guided by the corresponding Ids in the main database. If the question is correctly solved, the skill is assumed to be mastered and control transfers to the following skill. Nevertheless, if the question is incorrectly solved, the tutoring module tries to help. It provides the correct answer and allows the student to watch the corresponding video to understand the steps of the skill. A second question is then generated. If the second question is also incorrectly solved, the correct answer is generated and scaffolding starts. The module allows the student to watch the videos of the prerequisite skills one by one. A third question is then generated. If the error still persists, the correct answer is displayed, the skill is skipped to avoid getting stuck and control moves to the next skill. As previously noted, the student model module records the Ids of the questions that have been correctly/incorrectly solved by the students and the timestamp of each skill trial. It is obvious that this module is context-aware since it adapts the generation of the questions to the assessed level of knowledge of the student. The skill considered for each student depends on the skills that have been mastered (or skipped) so far. Additionally, the number of questions generated per skill (one, two or three) depends on the number of errors. This is also true about provided help and scaffolding.

3.5 The reporting module

The reporting module is responsible for reporting the student progress to the teacher upon request. The goal is to evaluate TS-ITS data and determine necessary updates. The reporting module first preprocesses the data in the student model data store:

- A skill with one correctly solved question is a skill mastered without any help.
- A skill mastered after two trials (an incorrectly solved question and a correctly solved one) has triggered video access.
- A skill mastered after three trials (two incorrectly solved questions and a correctly solved one) has triggered both video access and scaffolding.
- A skill with three incorrectly solved questions, on the other hand is a skipped skill.

For each skill the reporting module reports the percentage of students in each of those four categories. Additionally, for each skill, the module reports for each corresponding question the percentage of times it has been correctly/incorrectly solved across all students, to whom this question has been presented.

3.6 The update module

The update module is the module through which the data in the database and the different data stores can be first defined and/or inserted and later updated by the teacher. Initialization is pretty easy involving definitions of the main database hierarchy; insertion of keywords, videos and questions; and specification of the prerequisites. After specifying the prerequisites, the module generates their topological sort and in case of any error, the teacher is informed to correct it. On the other hand, decisions for update are typically based on the output of the reporting module and the information in the feedback data store. For example:

- A skill mastered after two trials for most students is interpreted as a hard skill and may signal missing prerequisite skills.
- A skill mastered after three trials for most students may also signal an ineffective video that should be revised.
- A skipped skill by most students need more careful inspection.
- A question with a relatively large percentage of error in comparison to the questions corresponding to the same skill should be reviewed for possible vagueness.
- Feedback from the students should be also considered for hierarchy, questions, videos or keywords update.

Since data in the database and the data stores are correlated, changes in the main database should be reflected in the data stores and vice versa in order to avoid inconsistencies; for example:

- The Ids and links of deleted questions and videos respectively should be deleted from the database.
- Updated questions should be given new Ids and the old Ids in the database should be replaced by the new ones.
- Questions and videos whose Ids and links are deleted from the database may be deleted from the data stores to save storage space.
- Skills that suffer from changes of questions, videos or prerequisites including corresponding videos should be deleted from the student model data store.

3.7 The teacher application and the student mobile application

The teacher application is a Web-based application that allows the teacher to easily and conveniently access the reporting module, the update module and the feedback data store. GUI items such as icons and menu items are provided for this purpose. The student application, on the other hand is designed as a mobile application to be conveniently and ubiquitously used anywhere and at any time. Unlike Web-based applications, the interface should suit the relatively small mobile screen. Through this application, the student can access the search module, the tutoring module and the feedback module. The application also allows the student to request training in a specific topic including all its skills or in a set of skills corresponding to a given topic for self-paced learning. In this case, TS-ITS starts tutoring the students as usual, but the

progress is not recorded in the student model data store. Alternatively, the questions can be downloaded to the student mobiles after assessing the adequacy of the available resources [34] to be used offline. In this case, copyrighted images delivered to the mobiles should be copyright-protected, for example via watermarking [35]. Additionally, only correct answers may be provided for help without video access.

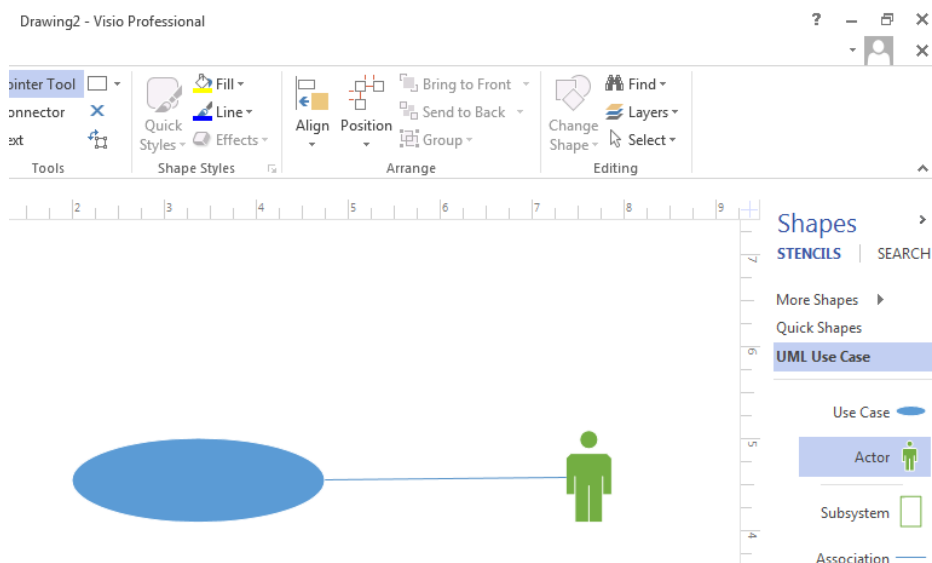


Fig. 3. An example video recorded using Camtasia Studio depicting how to draw a basic UML use case diagram.

4 Prototype Implementation

A prototype has been implemented based on TS-ITS architecture. This prototype was developed to tutor students the technical skills needed to draw Unified Modeling Language (UML) diagrams [36] using Microsoft Visio. It targeted students studying a course about object-oriented analysis and design at the university level. Using the prototype, we created a hierarchy of topics and skills. Each topic was concerned with one type of UML diagrams (such as use case diagram and class diagram). Under each topic, a set of fine-grained skills were defined. A set of multiple choice questions were also prepared under each skill. Each question was composed of a figure, a question statement and a set of multiple choice questions. Very short videos were prepared for the different skills using Camtasia Studio [37]. Figure 3 shows an example video demonstrating steps to draw a basic UML use case diagram. As previously noted, skills were arranged topologically in order of prerequisites to enable scaffolding.

To efficiently implement the search module, keywords were sorted alphabetically and each set of keywords starting with the same letter was indexed using this letter. So that search is limited to the keywords sharing the same initial with the input key-

word. Search proceeds in the candidate set using binary search [21], because it runs with complexity of $O(\log n)$, where n is the number of keywords to be matched. Each keyword indexes the corresponding video(s) to be retrieved whenever there is a match. To improve the matching process effectiveness, keywords are accompanied by synonyms and close matches. For example, the verb “insert” is accompanied by “put,” “inserted,” “inserting,” and “putting.”

5 Evaluation

This section provides the results of the empirical evaluation of the prototype implementation of TS-ITS both quantitatively and qualitatively.

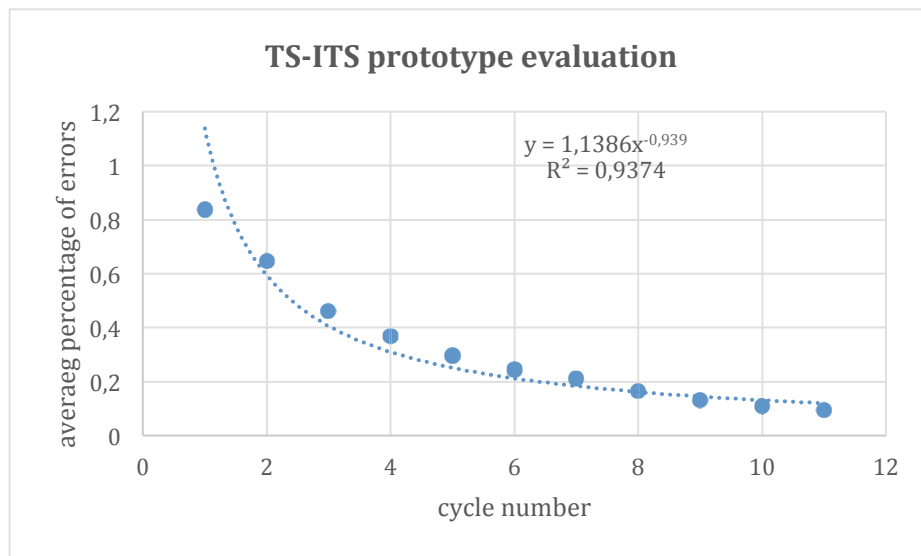


Fig. 4. Power curve depicting reduction in the average percentage of errors from one training cycle to another using TS-ITS prototype.

5.1 Quantitative evaluation

In order to evaluate the prototype implementation of TS-ITS quantitatively, we selected eighteen students who volunteered to participate in this study. The students needed to learn how to draw UML diagrams as a requirement of a course about object-oriented analysis and design. At the beginning, the students sat for a pre-test exam. They were then allowed to get trained using the tool as long as they wished for two weeks. At the end, they sat for a post-test exam. Training proceeded in cycles. In each cycle, the questions were provided in order of skill prerequisites armed by demonstration videos and scaffolding as previously noted. This continued as long as the student wished to learn or until the student solved correctly a question in the first trial of each skill in a single cycle. The fact that the post-test exam was a requirement

in the course encouraged the students to use the tool extensively for many cycles and most students kept training till the end. Figure 4 shows a power curve demonstrating the average percentage of incorrectly solved questions among all the students in each cycle. The log-log plot is depicted in Figure 5.

As shown in the figures, TS-ITS prototype was effective in tutoring the students and reducing their errors in questions regarding how to draw UML diagrams using Microsoft Visio. The average percentage of incorrectly solved questions is clearly reducing from one cycle to another. A power curve is typically the result of the effectiveness of an ITS in improving the value of the assessed variable representing the learnt concept [6]. It is worth noting that the scores of the post-test exams of the students who used TS-ITS (experimental group) showed an average improvement of 30% in comparison to the scores of their pre-test exams and 20% in comparison to the scores of the exams of the students who did not use the system (control group). In other words, TS-ITS prototype was also effective in tutoring the students and improving their technical skills in drawing UML diagrams using Microsoft Visio.

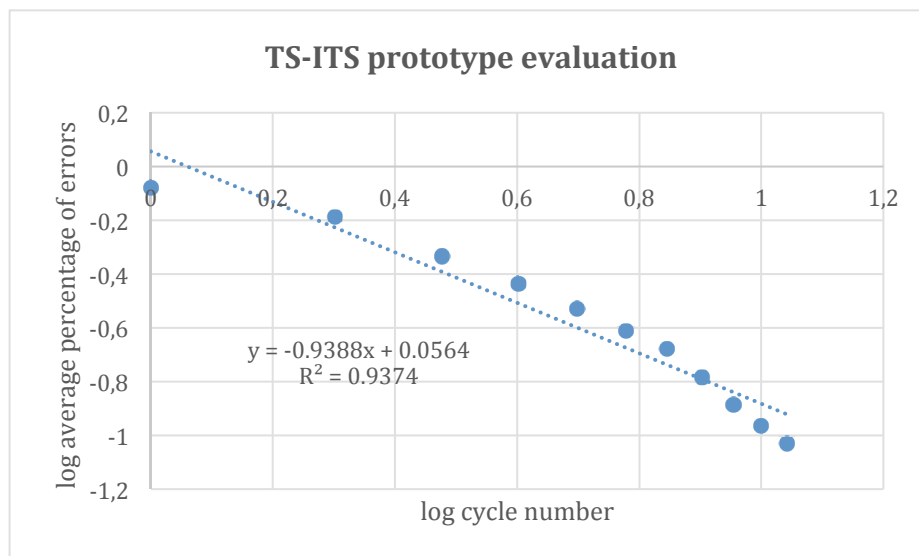


Fig. 5. Log-log plot for the average percentage of errors in each cycle using TS-ITS prototype.

5.2 Qualitative evaluation

We prepared a satisfaction questionnaire composed of a number of questions to evaluate TS-ITS qualitatively by assessing the degree of satisfaction of both the students and teachers with the novel tutoring system. The participants were required to provide an answer to each presented question on a Likert scale using one of the integer values in the range [1, 7], where a value of one represents extreme dissatisfaction with the ITS, while a value of seven represents the inverse, which is extreme satisfac-

tion. Ten teachers participated in the questionnaire in addition to the eighteen students in the experimental group that got trained using TS-ITS.

An average value was computed for each participant responses and then the global average was computed for all the participants in each of the students and the teachers questionnaires. The obtained value of the global average was 6.87 for the students and 6.58 for the teachers. Although the results indicate a very high degree of satisfaction with TS-ITS, it is interesting that the students were more excited about it in comparison to the teachers. The estimation of each evaluated factor is depicted in Table 1. As shown in the table, the participants believe that TS-ITS is an effective learning tool that is conducive to improving technical computer skills and encouraging ubiquitous and self-paced learning. They also believe it is a promising prototype of future learning systems. We estimated the reliability and internal consistency of the results of the questionnaires using *Cronbach's alpha*. We obtained values of 0.92 and 0.96 for the two questionnaires respectively reflecting extremely high degrees of reliability.

Table 1. Estimation of each evaluated factor in TS-ITS questionnaire

Evaluated factor	Student average	Teacher average
<i>Conducive to improving technical computer skills</i>	6.83	6.70
<i>Conducive to encouraging ubiquitous learning</i>	6.88	6.40
<i>Conducive to encouraging self-paced learning</i>	6.83	6.50
<i>Effective learning tool</i>	6.94	6.60
<i>Promising as a future learning system prototype</i>	6.88	6.70

6 Conclusion

This paper presented TS-ITS, a cloud-based context-aware mobile ITS of technical computer skills, that are of ultimate importance for most disciplines. TS-ITS is a mobile application that can be ubiquitously used making it a valuable tutoring tool. Mobile cloud computing technology allows storing and extending resource-intensive tutoring material on the cloud while facilitating their maintenance. This is in addition to serving an enormous number of students maintaining their progress information to be provided to their teachers upon request. TS-ITS integrates context-awareness with scaffolding and self-paced learning capability for an effective learning process. Empirical evaluation of a prototype implementation of TS-ITS shows its effectiveness in tutoring the students technical skills of using Microsoft Visio to draw UML diagrams. As future work, TS-ITS will be developed as a full system rather than merely a prototype. It will be applied to various computer skills and Microsoft Office components. It will be also extended by adding more advanced material such as interactive and VR-based immersive environments and will be adapted to other domains. We hope that TS-ITS is a step towards future ubiquitous self-paced ITSs.

7 References

- [1] Corbett, A., Koedinger, K. and Anderson, J. (1997). Intelligent tutoring systems. In *Handbook of Human-Computer Interaction*, Helander, M., Landauer, T. and Prabhu, P. (Eds.), Elsevier Science, pp. 849-874. <https://doi.org/10.1016/b978-044481862-1.50103-5>
- [2] Woolf, B. (1984). Context-dependent transitions in tutoring discourse. The National Conference on Artificial Intelligence, Austin, TX, USA.
- [3] Corbett, A., Kauffman, L., Maclaren, B., Wagner, A. and Jones, E. (2010). A cognitive tutor for genetics problem solving: Learning gains and student modeling. *Journal of Educational Computing Research*, 42(2):219-239. <https://doi.org/10.2190/EC.42.2.e>
- [4] Jaques, P., Seffrin, H., Rubi, G., de Morais, F., Ghilardi, C., Bittencourt, I. and Isotani, S. (2013). Rule-based expert systems to support step-by-step guidance in algebraic problem solving: The case of the tutor PAT2Math. *Expert Systems with Applications*, 40:5456-5465. <https://doi.org/10.1016/j.eswa.2013.04.004>
- [5] Elazhary, H. and Khodeir, N. (2017). A cognitive tutor of Arabic word root extraction using artificial word generation, scaffolding and self-explanation. *International Journal of Emerging Technologies in Learning* (in press).
- [6] Mitrovic, A. and Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, 10:238-256.
- [7] Gopalakrishnan, M., Kumar, Y. and Sangaiah, A. (2014). A method of constraint-based tutor for object-oriented analysis and design curriculum. *SSRG International Journal of Computer Science and Engineering*, 1(9):16-22.
- [8] Hooshyar, D., Ahmad, R., Yousefi, M., Fathi, M., Horng, S. and Lim, H. (2016). Applying an online game-based formative assessment in a flowchart-based intelligent tutoring system for improving problem-solving skills. *Computers & Education*, 94:18-36. <https://doi.org/10.1016/j.compedu.2015.10.013>
- [9] Yilmaz, I. and Yalcin, N. The relationship of procedural and declarative knowledge of science teacher candidates in Newton's laws of motion to understanding. (2012). *American International Journal of Contemporary Research*, 2(3):50-56.
- [10] Krathwohl, D. (2002). A revision of Bloom's taxonomy: An overview. *Theory Into Practice*, 41(4):212-218. https://doi.org/10.1207/s15430421tip4104_2
- [11] Mukuni, J. and Price, B. (2013). Portability of technical skills across occupations: A case for demolition of disciplinary silos? *International Journal of Vocational and Technical Education*, 5(2):21-28.
- [12] Dasmani, A. (2011). Challenges facing technical institute graduates in practical skills acquisition in the upper east region of Ghana. *Asia-Pacific Journal of Cooperative Education*, 12(2):67-77.
- [13] Imbert, R., Sánchez, L., de Antonio, A., Méndez, G. and Ramírez, J. (2007). A multiagent extension for virtual reality based intelligent tutoring systems. 7th IEEE International Conference on Advanced Learning Technologies, Niigata, Japan. <https://doi.org/10.1109/icalt.2007.23>
- [14] Sinatra, A. (2016). Considerations for immersive learning in intelligent tutoring systems. *International Conference on Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience*, Toronto, ON, Canada, pp 76-84. https://doi.org/10.1007/978-3-319-39952-2_8
- [15] Chang, B., Sheldon, L., Si, M. and Hand, A. (2012). Foreign language learning in immersive virtual environments. SPIE 8289, *The Engineering Reality of Virtual Reality*, Burlingame, California, USA.

- [16] Borchert, O., Hokanson, G., Slator, B., Vender, B., Yan, P., Aggarwal, V., Kariluoma, M., Marry, A. and Cosmano, B. (2013). A 3D immersive virtual environment for secondary biology education. Society for Information Technology & Teacher Education International Conference, New Orleans, Louisiana, USA.
- [17] Hospers, M., Kroezen, E., Nijholt, A., op den Akker, R. and Heylen, D. (2003). An agent-based intelligent tutoring system for nurse education. In Applications of Software Agent Technology in the Health Care Domain, Moreno, A. and Nealon, J. (Eds.), Birkhäuser Basel, pp. 143-159. https://doi.org/10.1007/978-3-0348-7976-7_10
- [18] Elazhary, H. (2015). A cloud-based framework for context-aware intelligent mobile user interfaces in healthcare applications. Journal of Medical Imaging and Health Informatics, 5:1680-1687. <https://doi.org/10.1166/jmihi.2015.1620>
- [19] Elazhary, H. (2014). Cloud computing for Big Data. MAGNT Research Report, 2(4):135-144.
- [20] Dinh, H., Lee, C., Niyato, D. and Wang, P. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. Wireless Communications and Mobile Computing, 13:1587-1611. <https://doi.org/10.1002/wcm.1203>
- [21] Levitin, A. (2012). The Design and Analysis of Algorithms, Pearson.
- [22] Christopher, W., Procter, S. and Anderson, T. (1993). The Nachos instructional operating system. The USENIX Winter 1993 Conference, San Diego, California, USA.
- [23] Jones, D. and Newman, A. (2001). RCOS.Java: A simulated operating system with animations. Computer Based Learning in Science Conference, Brno, Czech Republic.
- [24] Buendia, F. and Cano, J. (2006). WebgeneOS: A generative and web-based learning architecture to teach operating systems in undergraduate courses. IEEE Transactions on Education, 49(4):464-473. <https://doi.org/10.1109/TE.2006.882368>
- [25] Mustafa, B. (2011). Visualizing the modern operating system: Simulation experiments supporting enhanced learning. The 12th Annual Conference on Information Technology Education, West Point, New York, USA. <https://doi.org/10.1145/2047594.2047650>
- [26] Hahn, S., Song, J., Tak, K. and Kim, J. (1997). An intelligent tutoring system for introductory C language course. Computers & Education, 28(2):93-102. [https://doi.org/10.1016/S0360-1315\(97\)00003-1](https://doi.org/10.1016/S0360-1315(97)00003-1)
- [27] Abu Naser, S. (2008). Developing an intelligent tutoring system for students learning to program in C++. Information Technology Journal, 7(7):1055-1060. <https://doi.org/10.3923/itj.2008.1055.1060>
- [28] Gavrilovic, N. and Jovanovic, S. (2015). Implementing ITS for Java programming e-learning. The 6th International Conference on e-Learning, Belgrade, Serbia.
- [29] Wang, D., Han, H., Zhan, Z., Xu, J., Liu, Q. and Ren, G. (2015). A problem solving oriented intelligent tutoring system to improve students' acquisition of basic computer skills. Computers & Education, 81:102-112. <https://doi.org/10.1016/j.compedu.2014.10.003>
- [30] Brown, Q. (2009). Mobile Intelligent Tutoring System: Moving Intelligent Tutoring Systems off the Desktop. Ph.D. thesis, Drexel University, Philadelphia, PA, USA.
- [31] Gomez, S., Zervas, P., Sampson, D. and Fabregat, R. (2014). Context-aware adaptive and personalized mobile learning delivery supported by UoLmP. Journal of King Saud University - Computer and Information Sciences, 26:47-61.
- [32] Khemaja, M. and Taamallah, A. (2016). Towards situation driven mobile tutoring system for learning languages and communication skills: Application to users with specific needs. Educational Technology & Society, 19(1):113-128.
- [33] Wu, P., Hwang, G., Su, L. and Huang, Y. (2012). A context-aware mobile learning system for supporting cognitive apprenticeships in nursing skills training. Educational Technology & Society, 15(1):223-236.

- [34] Elazhary, H., Althubiani, A., Ahmed, L., Alharbi, B., Alzahrani, N. and Almutairi, R. (2017). Context management for supporting context-aware Android applications development. *International Journal of Interactive Mobile Technologies* (in press).
- [35] Elazhary, H. and Morkos, S. (2010). Blind robust transparent DCT-based digital image watermarking for copyright protection. *International Journal of Computer Science and Information Security*, 8(7): 183-188.
- [36] The Unified Modeling Language, <http://www.uml-diagrams.org/> [Online; accessed: 2017-03-01].
- [37] Camtasia, <https://www.techsmith.com/camtasia.html> [Online; accessed: 2017-03-01].

8 Author

Hanan Elazhary earned her B.Sc. and M.Sc. degrees from the Department of Electronics and Communications Engineering, Cairo University. She earned her Ph.D. degree in Computer Science and Engineering from the University of Connecticut, USA. Currently, she is an associate professor in the Computer Science Department, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia and the Computers and Systems Department, Electronics Research Institute, Cairo, Egypt. Her research interests include distributed systems, software engineering and intelligent tutoring systems.

Article submitted 07 March 2017. Published as resubmitted by the author 18 April 2017.