Intelligent Tutoring Systems Authoring Tools For Nonprogrammer Authors: A Systematic Review

Nur Rokhman[1], Syeh Abidin Kobar[2]
[1,2]Department of Computer Sciences, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada, Yogyakarta, Indonesia
[1]nurrokhman@ugm.ac.id, [2]syeh.a@mail.ugm.ac.id

Abstract— Many studies have been shown that Intelligent Tutoring Systems (ITSs) contribute to a remarkable improvement in the educational outcomes of students. Several authoring tools have been developed to contribute to making ITSs easier to build and widespread. However, there is still a lacks of current understanding in the ITSs community on how authoring tools could support nonprogrammer authors to build ITSs. This study uses a systematic literature review (SLR) method to identify primary empirical studies for nonprogrammers regarding the usage of ITSs authoring tools. The concern of this study is summarizing and analyzing the state of the art of the development of authoring tools. This study identifies the ITSs components and types that can be authored, the technologies used, and also search for proof of the advantages of ITSs authoring tools as well.

Keywords— authoring tools, intelligent tutoring systems, systematic literature review

I. INTRODUCTION

Intelligent Tutoring Systems (ITSs) are computer systems that offer automated tutoring environment to fit users based on various well-established cognitive concepts and algorithms [1]. ITSs provide personalized guidance and could adapt to each student's learning abilities, knowledge, and needs [2]. The use of ITSs for improving students' learning outcomes have been widely proven in many kinds of research [3]-[10]. However, it is difficult and costly to design and build pedagogically efficient ITSs [11]. As a result, several researchers have developed authoring tools to help the tutor development process, to speed up the development of ITSs, to reduce development costs, to increase the number of ITSs development process participants, to increase the number and diversity of tutors available, and so on [12]-[14].

Since the beginning of the ITSs study, researchers have been also researching ITSs authoring tools. At that time, the commercially available authoring tools are only for multimedia-based training and conventional computer-aided instruction, but these authoring tools still lack the sophistication needed to create intelligent tutors. Commercial multimedia authoring tools excel in providing instructional designer tools to create visually enticing and interactive displays, but behind the screens is a shallow representation of pedagogy and content [15]. Early ITSs authoring tools research fell into two broad categories: performance-oriented and pedagogy-oriented [16]. Pedagogy-oriented systems focus on how to sequence and teach relatively canned content. Meanwhile, performance-oriented systems concentrate on providing rich learning environments in which by practicing them and gaining feedback, students can develop abilities. One of the first authoring tool developed was Demonstr8 that belongs to performance-oriented systems category and use PUPS Tutoring Architecture to support the production rules authoring [17].

Due to the growing interest in ITSs authoring tools, researchers try to summarize the various research contributions to describe the state of the art in this field. A detailed study of the state-of-the-art authoring tools has been done [12]. It determines the types of ITSs built with authoring tools, the features, and techniques used, etc. Later studies in [13] and [14] used the

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038

classification of [12] to update the state of the art on the topic by adding some contributions to the authoring of various styles of ITSs, such as agent-based, and dialogue-based tutors.

Even though these contributions summarize the ITSs authoring tools knowledge well, they did not catch some interesting and recent aspects of this topic, such as evaluating the evidence provided by empirical studies, identifying cutting edge technologies and features used to support the tutor authoring for nonprogrammers, and so on. Moreover, the studies mentioned above did not perform a systematic literature review (SLR) method covering the use of authoring tools. Therefore, the purpose of this study is to do a systematic review of the usage of ITSs authoring tools for nonprogrammer authors in recent years. Thus, motivated by previous studies, this study is intended to understand:

1. What are the ITSs components that can be authored?
2. What types of ITSs can be built using authoring tools?
3. What authoring technologies have been used to build ITSs?

II. METHOD

This study uses a systematic literature review (SLR) to identify, evaluate, and interpret the available research findings related to the research questions. The studies used in this review range from over the past decade, between 2010-2020.

A. INCLUSION AND EXCLUSION CRITERIA

Studies were eligible for inclusion in this review if they were published between 2010-2020 and recognized as a peer-reviewed primary empirical study. To minimize repetitive effort, this decision on such a time was made as well as gather more recent knowledge in emerging technologies used by authoring tools.

Studies were excluded if they were non-peer-reviewed secondary study, duplicate, non-English written, gray literature, and not focusing on the use of authoring tools to support nonprogrammer authors.

B. SOURCES SELECTIONS AND SEARCH

Based on [18] recommendations, the databases were used: ACM Digital Library, Compendex, IEEE Xplore, ISI Web of Science, ScienceDirect, Scopus, and SpringerLink. The search for papers was carried out using a selection strategy based on the year, keywords, and appropriate criteria.

C. DATA EXTRACTION AND SYNTHESIS

The data extraction was performed after the search and selection processes by reading each one of the selected papers. In this process, 14 primary studies were obtained that fit the criteria and then synthesized according to the questions that had been previously asked.

III. RESULTS AND ANALYSIS

The criteria to be reviewed were met by a total of 14 papers. The results are shown in Table 3.1 sorted by year of publication.

A. AUTHORING TOOLS IN ITSS COMPONENTS

This question had a purpose to identify the main ITSs components that have been supported by the use of authoring tools. The aim of this question was to analyze ITSs components supported by the use of authoring tools. Categorization based on the recognized ITSs components which are: pedagogical model, domain model, student model, interface model (see Table 3.2) [13].

The findings shown in Table 3.2 suggest that the studies in the past decades cover all ITSs components. The pedagogical model covered 79% of the reviewed papers. This result somewhat anticipated because research on authoring tools for nonprogrammer authors aims to customize the learning process in the ITSs. There is also a significant number of studies in the domain model, about 86%. It is interesting because it shows that many studies delegate or assist authors in determining what students can learn by using the created ITSs. Nine papers

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038

have also met both the domain and pedagogical model (P2, P3, P4, P5, P8, P10, P11, P12, P13) point out the interest in using authoring tools not only to customize the learning process but also to enable the learning process to allow the definition of domain like contents, problems, and so on.

Table 3.1

ID	Year	Authors
P1	2010	Chakraborty et. al. [19]
P2	2010	Suraweera et. al. [20]
P3	2011	Chou et. al. [21]
P4	2011	Zatarian-Cabada et. al. [22]
P5	2012	Devasani et. al. [23]
P6	2014	Abbas et. al. [24]
P7	2014	MacLellan et. al. [25]
P8	2015	Blessing et. al. [26]
P9	2015	Gilbert et. al. [27]
P10	2015	Lane et. al. [28]
P11	2015	Matsuda et. al. [29]
P12	2016	Aleven et. al. [30]
P13	2020	MacLellan et. al. [31]
P14	2020	Weitekamp et. al. [32]

Table 3.2

ITSs Components	Studies	Freq.	%
Pedagogical Model	P2, P3, P4, P5, P7, P8, P10, P11, P12, P13, P14	11	79%
Domain Model	P1, P2, P3, P4, P5, P6, P8, P9, P10, P11, P12, P13	12	86%
Student Model	P1	1	7%
Interface Model	P11, P12, P13	3	21%

The usage of authoring tools to design the student model (7%) and the interface model (21%) are not so significant. For the student model, these results may happen because most of the studies rely on the artificial intelligence features of tutoring systems to automatically represent the student model during the process. However, For different categories of students, P2 still enables teachers to author some aspects of the student model. On the other hand, the interface model has so few studies because most of the authoring tools found in the reviewed papers are relying on a fixed tutor interface. It may because this component was not favored in the ITSs authoring research. Even though, few works are allowing authoring of interface model such as The Cognitive Tutor Authoring Tools (CTAT) (P12) where authors could design and create more than one tutor interface distinct to the type of the problem for which the tutor will provide tutoring. The authoring process can be done through drag and drop techniques using the existing interface builder.

B. AUTHORING TOOLS FOR ITSS TYPES

The question aimed to identify the types of ITSs that have been supported and developed by the authoring tools. Even though an ITSs could be categorized in more than one category, the classification of ITSs types in this study was identified according to the type explicitly stated in the papers. This study also defined some categories according to the ITSs features discussed in the studies. Table 3.3 shows the percentage of each type of ITSs. Two studies that could not be defined as specific ITSs type, so they were categorized as an Unspecified type.

Tabel 3.3

ITSs Types	Studies	Freq.	%
Model-Tracing/ Cognitive Tutor	P9, P11, P13, P14	4	29%
Example-Tracing	P5, P7, P12	3	21%
Content and Problem-Based	P1, P10	2	14%
Constraint-Based	P2	1	7%
Dialogue-Based	P8	1	7%
Machine and Human-Based	P3	1	7%
Unspecified	P4, P6	2	14%

Model-Tracing Tutor contains a cognitive model that is used by the tutor to check the response of the students. This model is based on the problem-solving and learning process theory of cognitive psychology. The tutor verifies each problem-solving step to keeps the student in the model path. The Cognitive Tutor is a special trademark product of Carnegie Learning that implements model-tracing and knowledge-tracing. Knowledge-tracing is used to calculate the required skills students learned to predict the performance of students in subsequent tutor interactions. The combination of both is essentially used to monitor students' learning progress, guide students to the correct path to problem-solving, and provide feedback. The emphasis

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038

of this type of ITSs studies is on a pedagogical and domain model.

Example-Tracing Tutor is also significantly identified in authoring for nonprogrammers. This type of ITSs interprets and assesses student behavior based on generalized examples of problem-solving. These examples intend to ease domain experts to create a cognitive model by demonstrating rather than by programming a production rule model [25].

Other types have smaller portions than the two types previously described. Content and Problem-Based relies on authoring tools to author content and problem specified for ITSs. Constraint-Based is based on Ohlsson's theory of learning from performance errors and are designed to decrease the effort required to develop a generic domain model. Dialogue-Based using NLP mechanisms to provide more human-like tutoring. Machine and Human-Based using both machine and human intelligence together during the tutoring process. The last is the Unspecified category which has distinct features so they did not deserve an own category. This shows that there is no common understanding in the ITSs community about the underlying theories, technologies, and features of ITSs since many researchers are developing authoring tools to build their own ITSs type.

C. TECHNOLOGIES USED IN AUTHORING TOOLS

This question is to identify the main technologies used to build authoring tools and the problems these technologies address. The question aimed to define the key technologies used in authoring tools and the problems solved by these technologies.

Table 3.4 shows that 57% of studies use AI theories, concepts, or technologies. Ontologies are used by the studies P2 and P6 to support the representation of the domain knowledge. These studies help authors in defining the tutor's domain model as well as relying on ontology's reasoning and inference capabilities to effectively use the domain model during tutoring. On other hand, to overcome different types of problems, P4 uses sophisticated algorithms based on neural networks. P8 uses NLP techniques for improving the authoring of natural language ITSs. P1 using the fuzzy rule-based approach to supports pedagogical model authoring, enabling writers to customize the rule base and define the teaching strategy. P11 developed SimStudent to help novice authors to develop a cognitive tutor by teaching it how to solve problems. P13 and P14 are the latest developments of SimStudent, namely Simulated Learner, whose learning mechanism can be modulated to separate mechanisms and can use different AI algorithms for each mechanism.

Tabel 3.4

Technologies	Studies	Freq.	%
Frameworks, Tools, Plugin	P7, P9, P11, P12, P13, P14	6	43%
AI theories, concepts, technologies	P1, P2, P4, P6, P8, P11, P13, P14	8	57%
Unspecified	P3, P5, P10	3	21%

Regarding 43% of studies addressing the use of software technologies, most of the studies (P7, P11, P12, P13) use The Cognitive Tutor Authoring Tool (CTAT), Apprentice Learner Framework (P13, P14), and Tutor Link Plugin (P9). CTAT is a widely known authoring tool, and the most complete tool to develop different types of ITSs such as Example-Tracing and Cognitive-Tutor. CTAT mainly focuses on helping authors nonprogrammer authors to effectively and efficiently develop ITSs capable of doing sophisticated tutoring behaviors in assisting students to learn in many domains [30]. On the other hand, Apprentice Learner Framework intended to support authors in developing Cognitive-Tutor by using some kind of machine teaching method to the Simulated Learner. The last is the Tutor Link Plugin, which makes it possible to extend an existing tool (called xPST) to act as an intermediary between third-party applications and the xPST engine. It maps the behavior in the interface to the proper parts of the tutor model. It can also show suggestions and other tutoring details in the application [27].

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038

IV. CONCLUSION

In this study, it shows that: the domain and the pedagogical model were mostly targeted by the authoring tools (86% of studies address the domain model, 76% address the pedagogical model, and 64% support both); several ITSs types have been supported by authoring tools to be built by nonprogrammer authors, with emphasis on Model-Tracing/ Cognitive Tutor (29%) and Example-Tracing Tutor (21%); various kinds of technologies, includes AI and software solutions have been supporting authoring of the pedagogical model, domain model, and interface model of ITSs.

However, it can be noted that further research and development is still needed in the field of ITS authoring tools, based on the results shown in this study. In order to create easier and more accessible solutions for non-programmer authors, more research on the interface model of authoring tools needs to be done; more development of authoring tools for various types of tutors; more exploration of the use of emerging technology, such as mobile learning, persuasive technologies, gamification, device-based instruction, etc.

Hopefully, the results of this systematic literature review study can be advantageous for the educational technology community especially in ITSs community as it gathers evidence from the empirical primary studies included in the review to reveals a recent body of information on the use of ITSs authoring tools for nonprogrammer authors.

REFERENCES

[1] Anderson, J. R. (1982). Acquisition of a Cognitive Skill. Psychological Review, 89, 369-406.

[2] Sarrafzadeh, A., Alexander, S., & Dadgostar, F. (2008). "How Do You Know That I Don't Understand?" A Look at The Future of Intelligent Tutoring Systems. Computers in Human Behavior, 24, 1342-1363.

[3] Beal, C. R., Walles, R., Arroyo, I., & Woolf, B. P. (2007). Online Tutoring for Math Achievement Testing: A Controlled Evaluation. Journal of Interactive Online Learning, 6, 1-13.

[4] Corbett A. (2001) Cognitive Computer Tutors: Solving the Two-Sigma Problem. In: Bauer M., Gmytrasiewicz P.J., Vassileva J. (eds) User Modeling 2001. UM 2001. Lecture Notes in Computer Science, vol 2109. Springer, Berlin, Heidelberg.

[5] Graesser, A. C., VanLehn, K., Rose, C., Jordan, P. W., & Harter, D. (2001). Intelligent Tutoring Systems with Conversational Dialogue. AI Magazine, 22, 39.

[6] Koedinger, K. R., & Anderson, J. R. (1997). Intelligent Tutoring Goes to School in The Big City. International Journal of Artificial Intelligence in Education, 8, 1-14.

[7] Mitrovic, A., Martin, B., & Mayo, M. (2002). Using Evaluation to Shape ITSs Design: Results and Experiences with SQL-Tutor. User Modelling and User-Adapted Interaction, 12, 243-279.

[8] Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. T. (2007). Cognitive Tutor: Applied Research in Mathematics Education. Psychonomic Bulletin & Review, 14(2), 249-255.

[9] Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L. Treacy, D., Weinstein, A., and Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence in Education. 15. pp. 147-204.

[10] VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. Educational Psychologist, 46(4), 197-221.

[11] Murray, T. (2005). Having It All, Maybe: Design Tradeoffs in ITSs Authoring Tools. Proceedings of The 3rd International Conference on Intelligent Tutoring Systems, 93-101.

[12] Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038

Tools: Updated Analysis of The State of The Art. In Authoring Tools for Advanced Technology Learning Environments (pp. 491-544). Springer.

[13] Woolf, B. P. (2010). Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-learning. Morgan Kaufmann.

[14] Sottilare, R., Graesser, A., Hu, X., & Brawner, K. (2015). Design Recommendations for Intelligent Tutoring Systems: Authoring Tools and Expert Modeling Techniques. US Army Research Laboratory.

[15] Murray, T. (1999). Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. International Journal of Artificial Intelligence in Education. Vol 10. pp. 98-129.

[16] Murray, T. (1997) Expanding the knowledge acquisition bottleneck for intelligent tutoring systems. International Journal of Artificial Intelligence in Education. Vol. 8 , No. 3-4, pp. 222-232.

[17] Anderson, J. & Skwarecki, E. (1986). The Automated Tutoring of IntroductoryComputer Programming. Communications of the ACM, Vol. 29 No. 9. 842-849

[18] Chen, L., Babar, M. A., & Zhang, H. (2010). Towards an Evidence-Based Understanding of Electronic Data Sources. Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering, (British Computer Society, Swinton, UK, EASE'10), 135–138.

[19] Chakraborty, S., Roy, D., Bowmick, P. K., & Basu, A. (2010). An authoring system for developing Intelligent Tutoring Systems. Proceedings of the IEEE Students' Technology Symposium (TechSym), 196–205.

[20] Suraweera, P., Mitrovic, A., & Martin, B. (2010). Widening the knowledge acquisition bottleneck for constraint-based tutors. International Journal of Artificial Intelligence in Education, 20(2), 137–173.

[21] Chou, C. Y., Huang, B. H., & Lin, C. J. (2011). Complementary machine intelligence and human intelligence in virtual teaching assistant for tutoring program tracing. Computers and Education, 57(4), 2303–2312.

[22] Zatarian-Cabada, R., Barron-Estrada, M., & Reyes Garcia, C. A. (2011). EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. Expert Systems with Applications, 38(8), 9522–9529.

[23] Devasani, S., Gilbert, S., & Blessing, S. (2012). Evaluation of two intelligent tutoring system authoring tool paradigms graphical user interface-based and text-based. Proceedings of the 21st Annual Conference on Behavior Representation in Modeling and Simulation, BRiMS 2012, 51–58.

[24] Abbas, M. A., Ahmad, W., & Kalid, K. S. (2014). Semantic Web Technologies for Preschool Cognitive Skills Tutoring System. Journal of Information Science and Engineering, 30(3), 835–851.

[25] MacLellan, C. J., Koedinger, K. R., & Matsuda, N. (2014). Authoring tutors with Simstudent: An evaluation of efficiency and model quality. International Conference on Intelligent Tutoring Systems, Springer, 551–560.

[26] Blessing, S. B., Devasani, S., Gilbert, S. B., & Sinapov, J. (2015). Using concept grid as an easy authoring technique to check natural language responses. International Journal of Learning Technology, 10(1), 50–70.

[27] Gilbert, S. B., Blessing, S. B., & Guo, E. (2015). Authoring effective embedded tutors: An overview of the extensible problem-specific tutor (xPST) system. International Journal of Artificial Intelligence in Education, 25(3), 428–45.

[28] Lane, H. C., Core, M. G., Hays, M. J., Auerbach, D., & Rosenberg, M. (2015). Situated pedagogical authoring:

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038

Authoring intelligent tutors from a student's perspective. International Conference Artificial Intelligence in Education, Springer International Publishing, 195–204.

[29] Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2015). Teaching the teacher: tutoring Simstudent leads to more effective cognitive tutor authoring. International Journal of Artificial Intelligence in Education, 25(1), 1–34.

[30] Aleven, V., McLaren, B. M., Sewall, J., van Velsen, M., Popescu, O., Demi, S., Ringenberg, M., & Koedinger, K. R. (2016). Example-Tracing Tutors: Intelligent Tutor Development for Non-Programmers. International Journal of Artificial Intelligence in Education, 26(1), 224–269.

[31] MacLellan, C. J., & Koedinger, K. R. (2020). Domain-General Tutor Authoring with Apprentice Learner Models. International Journal of Artificial Intelligence in Education.

[32] Weitekamp, D., Harpstead, E., & Koedinger, K. R. (2020). An Interaction Design for Machine Teaching to Develop AI Tutors. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 1-11.

ISSN 2442-7888 (online) DOI 10.24167/sisforma.v8i2.4038