International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:13, No:11, 2019

Leveraging xAPI in a Corporate e-Learning Environment to Facilitate the Tracking, Modelling, and Predictive Analysis of Learner Behaviour

Libor Zachoval, Daire O Broin, Oisin Cawley

***Abstract*—**E-learning platforms, such as Blackboard have two major shortcomings: limited data capture as a result of the limitations of SCORM (Shareable Content Object Reference Model), and lack of incorporation of Artificial Intelligence (AI) and machine learning algorithms which could lead to better course adaptations. With the recent development of Experience Application Programming Interface (xAPI), a large amount of additional types of data can be captured and that opens a window of possibilities from which online education can benefit. In a corporate setting, where companies invest billions on the learning and development of their employees, some learner behaviours can be troublesome for they can hinder the knowledge development of a learner. Behaviours that hinder the knowledge development also raise ambiguity about learner's knowledge mastery, specifically those related to gaming the system. Furthermore, a company receives little benefit from their investment if employees are passing courses without possessing the required knowledge and potential compliance risks may arise. Using xAPI and rules derived from a state-of-the-art review, we identified three learner behaviours, primarily related to guessing, in a corporate compliance course. The identified behaviours are: trying each option for a question, specifically for multiple-choice questions; selecting a single option for all the questions on the test; and continuously repeating tests upon failing as opposed to going over the learning material. These behaviours were detected on learners who repeated the test at least 4 times before passing the course. These findings suggest that gauging the mastery of a learner from multiple-choice questions test scores alone is a naive approach. Thus, next steps will consider the incorporation of additional data points, knowledge estimation models to model knowledge mastery of a learner more accurately, and analysis of the data for correlations between knowledge development and identified learner behaviours. Additional work could explore how learner behaviours could be utilised to make changes to a course. For example, course content may require modifications (certain sections of learning material may be shown to not be helpful to many learners to master the learning outcomes aimed at) or course design (such as the type and duration of feedback).

***Keywords*—**Compliance Course, Corporate Training, Learner Behaviours, xAPI.

I. Introduction

E-LEARNING platforms, often referred to as Learning Management Systems (LMSs) such as Blackboard, Moodle, and Canvas have been widely adopted by schools and universities around the globe. However, the types of data that can be collected by these systems are very limited due to the limitations of SCORM [1]. This data limitation constraints the use and applicability of many AI and machine learning algorithms that could be used to refine and further adapt online courses [2], [3], often these algorithms require a huge amount of data and appropriate features which often could not be captured. However, the xAPI tares down these barriers. The xAPI can capture a vast number of additional data types across multiple devices whether the user is online or offline [4]. Many innovative opportunities arise from this development but also uncertainties, such as, what data are worth capturing.

The corporate world is highly competitive and in order for companies to stay competitive they invest into the training and development of their employees [5], [6]. Training and development may involve mandatory compliance courses which employees are required to pass. The emphasis on passing may divert focus away from learning towards passing the test. At present, scores achieved in multiple-choice questions gauge the mastery of a learner. However, this is a naive approach since a course may be passed without possessing the required knowledge. Learners can attempt to "game-the-system" by exploiting specific features of the system or the course to obtain correct answers [7], [8]. The identification of behaviours around gaming the system is quite a popular topic in the Intelligent Tutoring Community as gaming behaviours lead to poorer learning [7], [8]. In a corporate setting, such behaviours are troublesome, because the company receives only a little benefit from their investment if employees are passing courses without possessing the required knowledge and potential compliance risks may arise. Therefore, it is important to be aware of behaviours that hinder the development of learner's knowledge and consider them while gauging one's knowledge mastery.

II. Terminology

A. SCORM

SCORM carries two meanings in its name [9]-[12]. The first meaning lies in the Shareable Content Object (SCO), a simple learning object that together with a combination of other SCOs may form a course (an indication of object creation which can be shared across systems). The second meaning lies in the Reference Model (RM), a description of how existing technical specifications may be properly used by the developers, in this case it defines a way to construct an e-learning platform such as Learning Management System

L. Zachoval, D. O'Broin, and O. Cawley are with the Institute of Technology Carlow, Co. Carlow, Ireland (e-mail: Libor.Zachoval@itcarlow.ie, Daire.OBroin@itcarlow.ie, oisin.cawley@itcarlow.ie).

(LMS) or Virtual Learning Environment (VLE) so that its content can be shared with other SCORM conformant systems. SCORM has become the de facto industry standard for the way web-based content works within an e-learning system [4], [9]. Apart from being a standard, SCORM is used to track progress (such as, curriculum completion) and performance (for example, score or pass/fail) of a learner using Shareable Content Objects (SCOs). An SCO, also referred to as electronic units of learning, or training components represents the most granular piece of training that is both independent and reusable [9], [11], [13]. Table I contains a list of features that an SCO can capture.

TABLE I
A LIST OF FEATURES THAT AN SCO CAN CAPTURE [10], [11]

Feature	Description
Completion	Whether or not has the learner finish the course.
Score	What score did the learner achieve? For example, the test score.
Pass/Fail	Has the student failed or pass the course?
Duration	How long did it take the learner to complete the course?
Bookmark	Allows the learner to bookmark a page (SCORM 2004 Feature).

Despite the benefits offered by SCORM, there are also many limitations [4], [10]: it is difficult to edit published content because it needs to be re-authored and redeployed; SCORM requires JavaScript in a web browser to function which makes it insecure; SCORM stores tracking data and solutions to questions in the web browser cache meaning it can be accessed by the learner; SCOs aren't easily shareable, they require multiple copies of a single file when reusing SCOs in different courses; other issues include elaboration on reports and in-depth analysis of user activity.

B. xAPI

The Experience API, also known as tin-can API, is the "successor" [4] or "next-generation" [12] of SCORM. The xAPI has adopted all key benefits of SCORM but with more advantages [4], [14]: it can capture the various learning types (e.g. blended learning, serious games, mobile learning); it doesn't require a browser-based application or JavaScript; it can work outside of an LMS; evolved portability due to the Learning Record Store (LRS, a data store system that serves as a repository for learning records which are stored as a collection of data about a learner stored as triples (actor, verb, object) necessary for using the Experience API, allowing for online and offline data capture); in-depth analysis of assessments and results. The xAPI allows for capturing of results, interactions during actions, events and due to its use of the JSON format. The JSON format makes it easy to create hierarchical trees between objects and visualise how data is related. The xAPI is a promising solution "for better interoperability between different types of educational systems and devices" [15]. It is very versatile, and it can work with any language as long as that system can communicate with LRS [4]. It has been used for adaptive training, for example, to reduce training time for the soldiers while maintaining the same quality of the training through the use of a virtual simulation training system (Unstabilised Gunnery Trainer) [16]. Furthermore, there is a trade-off between interoperability and data collection performance. Dodero et al. [17] discovered that common LRS implementations may not perform well for simulation engines and real-time systems.

C. Learner Behaviours

This paper focuses on behaviours that hinder the knowledge development of a learner. Specifically, gaming behaviours. The identification of gaming behaviours is a popular topic in the Intelligent Tutoring System (ITS) community. An ITS is a system that operates as a private tutor, it provides learners with personalised instructions by adapting to each learner's behaviour separately [18], [19]. Gaming behaviours, also referred to as "gaming the system" is associated with poorer learning and refers to when a learner exploits specific features of the system (hints and feedback) to obtain correct answers rather than by possessing the required knowledge [7], [8], [20], [21]. Three types of gaming behaviours were identified by other researchers, guessing, hint abuse, and question avoidance [7], [8], [20]-[22]. Guessing suggests that the learner selects answers at random, either in a single order (option 1) or systematically (option 1, option 2, option 3, and so forth). The learner can also use similar answers or try all combinations of possible answers in multiple-choice questions. Hint abuse refers to a learner going through all hints until an answer is given (bottom-out hints), or the learner answers incorrectly on purpose to receive the answer through feedback. Question avoidance is related to skipping problems that are too difficult or changing between problems. Several systems were developed to detect gaming behaviours via Machine Learning [7], [8], Knowledge Engineering [20], and a hybrid model which is a combination of the two [21].

III. RELATED WORK

No work has been found in the area of examining gaming behaviours in corporate training or compliance courses to date. However, the identification of gaming behaviour, a form of off-task behaviour has received a lot of attention from the Intelligent Tutoring Community because it was associated with poorer learning [7]. The first type of gaming behaviour that was discovered was hint abuse [22], which has led to the development of a help-seeking tutor agent model (a rule-based model which contains 57 rules that capture both productive and unproductive help-seeking behaviours). It can be considered as a type of intervention system. Since then a machine learning model was developed [8]. A Latent Response Model was selected by the researchers to detect learners that are hurt by gaming the system. This model was trained on 26 features consisting of students' action logs, students' learning outcomes, and human-coded observations. Researchers identified two types of gaming students. One that does not get hurt, despite gaming the system it had little effect on their post-test results. Second that gets hurt, resulting in lower post-test results, primarily due to a low initial knowledge on the subject. An interesting hypothesis raised by the researchers was that students may choose to game the system when it hurts their learning the most, specifically, when they are having difficulties with the material. Another model considered for the task of detecting gaming behaviour is a knowledge engineered model [20]. Researchers developed this model via a cognitive task analysis to learn how experts determine whether students are gaming the system or not. The authors interviewed an expert and refined their model resulting in 13 rules that identify gaming behaviours. A relatively new model [21], a hybrid model which combined the machine learning features with knowledge engineering features was found to outperform the machine learning model across different contexts. It achieved comparable results to the knowledge engineered model which was found to generalise well between contexts of scatterplots and algebra, making it applicable on different subjects.

IV. Study

Unusual learning behaviour was observed during the initial analysis of an anonymised dataset for a single compliance course. Learners who have undertaken this course repeated the post-test several times in a row, despite constantly failing. This observation suggests that learners pass the test by learning correct answers rather than by knowing the course material.

A. Design

The objective of this study was to identify learner behaviours that might affect knowledge development present in a corporate compliance course. To achieve this, several steps had to be undertaken. An in-depth analysis of the dataset was required to identify any useful features that could represent gaming behaviours (for example, the number of tests taken, or time between tests). The extraction of available features from the JSON statements captured by the xAPI had to be considered (for example, the number of characters present in the asked question, or the number of options available). Feature engineering was an important step in this study. Features were engineered similarly to those used by [8] and additional appropriate features were added, such as the number of tests taken. The engineered dataset was then evaluated utilising rules that were identified by [20].

B. Compliance Course

A compliance course was equipped with xAPI statements. The xAPI statements were focused on capturing in-test information, specifically, information about the question at hand, response and confidence level selected, and the test outcome. The compliance course was a short course composed of a pre-test (optional), learning content, and a post-test. Questions on either of the two tests were randomly selected out of a question pool that contained 15 questions. The course was not equipped with any form of a hint mechanism. The pre-test contained 10 random questions, a combination of multiple-choice and true/false questions, If the learner achieved 100% in the pre-test then they did not have to take the course. The post-test contained five random questions, also a combination of multiple-choice and true/false questions, the learner was also required to achieve 100%, however, if a question was answered incorrectly immediate feedback was provided with the correct answer to the posed problem. During the post-test, a confidence level had to also be selected by a learner for each question (low, medium, high). However, selecting a confidence level had no effect on the learner, specifically, if a learner answered incorrectly and selected high confidence, the learner was not penalised.

C. Data

Data were gathered for a duration of eight months, capturing 125,317 statements in total (for the single compliance course). The raw dataset was represented in the form of a JSON format from which specific features had to be extracted. Refer to (Table II) for the list of extracted features. The cleansed dataset consisted of 110,630 rows of data.

TABLE II
A LIST OF EXTRACTED FEATURES

Feature	Type	Description
Learner ID	String	The unique anonymised ID of a learner.
Question Asked	String	For example, what is 1+1?
Result	Number	1 = Correct, 0 = Incorrect.
Confidence	Number	0 = low, 1 = medium, 2 = high.
Options Available	String	List of options (option1, option2).
Option Selected	String	For example, option1.
Timestamp	Time	The time when this message was recorded.

The list of features displayed in (Table II) was also used to represent the test results. “Learner ID” was set, “Result” represented whether a learner passed the test, and “Timestamp” represented the time when the message was recorded. The rest of the results were set to a default value. The outcome of the data extraction resulted in a dataset that consisted of 110,630 rows of data split between pre-tests (21,520 rows) and post-tests (89,110 rows).

For this study, the authors were only interested in examining the post-test data (89,110 rows) as pre-test was skipped by most learners and could have only been taken once. The post-test data contained both attempted questions (74,749 rows) and test results (14,361 rows), for 5,526 unique learners.

D. Methodology

The dataset was feature engineered similarly to the feature list described by [8] and critically analysed by the rules identified by the current state-of-the-art on the detection of gaming behaviours, specifically rules identified by [20]. The data was engineered on a test level. In other words, a row of data contained information specific to the test and relative to the learner who took it, specifically, per question information. The test-specific information contains features such as the test count, correct and incorrect responses, whether they were first attempts, time spent on the test, time spent from the last test to this test (time between tests), the difficulty of the test, and so forth. Per question information is relative to the learner, meaning it is updated after each test the learner takes. For example, question 8 is in three consecutive tests and the learner tries a different option every time (for example option

1, option 2, and option 3), the result of this is option 1 on the first test, and option 1, option 2, and option 3 on the third test. Per question information includes features such as confidence selected, the option selected, difficulty, number of times attempted, and so forth. Each post-test had exactly five questions chosen at random from a pool of 15 questions. The engineered dataset resulted in 14,259 rows of data and 102 possible features (12 features per question on the test).

Once the dataset was engineered it was examined. Rules that indicate gaming behaviours were considered. Jupyter Notebook was utilised and with the use of Python programming language a sequence of tests was examined for each learner looking for specific information, such as whether a learner managed to use majority if not all of the options available for a multiple-choice question, how many tests were completed using a single option only, and how many times the learner spent less than 2.5 minutes before taking the post-test again (this value was selected from the data analysis stage as most learners repeated the test within that timeframe, and furthermore it does not give them the opportunity to go over the course material). Additional rules were considered, however, due to the limitation of the dataset they were not plausible. An example of such limitation was considering the amount of time spent on an individual question as it included time spent on immediate feedback.

E. Findings

This study has led to three outcomes: a list of identified learner behaviours, and features that describe these behaviours (see Table III), and a list of limitations. The three behaviours that were identified in this study are: guessing by trying all of the options (for multiple-choice questions only); guessing by using a single option (and answering mostly incorrectly on the test), where the learner uses a single option for all of the questions on the test; and continuously repeating tests (applies to learners who took more than three tests due to the limitations of the current dataset). From 5,526 unique learners, 1,259 (22.78%) exhibited at least one gaming behaviour and out of 14,259 rows of the engineered dataset, 6,065 (42.53%) were flagged with a gaming behaviour.

TABLE III
A LIST OF FEATURES USED TO IDENTIFY THESE BEHAVIOURS.

Behaviour	Features
Guessing by trying all the options (for multiple-choice questions only)	The ID of the question Options tried by the learner for this question Options available for the questions
Guessing by using a single option (and answering mostly incorrectly)	The ID of the question Selected option for a question
Continuously repeating tests (and completing more than 3 tests)	Time between tests Number of tests taken

This study has identified several limitations with the current data captured approach, mostly related to insufficient data points which lead to a certain level of ambiguity. In the given dataset, it was impossible to determine whether learners who passed within the first three tests passed by possessing the required knowledge, or whether they guessed and got lucky. The only way to determine when the test began was by predicting how long the learner might spend before answering the first question. The learner was provided with immediate feedback upon submitting an answer to a question which was also not accounted for in the dataset. If these points were present in the current dataset additional behaviours would have been identified. Other points worth capturing are how much time learners invest on the learning content or individual sections of the course. These points could also provide further insights into additional behaviours, both positive and negative, or shed light onto why they occur. The course structure also needs to be considered, upon failing the test, the learner was brought back to a page right before the test, and the navigation was limited to a single arrow to traverse through the course. Such a design could potentially encourage the learner to try again and game the system, rather than to search for a page that may contain the required information, especially since there were no penalties for continuously failing. This suggests that gauging the mastery of a learner from a score on multiple-choice questions alone is a very naive approach as learners can pass a course by exploiting specific features of an e-learning platform or the course design itself.

V. CONCLUSION

This paper has discussed: the limitations of SCORM and the benefits of xAPI in e-learning platforms, and how the xAPI was utilised by a compliance course hosted by a corporate e-learning environment, which resulted in the identification of three learner behaviours and limitations present by the data captured approach. The three learner behaviours that were identified in the anonymised dataset were, trying each option for a question, selecting a single option for all questions on the test, and constantly retaking the test upon failing as opposed to going over the learning material. These behaviours are a form of off-task behaviours, associated with gaming the system and lead to poorer learning. These findings suggest gauging learner mastery from a score on multiple-choice questions is not enough because learners can find other ways to obtain correct answers and avoid the learning process. The limitations presented by the dataset were related to insufficient data points, for instance, the inability to accurately determine when the test began, or how long a learner spent on a question.

VI. FUTURE WORK

The next steps are to: consider the incorporation of additional data points, analyse a dataset containing additional nine compliance courses, utilise knowledge estimation models to obtain a more accurate representation of learner mastery, and carry out a correlation analysis to measure the impact that identified learning behaviours have on knowledge development. Models considered for the task of knowledge estimation are: Item Response Theory, Bayesian Networks, and an Overlay model (serving as a baseline). These probabilistic models will be built using performance-specific features (such as, correct/incorrect response, question difficulty, number of characters in the question) and learner-specific features (such as, opportunity count, number of tests completed). Selected approaches are parameterised using a data-driven approach. The trained models are validated against each other by predicting whether a learner passes the test. The dominant approach is selected to estimate the knowledge of a learner. A correlation analysis is then carried out to measure the impact that identified learning behaviours have on the knowledge development of a learner. For example, continuously repeating tests upon failing may correlate with decreased knowledge, whereas, spending time on the learning content instead could correlate with increased knowledge. Further work could explore how learner behaviours may be utilised to make changes to a course. For example, course content modifications (certain sections of learning material may be shown to not be helpful to many learners to master the learning outcomes aimed at) or on the course design (such as the type and duration of feedback).

Acknowledgment

This work was funded by the Irish Research Council under the Postgraduate Enterprise Partnership Scheme.

References

[1] Gibson, M., 2013. Does SCORM prevent good education? (WWW Document). URL http://markgibson.info/2013/07/does-scorm-prevent-good-education.html (accessed 3.20.18).
[2] Graf, S., List, B., 2005. An evaluation of open source e-learning platforms stressing adaptation issues. IEEE, pp. 163–165. https://doi.org/10.1109/ICALT.2005.54
[3] Aleven, V., Sewall, J., Popescu, O., Xhakaj, F., Chand, D., Baker, R., Wang, Y., Siemens, G., Rosé, C., Gasevic, D., 2015. The Beginning of a Beautiful Friendship? Intelligent Tutoring Systems and MOOCs, in: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (Eds.), Artificial Intelligence in Education. Springer International Publishing, Cham, pp. 525–528. https://doi.org/10.1007/978-3-319-19773-9_53
[4] My e-Learning World, 2016. SCORM vs Tin Can vs AICC: Which Best Suits Your Needs? (WWW Document). My Blog. URL https://myelearningworld.com/scorm-vs-tin-can-vs-aicc-the-lms-standard-showdown/ (accessed 3.20.18).
[5] Grossman, R., Salas, E., 2011. The transfer of training: what really matters: The transfer of training. International Journal of Training and Development 15, 103–120. https://doi.org/10.1111/j.1468-2419.2011.00373.x
[6] Little, B., 2014. Best practices to ensure the maximum ROI in learning and development. Industrial and Commercial Training 46, 400–405. https://doi.org/10.1108/ICT-08-2014-0051
[7] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z., 2004. Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System," in: In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 8.
[8] Baker, R.S.J.D., Corbett, A.T., Roll, I., Koedinger, K.R., 2008. Developing a generalizable detector of when students game the system. User Modeling and User-Adapted Interaction 18, 287–314. https://doi.org/10.1007/s11257-007-9045-6
[9] Bailey, W., 2005. standards briefings series 4.
[10] Parr, D., 2016. SCORM – Set of Standards that make Courses and LMS Compatible. Paradiso eLearning Blog.
[11] Rustici Software, 2018b. One Minute SCORM Overview for Anyone - SCORM – (WWW Document). SCORM -. URL https://scorm.com/scorm-explained/one-minute-scorm-overview/ (accessed 3.5.18).
[12] Rustici Software, 2018c. SCORM Explained (WWW Document). SCORM -. URL https://scorm.com/scorm-explained/ (accessed 3.20.18).
[13] Rustici Software, 2018a. Benefits of SCORM - SCORM – (WWW Document). SCORM -. URL https://scorm.com/scorm-explained/business-of-scorm/benefits-of-scorm/ (accessed 3.20.18).
[14] Glahn, C., 2013, September. Using the adl experience api for mobile learning, sensing, informing, encouraging, orchestrating. In Next Generation Mobile Apps, Services and Technologies (NGMAST), 2013 Seventh International Conference on (pp. 268-273). IEEE.
[15] Bakharia, A., Kitto, K., Pardo, A., Gašević, D. and Dawson, S., 2016, April. Recipe for success: lessons learnt from using xAPI within the connected learning analytics toolkit. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (pp. 378-382). ACM.
[16] Long, R., Murphy, J., Newton, C., Hruska, M., Medford, A., Kilcullen, T. and Harvey Jr, R., 2015. Adapting gunnery training using the experience API. In Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).
[17] Dodero, J.M., González-Conejero, E.J., Gutiérrez-Herrera, G., Peinado, S., Tocino, J.T. and Ruiz-Rube, I., 2017. Trade-off between interoperability and data collection performance when designing an architecture for learning analytics. Future Generation Computer Systems, 68, pp.31-37.
[18] Conati, C., 2009. Intelligent Tutoring Systems: New Challenges and Directions., in: IJCAI. Presented at the Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09), Morgan Kaufmann, San Francisco, pp. 2–7.
[19] Ghali, M.A., Ayyad, A.A., Abu-Naser, S.S., Laban, M.A., 2018. An Intelligent Tutoring System for Teaching English Grammar. International Journal of Academic Engineering Research (IJAER) 2, 1–6.
[20] Paquette, L., de Carvalho, A.M., Baker, R.S., 2014. Towards Understanding Expert Coding of Student Disengagement in Online Learning. CogSci 6.
[21] Paquette, L., Baker, R.S., 2019. Comparing machine learning to knowledge engineering for student behavior modeling: a case study in gaming the system. Interactive Learning Environments 1–13. https://doi.org/10.1080/10494820.2019.1610450
[22] Aleven, V., Mclaren, B.M., Roll, O., Koedinger, K., Aleven, V., Mclaren, B., Roll, I., Koedinger, K., 2004. Toward Tutoring Help Seeking: Applying Cognitive Modeling to Meta-Cognitive Skills, in: In Proceedings of the 7th International Conference on Intelligent Tutoring Systems. Springer-Verlag.