Supporting Learning Analytics in Educational Games in consideration of Qualifications-Based Learning

Ramona Srbecky[1], Marcus Frangenberg[2], Benjamin Wallenborn[1], Matthias Then[1], Iván-José Pérez-Colado[3], Cristina Alonso-Fernandez[4], Baltasar Fernandez-Manjon[5], Matthias Hemmje[1]

Abstract: Computer and video games have established themselves in society and are increasingly finding their way into learning with so-called Educational Games (EduGame). EduGames often provides a fundamental analysis of learners' learning results, but there is currently no existing approach to map the learning results automatically to digital standardized machine-readable Qualifications. In addition, different learning providers tend to use different approaches to describe Qualifications, often in the form of free text. Therefore, it is currently impossible to compare the reached Competencies and Qualifications (CQ) across different learning providers or only manually with high effort. In this paper, a prototypical implementation for automated mapping of learners' learning results in EduGames to standardized CQ will be introduced. The paper presents the conceptual work, the subsequent prototypical implementation with the chosen Analytics Environment, Game Engine, Learning Management System, and the evaluation results.

Keywords: Qualifications-Based Learning, Learning Analytics, Game-Based Learning, Unity Engine, Educational Games, RAGE Analytics, Moodle, Qualification Based Learning Model, Learning Management System

1 Introduction and Motivation

An experiment [Be15] conducted at the Hult International Business School proves that game-based knowledge transfer has potential. In this experiment, teaching content through a video game was similarly successful as teaching directly by the teacher [Be15]. EduGames often provide only restricted analysis of learners' learning results. There is currently no existing approach to automatically map the learning results to digital standardized machine-readable Competencies and Qualifications (CQs). In the context of this research, the term CQ will be used for Qualifications. CQs consist of competencies, skills, and Proficiency Levels. If an assessment is performed and the CQ of the learners gets attested, it is a Qualification. Otherwise, learners acquire only a Competency. To prove CQ in assessments, Learning Analytics should be used. Therefore, it is currently impossible to compare the reached CQ across different learning providers or manually with high effort. To address this problem, the use of standardized CQ lends itself. The problem that there is no comparability between CQ was taken up by [Wa18] and [Th20]

[1]FernUniversität in Hagen, [vorname.nachname]@fernuni-hagen.de
[2]FernUniversität in Hagen, [vorname.nachname]@studium.fernuni-hagen.de
[3]Universidad Complutense de Madrid, ivanjper@ucm.es
[4]Universidad Complutense de Madrid, crisal03@ucm.es
[5]Universidad Complutense de Madrid, balta@fdi.ucm.es

with the Qualifications-Based Learning Model (QBLM). The QBLM allows the modeling of learning scenarios using digital (machine-readable) CQs [Th20]. Based on this model, [Th20] designed the QBL4Moodle plugin for the Learning Management System (LMS) Moodle [Mo19] to work with QBLM-based CQ in Moodle. [Th20] also addresses the connection of EduGames with Qualification-based courses in Moodle, but they do not yet enable CQ-Profiles (CQPs) for individual users [Th20]. CQPs are describing each student's personal Learning Goals (target Profile) and the Current State of attested CQs (actual Profile). As a future topic, [Th20] addresses game behavior analysis and the mapping to standardized machine-readable CQ and their assignment to learners CQPs. In this context, game behavior is the player's individual interaction in the game. This includes the way the player acts or conducts himself in the game [Sff16]. This interaction causes various data that can be measured, stored, and analyzed. Since preliminary work has already been done with Moodle to support QBLM, this LMS is chosen as the technological basis for the present work. [Ne18] has realized the first prototypical implementations for analyzing and evaluating the game behavior data in EduGames. This paper's motivation is to track and analyze game behavior and subsequently enable mapping them to QBLM-based CQ.

Several Problem Statements (PS) can be derived from the motivation mentioned above. PS1 is that the individual evaluation of EduGame events is not possible. Various Analytics Engines have been established for the analysis and evaluation of game events. "An event is represented by any action or decision that is made by the player or the game itself." [Me09] However, [Ne18] encountered the problem in that an evaluation per player was not possible with Unity Analytics [Ui21], the chosen Analytics Engine in [Ne18]. However, a separation by user profiles on the EduGame or Analytics Engine side and the LMS is a prerequisite for synchronizing results and the learner's personal CQP. PS2 is that EduGame results cannot be transferred to the CQP of an LMS. PS3 is that there is currently no mapping between game behavior and achieved CQ. This means a lack of criteria determines the achievement of CQ and Proficiency Level of CQ. The PSs mentioned above result in the following two Research Questions (RQ). RQ1: Can player individual evaluations of EduGame events be achieved via Analytics Engines be achieve? RQ2: Can a model be designed to map learning game events and CQ in a CQ model? This paper focuses on RQ2. Based on the research methodology of [Ncp90], the following Research Objectives (ROs) were derived from RQ2. This RQ considers that RQ1 is implemented. RO1 is assigned to the Observation Phase (OP). In this phase, a suitable competency model is identified that can be mapped in an LMS to be determined. For this purpose, the QBLM, in particular, will be investigated. Also, possible interfaces between Analytics Engine and LMS are considered. RO2 is assigned to the Theory Building Phase (TBP). A concept is designed that shows how the EduGame results are transferred into the CQP. But also, a concept for mapping the game events with CQs will be designed. The System Development phase (SDP) moves the concept into a prototype and is assigned to RO3. The result of the SDP is evaluated in the Evaluation Phase (EMP) in the context of a Cognitive Walkthrough (CW) [Wi13]. The EMP is assigned to RO4. Here, all RQs are considered to be evaluated because RQ2 is built on RQ1. This paper is structured

according to the ROs. This means that in the State of the Art and Technology section, the OP is described. In the Conceptual Design section, the TBP is described, and the SDP phase is presented in this paper in the Proof of Concept implementation section. In the Evaluation section, the EMP is presented. The paper concludes with a summary and an indication of future developments.

2 State of the Art and Technology

Some research projects and software systems related to the research goals have already been mentioned in the previous section. In the following, the most important are described in more detail. To transfer QBLM-based CQs to Moodle [Mo19], the FernUni Hagen (FUH) [Fe21] developed the plugin QBL4Moodle [Th20]. This plugin is the interface between Moodle and Knowledge Management Ecosystem Portal (KM-EP) [Vu20]. QBL4Moodle is used to work with QBLM in Moodle itself and map CQs created with it to the Competency approach of Moodle itself. The plugin also serves to import QBLM-based CQs, profiles, and frameworks from other systems. Currently, this is realized for the KM-EP [Th20]. The Learning Management System (LMS) used at the FUH is Moodle. This LMS already offers digital learning content at the FUH. Therefore, the already existing LMS will be used in this work.

Besides software components that facilitate tracking students' behavior during gameplay, Realizing an Applied Gaming Ecosystem Analytics Environment (RAGE Analytics) [E-17], [Fr16] offers tools for analyzing the traces and displaying the evaluation results in diagrams or graphics. RAGE Analytics will be used to generate and collect the corresponding game data analyzed regarding CQ. The Player Competence Adaption Pack (P-CAP) was developed within the RAGE project. The P-CAP includes a collection of assets that empower developers to define CQs, in-game situations for qualification assessment, and learning path in-game creation. The Competency model of P-CAP is derived from the Competence-based Knowledge Space Theory [Al99]. A Competency State is given for every player who complies with the CQP in QBLM. P-CAP's Competency Assessment Asset provides the functionality to track and analyze the learner's gameplay in the respective task situations. When the player has completed the task, the corresponding CQ is stored in his own Competency State. There is no RAGE components usage in the P-CAP, but future development is considered [Th20].

3 Conceptual Design

To address RQ2, different software components and systems are required to be combined. Moodle [Mo19] is used as the LMS, RAGE Analytics [E-17], [Fr16] as the Analytics Engine and Unity Engine [Ax16] as the Game Engine. Q2 requires a mapping to take place between the EduGame outcomes and CQ. CQs, according to the QBLM and CQP, were Moodle via QBL4Moodle of the RAGE Analytics environment is to implement an

extension that performs the mapping and stores the CQs achieved. An interface to Moodle needs to be created to access the Moodle CQ Frameworks. The acquired CQs should also be transferred from the extension to the respective Moodle CQPs. The Mapping extension is located within RAGE Analytics, as the transferred game data in the environment is already stored in a database and can be accessed per player. Only information about the CQ progress has to be transferred to Moodle. The Game Event Framework (GEF) to be implemented within RAGE Analytics is classified in the QBLM Service Distribution Model (QSM) of [Th20]. The QSM describes the communication between Moodle, augmented by extensions for CQ-based learning, and the RAGE Ecosystem. All components of the model that already exist or are yet to be implemented are considered. The State of development of the overall system consisting of Moodle and RAGE Ecosystem does not allow to implementation of the GEF in its envisioned form within the scope of this work. The lack of access to the CFs of Moodle by RAGE Analytics prevents a mapping of game events with QBLM-based CQs. The yet-to-be-implemented EduGameConnector receives game data, without which no CQP is possible. Besides, QBL4Moodle does not yet enable CQPs for individual users. Therefore, the design was adapted to map game results from RAGE Analytics to CQs that are defined independently from Moodle. In the context of RQ2, the player's interactions or results are to be accessed via the provided Representational State Transfer (REST) API [Fi00]. The mapping and subsequent analysis will not take place within the RAGE Analytics environment as initially intended. The game events are transferred to the Analytics Engine via an in-game tracker. The mapping between the game results and the CQs will be made outside of RAGE Analytics. Therefore, the mapping is done in a simplified way. When answering the multiple-choice questions, not several game decisions lead to success. The game event can only be completed with the correctly selected answer. This means that no PLs can be mapped either. Each of the multiple-choice questions will be mapped to one of the CQs before the game starts. This will be done via a Comma-separated values (CSV) file [Mi19]. The game is then started. The player's results are retrieved in real-time by a script to be created via RAGE Analytics' REST API [E-17]. Immediately after the retrieval, the received JavaScript Object Notation (JSON) [JS21] object is evaluated respecting the defined mapping. The success key expresses whether a question was answered correctly or incorrectly. The CQs achieved by answering the multiple-choice questions are written to another CSV file.

4 Proof of Concept Implementation

This section will describe the Proof of Concept implementation of the automated retrieval of the data using a Hypertext Preprocessor (PHP) [Ph21] script. The goal is to output the data in a CSV file, as this can be easily read in and further processed by Moodle. It should also be possible to map game events (here: multiple-choice questions) and CQs. A simple mapping of CQs to multiple-choice questions will take place in another CSV file. The CSV file with the contained mapping is provided to the PHP script before execution. To

bundle results and achieved CQs for further processing, the mapping of CQs to questions should be included in the CSV file that stores the game results. The desired CSV file *file.csv* for outputting the game results with assigned CQs can be generated using PHP script. The script requires the file *mapping.csv* for the assignment of CQs to multiple-choice questions. The results file *file.csv* is generated in real-time while the game is running. Below is a description of how the CSV file *file.csv* is updated only when there is new game activity. The isChanged-function reads the JSON object's timestamp and writes it to the *timestamp.txt* file. Then the function compares this timestamp with that of the most recent JSON object, sets the variable "*changed = true*", and returns the string "*Changed*" if it is not equal. The function start() runs through a while-loop with a delay of 5 seconds and updates the file *file.csv* if the timestamp changes.

5 Initial Evaluation

In the form of a CW, an initial evaluation of the Proof of Concept implementation has been accomplished by domain experts in the field of education in Computer Science. The evaluation's main goal was to estimate the productive capacity of the implementation and orientate the future development. For RQ2, a CSV file with the achieved CQs is output after a game run in the last step. It must be possible to define the mapping between game events and CQs beforehand. The mapping between the multiple-choice questions and the CQs done via the CSV file is transferred to the results file. The game's CQs can be read from the results file with the game results obtained via the REST API. For RQ1, the system components of RAGE and the related services have been configured and evaluated. No errors or misbehavior were detected. Based on the evaluation, improvements and renewals were identified for RQ2. These will be implemented and taken into account in future work.

6 Conclusion and Future Work

In this paper, a mechanism for automated mapping for CQs gained in an EduGame has been introduced. In the concept section, the system's architecture and the mapping of CQs between the EduGame and the LMS had been presented. Afterward, the initial Proof of Concept implementation and the evaluation had been described. Regarding RO2, it was determined in the TBP that, based on current research, too many limitations exist to map game events in Moodle CQPs. Therefore, the conceptual system design was adapted. A mapping that is independent of CQ Frameworks mapping between game events and CQs could be implemented. Building directly on this work's findings, a follow-up effort could address the connection of the REST API of RAGE Analytics to Moodle using the EduGameConnector component, which is yet to be developed, and CQPs. The more robust integration of EduGames or EduGame platforms in Moodle offers the opportunity to strengthen the LMS in its position as a central digital learning location and increase acceptance among learners and teachers. Moodle and the KM-EP's extensions, which are currently only conceptual, would have to be realized in future developments.

7 References

[Al99] Albert, D.; Lukas, J.: Knowledge Spaces, Lawrence Erlbaum Associates, Mahwah,1999

[Ax16] Axon, S., Unity at 10: For better- or worse- game development has never been easier, https://bit.ly/35EGJwo, 04.01.2021

[Be15] Beck, J. C., The videogame that teaches business strategy better than professors, https://bit.ly/3zFz6DO, 03.01.2021

[E-u17] E-ucm/rage-analytics, https://bit.ly/2TLMCVO, 11.11.2020

[Fe21] FernUniversität in Hagen: Startseite, https://www.fernuni-hagen.de, 07.04.2021

[Fi00] Fielding, R.T.: Architectural Styles and the Design of Network-based Software, University of California, Irvine, 2000

[Fr16] Freire, M.; Serrano-Laguna, À.; Manero, B.; Martínez-Ortiz, I.; Moreno-Ger, P.; Fernández-Manjón, B.: Game Learning Analytics: Learning Analytics for Serious Games, Learning, Design, and Technology, S. 1-29., 2016

[JS21] JSON – Einführung in JSON, https://www.json.org/json-de.html, 2021

[Ne18] Neuber, T., Bachelorarbeit - Konzeption und Realisierung einer interaktiven Spiele-Laufzeitumgebung in ein mandantenfähiges Weiterbildungsportal auf Basis einer Game-Engine und eines Learning Development Systems, Hochschule RheinMain, 2018

[Me09] Medler, B.: Generations of game analytics, achievements and high scores, Eludamos. Journal for Computer Game Culture, 3 (2), p.177-194, 2009

[Mi19] Microsoft – Outlook CSV, https://bit.ly/3qa8WEO ,14.4.2021

[Mo19] Moodle - Was ist Moodle, https://docs.moodle.org/310/de/Was_ist_Moodle, 07.01.2021

[Ncp90] Nunamaker, J. F.; Chen, M.; Purdin, T. D.: System development in Information System Research, 1990

[Ph21] PHP – Was ist PHP?, https://www.php.net/manual/de/intro-whatis.php, 2021

[Sff16] Soares de Lima, E.; Feijó, B.; Furtado, A.: Player Behavior Modeling for Interactive Storytelling in Games, SBC-Proceedings of SBGames, 2016

[Th20] Then, M.: Supporting Qualifications-Based Learning (QBL) in a Higher Education Institution's IT-Infrastructure, deposit_hagen, Hagen, Deutschland, 2020

[Ui21] Unity Analytics, https://docs.unity3d.com/Manual/UnityAnalytics.html, 7.4.2021

[Vu20] Vu, B.: A Taxonomy Management System Supporting Crowd-based Taxonomy Generation, Evolution, and Management, deposit_hagen, Hagen, Deutschland, 2020

[Wa18] Wallenborn, B.: Entwicklung einer innovativen Autorenumgebung für die universitäre Fernlehre, deposit_hagen, Hagen, Deutschland, 2018

[Wi13] Wilson, C.: User Interface Inspection Methods: A User-Centered Design Method, Elsevier Science, Niederlande, 2013