Understanding how AI is applied in training: Case Studies

Abstract — "Artificial Intelligence" (AI) refers to technologies that emulate human intelligence, but the term is so broad that it is often hard to tell what is meant by it, how it is applied, and what value it brings. This presents a serious problem for those attempting to understand and evaluate the use of AI in training. This paper, which draws on work of the IEEE standards committee on Adaptive Instructional Systems, sheds some light on this murky area. The paper presents a framework for understanding the use of AI that clarifies inputs, outputs, the type of AI used (if any), and whether it is used to classify objects, provide recommendations, support simulations, or make decisions. The paper then illustrates the framework by applying it to use cases ranging from recommendation engines to simulations to systems that use AI to support the analysis and generation of training content.

1 Motivation

Artificial Intelligence (AI) is experiencing a renaissance due to the success of deep learning and the emergence of products such as IBM Watson, but this renaissance has led to overuse (and possibly abuse) of the term "AI" in marketing literature and capabilities descriptions. The fields of training and education are no exceptions. AI is used effectively in many systems that make personalized recommendations of learning activities and that adapt the learning to the state of the learner [1] [2], but it is difficult to interpret statements such as "*increase performance up to 50% with an AI-Driven Knowledge Cloud"* [3] and "*AI-powered knowledge technology makes training, development, and knowledge management more engaging, autonomous, intelligent, and effective than ever before*" [4]. Users, consumers, policy makers, and researchers need ways to understand what types of AI are used in training products, how they are applied, and how to evaluate their impact and training efficacy – preferably without requiring them to be experts in AI – while the manufacturers of training technologies that personalize or individualize instruction [5] have a vested interest in reducing the marketplace confusion that exists today and that has slowed the adoption of training AIS despite their potential to produce positive learning gains [6].

For the above reasons, the issue of how AI is used in *adaptive instructional systems* (AIS) is being addressed as part of a standardization effort taking place under the auspices of the IEEE Learning Technology Standards Committee (IEEE LTSC) [7]. This effort plans to produce technical interoperability standards, but its first project involves classifying AIS for the benefit of consumers, producers, and purchasers and another project involves recommended practices for evaluating AIS. This paper presents work contributed by the author to these efforts and applies it to a variety of use cases.

2 Types of AI: Rules versus Machine Learning

The Oxford dictionary defines AI as "*the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages*" [8]. This definition closely parallels that given in early books on the subjects [9]. IBM Watson gives a more current definition of AI as "*anything that makes machines act more intelligently, including basic and applied research in machine learning, deep question answering, search and planning, knowledge representation, and cognitive architectures"* [10].

Paraphrasing Arthur C. Clarke's third law [11], these definitions imply that *any sufficiently advanced machine behaviour is indistinguishable from AI*. This presents a dilemma since it is not uncommon for computer-based education and training systems to incorporate decision trees and rule engines that produce intelligence-mimicking behaviours, e.g., that make branching decisions based on assessment results or that recommend activities based on triggers and learner states. Although these algorithms create adaptive behaviours, they are deterministic and will always give the same output in response to the same input. As such they differ sharply from machine learning (ML) algorithms that learn what decisions to make and what actions to take from data and that can evolve with use.

ML can be more flexible, dynamic, and powerful than rules engines, which is often desirable, but at the same time ML can introduce unseen and unintended bias. Such bias can lie in the data used to train an algorithm, the algorithm itself, or in how an algorithm is applied. Transparency and explicability have become major themes in AI and are among the principles of ethically-aligned design [12]. In the proposed framework, components of an AIS are classified according to whether they are rule-based or ML-based, and analysts are encouraged to determine to what extent they work as intended, respect data privacy rights, and are free from (or explicit about) inherent bias.

3 Uses of AI: Decide versus Classify

In AIS, AI (whether rule-based or ML-based) is used in a limited number of ways. One set of uses involves determining or recommending the next topic, action, learning activity, or learning path for the learner or the system. In the proposed framework, these are called *decisions*. When consumers hear that a system is AI-based, they are likely to assume that the system uses AI to make decisions. However, in many systems AI is also used to generate the data on which the decisions are based. Examples of this include using algorithms to determine the knowledge, skills, aptitudes, affective state, and similar characteristics of a learner; to analyse and classify the attributes of learning resources and learning activities; to grade performance on assessments; and to determine the relevance, novelty, correctness, and other attributes of written or verbal responses. In the proposed framework, these uses of AI – which may involve rules, ML, and natural language processing (NLP) – are called *classification* because in almost all cases the goal is to classify the input data into a discrete set of categories or on a continuous scale. "*Classify*" is a passive action that produces data used by later algorithms, whereas "*decide*" is active and directs the learner or the system.

The major dichotomy in the proposed framework is between classify and decide, and in reporting which of these roles is played by a given AIS component, the analyst is asked to also identify what techniques are used. Since some, such as Bayesian Knowledge Tracing [12] and Knowledge Space theory [13], are well-documented in the literature, and others are considered proprietary and not published, and since systems may use third-party packages or services (e.g. IBM Watson), the practical goal is to understand the depth and general character of the AI used.

Another general use of AI in training systems is in interface components and "non-player characters" (NPC). These range from avatars in conversational tutors [13] to avatars, terrain, and NPC in simulation-based training and serious games. As indicated by the Wikipedia article on AI in video games [14], much of this is rule-based and term "AI" is often more of a marketing term than technology term, but in some cases NPCs adapt their expressions, language, and actions based on learner states and underlying learning science principles and play a crucial role in the personalization and individualization of the learning experience. In these cases, an analysis should include NPCs and other interface components in the list of AI-driven components of the AIS.

Finally, there are many other uses of AI in training and education. Products such as Alexa and Google Home use AI to understand and respond to commands, and it is not unreasonable to expect that someday students will be able to ask such products for help them with their homework. ML algorithms could presumably be used to learn better models for class composition or for matching students to teachers, AI is embedded in educational robots [15], AI can be used to drive virtual, augmented, and mixed reality used in training and performance support, and AI can be used develop and apply models for assessing workers as they perform their jobs based on data generated by the equipment they operate. Although the framework in this paper can be applied to these instances, the focus here is specifically on AIS.

4 Input Data

AI, and especially ML, requires data. In AIS, these data can be grouped into four broad categories:

1. *Activity Stream Data*: Learner interactions with activities and the results of those interactions. Standards such as xAPI [16] are designed to report activity stream data.
2. *Learner Data:* The learner's knowledge, skills, abilities, aptitudes, and other characteristics, potentially including affective state, biometric state, learning goals, and preferences.
3. *Activity Data:* The content of learning activities and properties such as those described in the IEEE Learning Object Metadata standard [17].
4. *Domain Data:* Data about the structure of the domain being training, often represented in the form of a topic map or competency model.

These are data that serve as inputs into algorithms that either perform classifications or make decisions. The proposed framework identifies these data, which is important for understanding the operation and limitations of the training system and for identifying potential issues concerning the governance of learner data, especially in educational settings.

The input data is meant to be data that comes from sources external to the system itself, including the learner, but in many cases internal algorithms use the results of other internal algorithms as input. For example, a combination of ML and NLP may be used to auto-classify the Bloom's levels topics, or educational alignment of a learning activity (see, e.g. [18]), all of which may be used in making recommendations. In the proposed framework, this is indicated by identifying the external data used by the classification algorithms and showing that the results are fed forward into the recommendation component.

Since ML algorithms need data to be trained, an understanding of how much data is required can be crucial and should be analysed if possible, not only in terms of the total data required but also in terms of the balance among the number of learners, the amount of data produced by each learner, and the length of time over which data is produced. If a system is delivered "pre-trained," then this may be less important but can serve as a reality check. Typically, the amount of data required to train ML-based classifiers and decision algorithms is at least measured in Gigabytes (GB), and if a system could not have been exposed to data sets of that size, then any claims of intelligence should be assumed to be rule-based or treated with healthy scepticism.

5 The Framework

As indicated by the preceding sections, the proposed framework for analysing the uses of AI in an AIS consists of identifying the major components that involve AI for personalization or individualization and, for each such component identifying:

1. the input data used;
2. whether the component uses rules or ML (and any known techniques or algorithms used);
3. whether the component decides or classifies; and

4. how data are fed forward among the components.

This can be visually summarized by representing each component on a diagram of the type shown in Fig. 1 with a short description of what each component does and indicating how data flows among them.

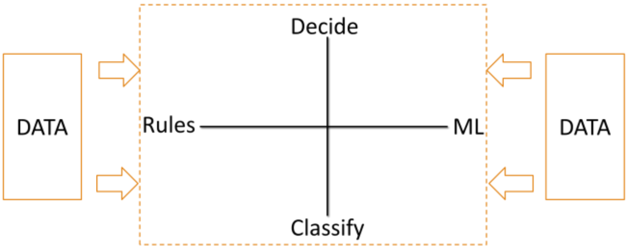

Fig. 1: Visual Representation of the Framework

A diagram of this nature suffices for many purposes but should be accompanied by a verbal description in which more detail can be provided. The next sections illustrate this using real-world AIS.

6 Gooru's Learning Navigator

The first use case presented is a recommender system, called *Learning Navigator* developed by Gooru Learning [19]. Navigator is used by millions of students in K-12 and is being adopted for use in military training as part of the US Advanced Distributed Learning (ADL) initiative's Total Learning Architecture (TLA) program [20]. Developed under the leadership of the former chief engineer for Google Research (who was also an engineer for Google Maps), Navigator aims to provide the equivalent of Google maps for learning. It does this by locating the learner within a multi-level competency framework and recommending a "route" from the learner's current location to the learner's desired location. The route consists of a series of learning activities that are catalogued by the system but that are not part of the system. In the Google maps analogy, the learning activities are the roads, stores, and houses that are not owned or maintained by Google but must be identified and understood by Google in order to recommend a route.

Gooru is presented as the first example because the way the Gooru uses AI is somewhat counterintuitive. One might expect that Gooru contains a recommender that uses collaborative filtering or other ML-based methods. In fact, routing and re-routing decisions are made by an event-condition-action table, i.e. a rules engine, that incorporates science of learning principles and that uses metadata about catalogued activities to make decisions. Gooru does use a lot of ML and NLP, but it uses it to generate the metadata! The metadata includes predictions about how often a student will acquire a given competency as a result of engaging in the activity and data, measures of engagement, associated misconceptions, and other pedagogical traits. These are computed from activity stream data, from the content of the activities, and in part from user ratings and recommendations.

Fig. 2 is a graphical representation of Navigator that summarizes how AI is used in Navigator. This is meant to provide an overview and to be augmented by a more detailed report (not included here) that, since Navigator is open source and not proprietary, explains what algorithms are used and what rules are applied.

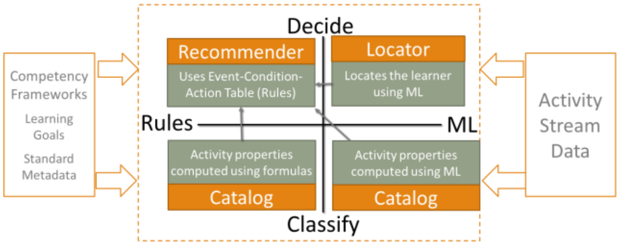

Fig. 2: Visual Representation of AI in Navigator

7 GIFT

The second example in this paper is the Generalized Intelligent Framework for Tutoring, or GIFT, which is an open source platform for creating intelligent tutoring systems (ITS) developed by the US Army Research Laboratory under guidance from Dr. Robert Sottilare. Numerous papers have been published about GIFT, and the GIFT community holds an annual symposium to exchange research and applications related to GIFT. The reader is referred to the GIFT web site for references to this material [21].

GIFT was designed around the "standard model" of an ITS that includes domain, learner (or student), expert, and pedagogical models with massive input from the ITS community. GIFT includes a domain module that uses an XML course file in which topics are hard-coded. The domain module determines what action GIFT will take based on learner state transitions. These, in turn, are tied to pre-defined instructional strategies that are hard-coded in a Domain Knowledge File (DKF). Data about the learner, such as attainment level (e.g. novice, journeyman, expert), media and delivery preferences, and motivation level, are managed by a learner module. These data often come from surveys but can also be derived from performance history and sensor input. Sensor input is a key part of GIFT, especially for non-cognitive and psychomotor domains such as marksmanship training or care under fire. Raw sensor data is managed in a sensor module that interprets sensor input into performance characterizations that are used to derive learner states. This interpretation is pre-programmed using Java. Theoretically, the Java code could implement machine-learned algorithms, but in practice the transformation of sensor data into learner states is deterministic and rule-based.

The final module in the standard release of GIFT is a pedagogical module that implements a model such as Merrill's component display theory [22]. Merrill's theory can be applied in GIFT using the "Engine for Management of Adaptive Pedagogy" (eMAP), but custom models can also be added. To apply a pedagogical model, pedagogical metadata must be added for each content object. This metadata describes how content fits into the pedagogical and learner models.

The combination of GIFT modules, content, sensor data, learner data, pedagogical models, and state transitions provide a flexible framework that can achieve multiple levels and types of adaptivity. However, without customization or alteration, these are rule-based and therefore relatively transparent compared to systems that use ML. Fig. 3 is a graphical representation of GIFT.

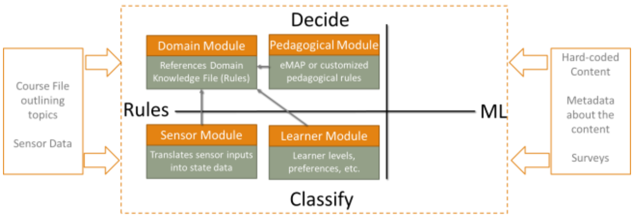

Fig. 3:Visual Representation of AI in GIFT

A GIFT example that includes an ML component is the Psychomotor Skills Training Agent-based Authoring Tool (PSTAAT) developed by colleagues of the author. PSTAAT added an expert model to use in the evaluation of psychomotor task performance. This model was trained with bio-harness data from expert and novice performers using supervised learning. The resulting classifier layered a novel application of inverse reinforcement learning [23] to compare learner performance to the models and to inform the learner how to get closer to expert performance. Fig. 4 shows how the visual representation for GIFT is updated to include the use of ML in PSTAAT.

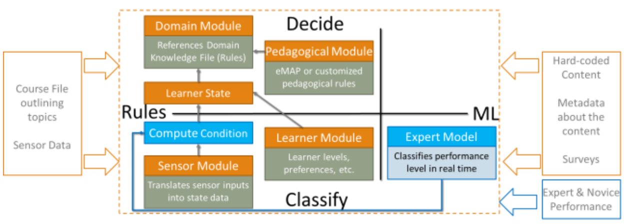

Fig. 4: Visual Representation of AI in PSTAAT (a GIFT Variant)

8 HUMAN INSTRUCTION

Having looked at two computer-based AIS, it is useful to put these in context by applying the same framework to the original adaptive instructional systems, namely human instruction. The entire field of ITS and AIS was, in some sense, launched by Benjamin Bloom's study of forms of instruction in which he determined that one-on-one tutoring showed a two-sigma effect size improvement over conventional instruction and posed the problem of achieving the same results with methods more practical than offering every student a private tutor [24]. ITS (and by extension AIS) are intended to provide the personalized and adaptive experience that is viewed as a critical causal component of the effect size observed by Bloom. Although several metanalyses of ITS conclude that they have a positive effect but that it is not close to two sigma [25], from an AI perspective a human instructor looks a lot like a relatively simple AIS with a very complicated set of inputs and even more complex classification and decision algorithms learned by the instructor.

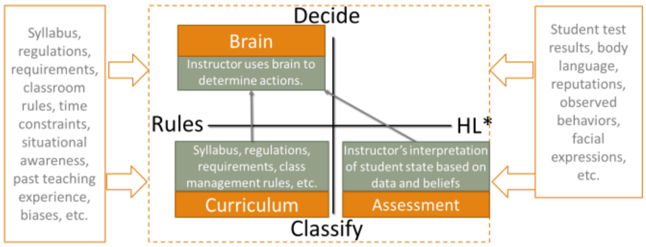

Fig. 5: Analogous Visual Representation of Human Instruction

9 KNOWLEDGE-SPACE BASED SYSTEMS

Several commercial AIS that are widely used in education, primarily in highly structured domains such as mathematics. Among the oldest of these is ALEKS, which is an acronym for Assessment and Learning in Knowledge Spaces. As explained in a study of the impact of ALEKS on an after-school math program [26], ALEKS uses a theoretical framework called Knowledge Space Theory [27] that is used to represent a student's current knowledge state and zone of proximal development (ZPD) and has developed set of assessments ranging from 400 – 700 problem types in a given subject and grade that, together with inferences learned from real-world data, can efficiently determine in which of several million knowledge states a student lies, usually on the basis of 25 to 35 questions that are adaptively selected in series based on the results of previous questions. The relationship between ALEKS assessments and a student's knowledge state are machine-learned from student results, while the underlying knowledge space is fixed.

In papers about ALEKS, ALEKS is typically described as an AI-based ITS that uses knowledge space theory, but very little is said about where the AI lies, how it works, or what data was used to train it. Some of this is proprietary, and most studies focus on the effects of using ALEKS and on the user experience rather than on its construction, but without understanding the use of AI in ALEKS it is difficult to know whether its value lies in its algorithms, in the knowledge space it uses, in its user interface, in the assessments it has developed, or somewhere else. It is also hard to know where biases might be introduced and how broadly ALEKS can be applied, both of which are required if one is to follow the principles of ethically aligned design. Although it would be difficult to apply ALEKS to fields for which no assessments exist in ALEKS, a system like ALEKS is inappropriate for student populations whose characteristics do not match those used to train the underlying algorithms or whose goal is not to learn all of a subject but instead to learn specific parts that are relevant to a related course of study.

In keeping with previous analyses, Fig. 6 is a visual representation of the use of AI in ALEKS. It is striking how much this representation looks like a simplified version of human instruction, with the difference that

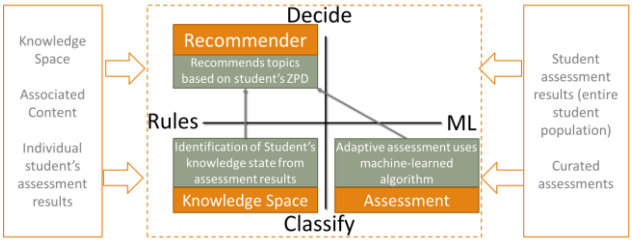

Fig. 6: Visual Representation of AI in ALEKS

ALEKS is based solely on assessment results and uses a computer algorithm to recommend topics, whereas human instructors consider a larger set of inputs and have many more actions at their disposal.

10 ElectronixTutor

As a final example, we analyse a system that was funded by the US Office of Naval Research (ONR) in the context of a STEM Grand Challenge and that is described in the International Journal of STEM Education [28]. The subject of ElectronixTutor is electronics as taught to Navy trainees in A-school. A unique aspect of this system is that it combines multiple AIS, including ASSISTments, Autotutor, Dragoon, and BEETLE-II, each of which uses different methods for adaptation. Autotutor, for example, is a dialogue-based tutoring system that, in the variation used in ElectronixTutor, uses latent semantic analysis (LSA) to compare learner inputs to text that represents correct responses and misconceptions. It then categorizes a learner's response and offers a pump (e.g., "Can you provide a bit more detail?"), a hint (e.g., What does Ohm's law say about this?"), a prompt (e.g., "If you set the resistance to ten ohms in this circuit the current will be ____?") or feedback (e.g. "Nicely done!"). This is shown visually in Fig. 7.

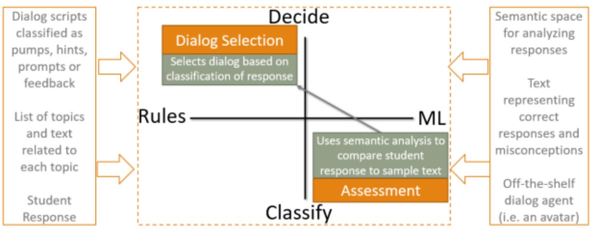

Figure 7: Visual Representation of AI in AutoTutor

At the core of ElectronixTutor is a recommender system that determines what will be presented to the learner. This recommender operates at the level of knowledge components (KCs) [29], which can be thought of as indecomposable bits of knowledge, skills, or abilities related to a subject. These are organized into more traditional topics that a student sees, which themselves are structured based on pre-requisites and on difficulty. For example, "Ohm's and Kirchhoff's Law" is a pre-requisite of both "Series and Parallel Circuits" and "PN Junctions" in a series of increasingly difficult topics that eventual lead to the study of multi-stage and push-pull amplifiers.

Each topic has associated learning resources (LR's) that are presented by the different systems aggregated in ElectronixTutor. Items include a Topic Summary (e.g. text or video), Conversational Reasoning (AutoTutor), Circuit Reasoning (multiple-choice questions from Learnform, a BBN product), Model Building (Dragoon), Circuit Basics (multiple-choice questions from BEETLE), Electronic Laws (ASSISTments) and Navy Manual Readings (from the free Navy Electricity and Electronics Training Series) [30]. Each of these LRs could have several associated items, e.g. several multiple-choice questions. The recommender guides students through topics and LRs and presents items based on assessed mastery of KCs, although the students never see and are never told about the KCs. In reference to the Van Lehn model of an ITS [31], topics are the outer loop, LRs and items are a middle loop, and the inner loop for each LR and item is handled by the system that was invoked to present the item. AI is used within ElectronixTutor components in several ways – for example in AutoTutor as described above – but the underlying recommender system is an algorithm that considers the difficulty of KCs and topics, sequencing rules derived from curricular considerations, and measures of mastery derived from performance, shown in Fig. 8

Comparing ElectronixTutor to Learning Navigator (Section 6), both gather data about learner performance at a granular level, bubble that up to coarser topic level, and recommend associated resources based on a programmed set of rules. Learning Navigator uses extensive ML and NLP methods to curate the resources. The ElectronixTutor project also used data from Mechanical Turk to estimate difficulty levels and used reading time measures from the Coh-Metrix system developed at the University of Memphis [32] which, in turn, are based on machine-learned data and NLP methods that fit into the general notion of AI. Nonetheless, the rules used in both systems can be considered transparent and explainable, with the greatest potential for unknown bias and the greatest opportunity for improving learning outcomes present in the learning activities and LRs themselves.

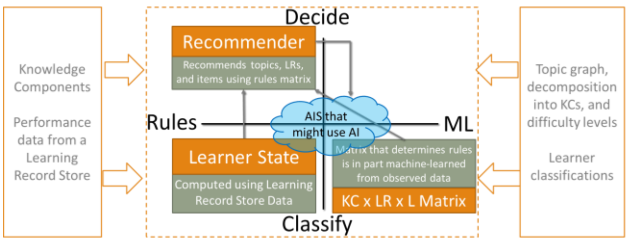

Figure8: Visual Representation of AI in ElectronixTutor

11 Conclusion and Further Work

The framework presented in this paper is being contributed to the IEEE AIS standards working group (P2247.1 on your IEEE dial). It will likely be refined as consensus is reached within the standards process. Updates and revisions are anticipated throughout 2019 and 2020 and will be reported, together with more information on detailed report formats, in future publications.

12 References

[1] B. Marr, “How Is AI Used In Education -- Real World Examples Of Today And A Peek Into The Future,” Forbes, Inc., 25 July 2018. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/07/25/how-is-ai-used-in-education-real-world-examples-of-today-and-a-peek-into-the-future/#7d0a9812586e. [Accessed 13 DEC 2018].

[2] A. Hathaway, “The Promise (and Pitfalls) of AI for Education,” THE Journal, 29 August 2018. [Online]. Available: https://thejournal.com/articles/2018/08/29/the-promise-of-ai-for-education.aspx. [Accessed 13 DEC 2018].

[3] A. Sivistrava, “Learning Experience Platforms – Top LXP Solutions,” [Online]. Available: https://www.learninglight.com/top-lxp-solutions/. [Accessed 13 DEC 2018].

[4] Volley for Enterprise, “Your personal knowledge engine.,” [Online]. Available: http://www.volley.com/. [Accessed 13 DEC 2018].

[5] D. Basye, “Personalization vs Differentiation vs Individualization,” International Society for Technology in Education, 24 January 2018. [Online]. Available: https://www.iste.org/explore/articleDetail?articleid=124. [Accessed 15 DEC 2018].

[6] R. Robson and A. Barr, “Lowering the Barrier to Adoption of Intelligent Tutoring Systems through Standardization,” in *Design Recommendations for Adaptive Intelligent Tutoring Systems Learner Modeling (Vol. I),* R. Sottilare, H. Holden, A. Graesser and X. Hu, Eds., US Army Research Laboratory, 2013, pp. 7 - 14.

[7] IEEE LTSC, “Adaptive Instructional Systems (C/LT/AIS) P2247.1,” [Online]. Available: http://sites.ieee.org/sagroups-2247-1/. [Accessed 13 DEC 2018].

[8] Oxford Dictionaries, “Artificial Intelligence,” [Online]. Available: https://en.oxforddictionaries.com/definition/artificial_intelligence. [Accessed 13 DEC 2018].

[9] A. Barr, E. Feigenbaum and C. Roads, The Handbook of Artificial Intelligence, vol. 1, Kaufmann, William Inc, 1982.

[10] IBM, “Artificial Intelligence (subdiscipline),” [Online]. Available: https://researcher.watson.ibm.com/researcher/view_group.php?id=135. [Accessed 13 DEC 2018].

[11] Wikipedia, “Clarke's three laws,” Wikimedia, [Online]. Available: https://en.wikipedia.org/wiki/Clarke%27s_three_laws. [Accessed 2013 DEC 2018].

[12] IEEE, “Ehtics in Action: The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems,” 2018. [Online]. Available: https://ethicsinaction.ieee.org/. [Accessed 16 DEC 2018].

[13] A. Graesser, K. VanLehn, C. Rose, P. Jordan and D. Harter, “Intelligent tutoring systems with conversational dialogue,” *AI Magazine,* vol. 22, no. 4, pp. 39 - 51, 2001.

[14] Wikipedia, “Artificial intelligence in video games,” Wikimedia Foundation, [Online]. Available: https://en.wikipedia.org/wiki/Artificial_intelligence_in_video_games. [Accessed 15 DEC 2018].

[15] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati and F. Tanaka, “Social robots for education: A review,” *Science Robotics,* vol. 3, no. 21, 2018.

[16] Advanced Distributed Learning Initiative, “xAPI Overview,” [Online]. Available: https://www.adlnet.gov/experience-api. [Accessed 14 DEC 2018].

[17] IEEE Standards Association, “1484.12.1-2002 - IEEE Standard for Learning Object Metadata,” 13 05 2009. [Online]. Available: https://ieeexplore.ieee.org/document/1032843. [Accessed 15 DEC 2018].

[18] R. Robson, F. Ray and Z. Cai, “Transforming Content into Dialogue-based Intelligent,” in *The Interservice/Industry Training, Simulation , and Education Conference*, Orlando, 2013.

[19] Gooru Learning, "Learning Navigator: Every Student Achieves Mastery," 2019. [Online]. Available: https://gooru.org/welcome/. [Accessed 14 April 2019].

[20] ADL, "Total Learning Architecture (TLA)," [Online]. Available: https://www.adlnet.gov/projects/tla. [Accessed 2015 DEC 2018].

[21] US Army Research Laboratory, "GIFT Documents," [Online]. Available: https://www.gifttutoring.org/projects/gift/documents. [Accessed 14 April 2019].

[22] D. Merrill, "Component Display Theory," in *Instructional Design Theories and Models: An Overview of their Current States,* Routledge, 1983, pp. 111 - 158.

[23] P. Abbeel and A. Y. Ng, "Apprenticeship Learning via Inverse Reinforcement Learning," *Proceedings of the twenty-first international conference on Machine learning,* pp. 1 - 8, 2004.

[24] B. S. Bloom, "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring," *Educational Researcher,* vol. 13, no. 6, pp. 4 - 16, 1984.

[25] A. Alkhatlan and J. K. Kalita, "Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments," 2018. [Online]. Available: https://arxiv.org/pdf/1812.09628. [Accessed 14 April 2019].

[26] S. D. Craig, X. Hu, A. C. Graesser, A. E. Bargagliotti, A. Sterbinsky, K. R. Cheney and T. Okwumabua, "The impact of a technology-based mathematics after-school program using ALEKS on student's knowledge and behaviors," *Computers & Education,* vol. 68, pp. 495 - 504, 2013.

[27] J.-P. Doignon and J.-C. Falmagne, Knowledge Spaces, Springer Sicence & Business Media, 2012.

[28] A. C. Graesser, X. Hu, B. D. Nye, K. VanLehn, R. Kumar, C. Heffernan, N. Heffernan, B. Woolf, A. M. Olney, V. Rus, F. Andrasik, P. Pavlik, Z. Cai, J. Wetzel, B. Morgan, A. J. Hampton, A. M. Lippert, L. Wang, Q. Cheng, J. E. Vinson, C. N. Kelly, C. McGlown, C. A. Majmudar, B. Morshed and W. Baer, *International Journal of STEM Education,* vol. 5, no. 15, 2018.

[29] LearnLab, "Knowledge Component," Cargegie Mellon University, [Online]. Available: https://www.learnlab.org/research/wiki/Knowledge_component. [Accessed 14 April 2019].

[30] A. C. Graesser, *ElectronixTutor Recommender System,* 2017 (Privately shared Google Doc).

[31] K. Vanlehn, "The behavior of tutoring systems," *International Journal of artificial intelligence in education,* vol. 16, no. 3, pp. 227 - 265, 2006.

[32] Coh-Metrix, [Online]. Available: www.cohmetrix.com. [Accessed 14 April 2019].

[33] R. Baker, A. Corbett and V. Aleven, "More Accurate Student Modeling Through Contextual," in *Intelligent Tutoring Systems*, Springer, 2008.

[34] D. Albert and C. Hockemeyer, "Adaptive and Dynamic Hypertext Tutoring Systems," in *Artificial Intelligence in Education,* Kobe, Japan, 1997.

[35] I. Witten, F. Elbe, M. A. Hall and C. J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.

[36] B. Bloom, Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain, Longman, 1956.

[37] Schema.org, "AlignmentObject," [Online]. Available: https://schema.org/AlignmentObject. [Accessed 15 DEC 2018].