Supporting the Mapping of Educational Game Events with Competency Models considering Qualifications-Based Learning

Ramona Srbecky[a], Marcus Frangenberg[a], Benjamin Wallenborn[a], Matthias Then[a], Iván-José Pérez-Colado[b], Cristina Alonso-Fernandez[b], Baltasar Fernandez-Manjon[b] and Matthias L. Hemmje[a]

[a] *FernUniversität Hagen, Universitätsstraße 47, Hagen, Germany*
[b] *Universidad Complutense de Madrid, Av. Séneca 2, Madrid, Spain*

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, Game-Based Learning, RAGE Analytics, CbKST, Learning Analytics

1. Introduction and Motivation

Computer and video games have established themselves in society and are increasingly finding their way into learning with so-called Educational Games (EduGame). Around 30,4 million Germans play at least occasionally, including 30 percent of 10- to 29-year-olds and 30 percent of 30- to 49-year-olds [19]. Schoolchildren and students are, therefore, very familiar with the medium. Learning content presented in game form builds on these experiences. An experiment [6] 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 [6]. However, 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 (CQ). 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 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 [36] and [32] with the Qualifications-Based Learning Model (QBLM). The QBLM allows the modeling of learning scenarios using digital (machine-readable) CQ [32] [36]. Based on this model, [32] designed the QBL4Moodle plugin for the Learning Management System (LMS) Moodle [25] to work with QBLM-based CQ in

CERC 2021: Collaborative European Research Conference, September 09–10, 2021, Cork, Ireland

{ramona.srbecky; benjamin.wallenborn; matthias.then; matthias.hemmje}@fernuni-hagen.de (R. Srbecky, B. Wallenborn, M. Then, M. Hemmje); marcus.frangenberg@studium.fernuni-hagen.de (M. Frangenberg); ivanjper@ucm.es (I. J. Perez-Colado); crisal03@ucm.es (C. Alonso-Fernandez); balta@fdi.ucm.es (B. Fernandez-Manjon)

Moodle. [32] also addresses the connection of EduGames with Qualification-based courses in Moodle, but they do not yet enable CQ-Profiles (CQPs) for individual users [32]. CQPs describe each student's personal Learning Goals (target Profile) and the Current State of attested CQs (actual Profile). As a future topic, [32] 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 [31]. 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. Through the LMS, students can access the study material and learning units for each study course [7]. The study courses are a subdivision of study programs from Higher Educational Institutions and are presented in different LMS formats. [28] has realized the first prototypical implementations for analyzing and evaluating the game behavior data in EduGames. This paper aims 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." [24] However, [28] encountered the problem in that an evaluation per player was not possible with Unity Analytics [34], the chosen Analytics Engine in [28]. 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?

Based on the research methodology of [27], the following Research Objectives (ROs) were derived from the RQs. RO1 is assigned to the Observation Phase (OP). In this phase, a suitable CQ model is identified that can be mapped in an LMS to be determined. For this purpose, the QBLM 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. However, 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 (EP) in the context of a Cognitive Walkthrough (CW) [38]. Finally, the EP is assigned to RO4. In this phase, both RQs are evaluated. This paper is structured according to the ROs. This means that in the State of the Art (SOA) 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 (PoC) implementation section. In the Evaluation section, the EP 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. The work in this section represents the results of the OP of the research methodology.

2.1. State of the Art

In this section the State of the Art of the QBLM model and the Competence-based Knowledge Space Theory (CbKST) will be briefly described.

To increase the comparability of learning objects based on Qualifications, the QBL-approach described by [33] was implemented at the University of Hagen [16]. The QBLM approach includes a

Domain Model (QDM), an Architectural Model, and diverse Service Distribution Models. The QDM introduces the Qualifications-Relevant Learning Element (QRLE). A QRLE instance can de facto stand for a Personal Development Plan (for example, a study program), a Unit of Learning, a Learning Activity, or a Knowledge Resource. In QBL, CQ-based Learning Goals and Access Requirements (ARs) are represented by CQPs consisting of CQ Instances (the actual CQs). CQPs are applied to specify QRLE-related Learning Goals and ARs and describe each student's personal Learning Goal (target profile) and the Current State of attested CQs (actual Profile). After successful completion of a QRLE, the student's actual Profile is updated. The gap between the actual and the target profile must be bridged by appropriate CQ programs, including suitable QRLEs such as courses, learning activities, and knowledge resources [32].

CbKST is an extension of the original Knowledge Space Theory (KST) [3]. The KST is a mathematical theoretical framework for representing the knowledge of learners in a certain learning domain. The domain is specified by a set of problems (in the following denoted with X), which the learner is able to solve [3]. The subset of problems that the learners can solve is defined as the knowledge state of the individual learner in the KST [20]. Because of the mutual psychological dependencies between the problems, not all possible subsets of X are plausible knowledge states. One possible approach to solving these dependencies is the concept of surmise systems. Here it is assumed that if a learner can solve the problem $x \in X$, with a family of subsets of X as clauses, the learner is also capable of solving all problems in one of the clauses [3]. CbKST involves psychological assumptions about underlying skills and competencies required to solve specific problems. In the CbKST approach, competencies are mapped to learning objects (taught competencies) and tested competencies [29]. A competency state can be defined similarly to a knowledge state. The competency state consists of a set of skills which the learner possesses. Based on a learner's current competency state and learning structure, personalized learning paths can be created [20].

2.2. State of Technology

In this section, the State of Technology will be addressed in more detail. First, the Course Authoring Tool (CAT) and the LMS will be introduced, followed by a short description of the data exchange formats for learning content. Afterward, a project for tracking students' behavior in gameplay and a Competence-based Knowledge Space Theory (CbKST) [3] based tool for CQ in-game assessment will be addressed. Finally, the section concludes with a short description of an EduGame prototype.

The Knowledge Management Educational Portal (KM-EP) [35] will be further developed within the project Realising an Applied Gaming Ecosystem (RAGE) [36]. The portal offers various web-based tools for Knowledge Management and user-friendly CAT to create Moodle courses without much previous knowledge [36]. In addition, the KM-EP enables QBLM-based work with CQs. For this purpose, there is a CQ administration, the possibility to create CQPs, and to assign learning units (modules, courses, and teaching resources within courses) to these profiles. Furthermore, courses with assigned CQs can be exported to Moodle and executed there [36].

Moodle stands for Modular Object-Oriented Dynamic Learning Environment [25] and is a freely available open-source software under the GNU Public License. It is software with which courses can be conducted and developed via the internet. With the Learning Tools Interoperability (LTI) [21] standard, further applications such as games can be integrated into an LMS like Moodle [21]. To transfer QBLM-based CQs to Moodle, the University of Hagen developed the plugin QBL4Moodle [32]. This plugin is the interface between Moodle and KM-EP. QBL4Moodle is used to work with QBLM in Moodle itself and map CQs created with it to the CQ 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 [32]. The Learning Management System (LMS) used at the University of Hagen is Moodle. This LMS already offers digital learning content at the University of Hagen. Therefore, the already existing LMS will be used in this work.

To gather more information about learners' digital or non-digital learning experience, the Advanced Distributed Learning Initiative developed the Experience API (xAPI) [1]. To facilitate the compatible exchange of data about a learner's behavior and performance, the xAPI is designed as a specific standard [2]. In the context of this research, the xAPI will be used in the RAGE Analytics component to track the data from the learners [11].

The Learning Tool Interoperability (LTI) [21] standard, introduced by IMS Global Learning Consortium, integrates external learning tools and content into existing LMSs [21]. In order to use LTI, the following two systems are required to build an LTI ecosystem: A platform or LMS does not provide certain functionalities itself but delegates these functions to other tools. The second required component is the tools. These are software systems that offer digital learning games or other services that the platform does not provide but are integrated into the platform [21]. For this research, the LTI standard will be used to connect the LMS Moodle, which aims to be the platform, and the RAGE Analytics Environment, which will be the second required component, also called the tool.

The Realizing an Applied Gaming Ecosystem Analytics Environment (RAGE Analytics) has been developed at the Complutense University of Madrid, Department of Software Engineering and Artificial Intelligence [11], [18]. Besides software components that facilitate tracking students' behavior during gameplay, it offers tools for analyzing the traces and displaying the evaluation results in diagrams or graphics. Furthermore, 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 [32]. 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 CQ model of P-CAP is derived from the CbKST [3]. A CQ State is given for every player who complies with the CQP in QBLM. P-CAP's CQ 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 CQ State. There is no RAGE Analytics components usage in the P-CAP, but future development is considered [32]. The basic concept of P-CAP is shown in Figure 1.

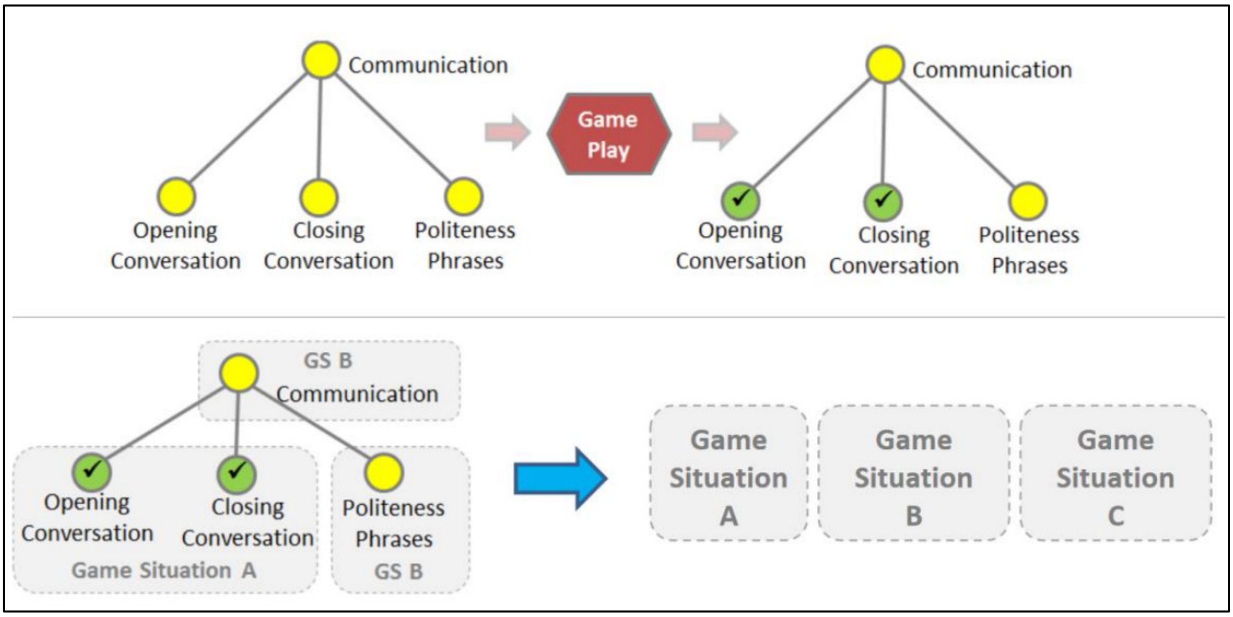


Figure 1: P-CAP Assets Overview [32]

The top-left graph in Figure 1 shows a simple CQ tree designed with the domain model asset included in the P-CAP. The CQ Communication is not considered to be achieved until the three sub-CQs, Opening Conversation, Closing Conversation, and Politeness Phrases, are satisfied. The upper right graph shows the CQ acquisition of a player after completing the game: The first two sub-CQs have been achieved, but the third (Politeness Phrases) has not. This is recorded in a so-called CQ State, which is kept for each player. This functionality is provided by the CQ Assessment Asset provided by P-CAP. The lower left graph shows the Game Situation from the example. The Opening Conversation and Closing Conversation sub-CQs were taught through Game Situation A. To teach Politeness Phrases, Game Situation B follows. Game Situations are created using the CQ Assessment Asset, and the following appropriate Game Situation is determined using the CQ-based Adaption Asset algorithms. Currently, P-CAP does not use any components from the RAGE Analytics environment to track and store game data, but this is being considered for the future [36] [32].

The Serious Game Prototype (SGP) by [28] was developed based on the Unity Engine. The Unity Engine is a multiplatform development environment for games [5]. The engine supports Windows, Linux, Android, WebGL, and current game consoles, among others [10]. The manufacturer's goal is to provide a user-friendly tool for developing virtual 3D content. After starting the SGP in the Unity editor, the player gets to the MainMenu. This event is to be referred to as opening the game. By clicking on the start button, the game begins. It offers the possibility to construct different buildings in an area. For this, raw materials are subtracted from the raw material starting values. Then, by clicking on the constructed building, the player is asked four multiple-choice questions in random order. As soon as a question is answered correctly, the player continuously receives new raw materials at intervals of seconds. The SGP developed by [28] is to be used as an EduGame in the work context.

In this section, the to-be-used software systems and approaches had been introduced. The KM-EP is used to map and store CQ Frameworks (CQF) such as QBLM-based ones quickly. Since this is currently not possible in Moodle, the QBL4Moodle plugin links Moodle and the KM-EP. Moodle should only run the related courses and provide interfaces such as LTI connection for external learning content in this approach. xAPI and Unity Engine are used for the EduGame, which an LTI connection in Moodle will provide, and RAGE Analytics and the P-CAP to analyze the player individual data regarding CQ. Since the focus of the work is not on the development of the EduGame, the existing SGP will be used as EduGame.

3. Concept

In the following, the results of the conceptual and theory-building work will be described in more detail. First, the overall concept is presented. Afterward, the use cases and the concept for a mapping component between the game events and the CQ model will be described. This is followed by the system design decisions for a modified approach for the implementation due to limitations of the current system. Subsequently, the concept is explained according to the modified system design.

3.1. Overall Concept and Use Cases

To address the RQs, different software components and systems are required to be combined. Moodle is used as the LMS, RAGE Analytics as the Analytics Engine, and Unity Engine as the Game Engine. The overall system architecture is described in Figure 2.

Moodle is to be connected to the Unity Engine and RAGE Analytics via "External Tool" activities. This allows the Unity game to be displayed directly in Moodle and enables authentication of the Moodle user. Evaluations from RAGE Analytics can also be displayed directly in Moodle, and authentication can also automatically occur here. The technical realization of the mentioned interfaces is done via LTI connections. 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 Competency progress must be transferred to Moodle.

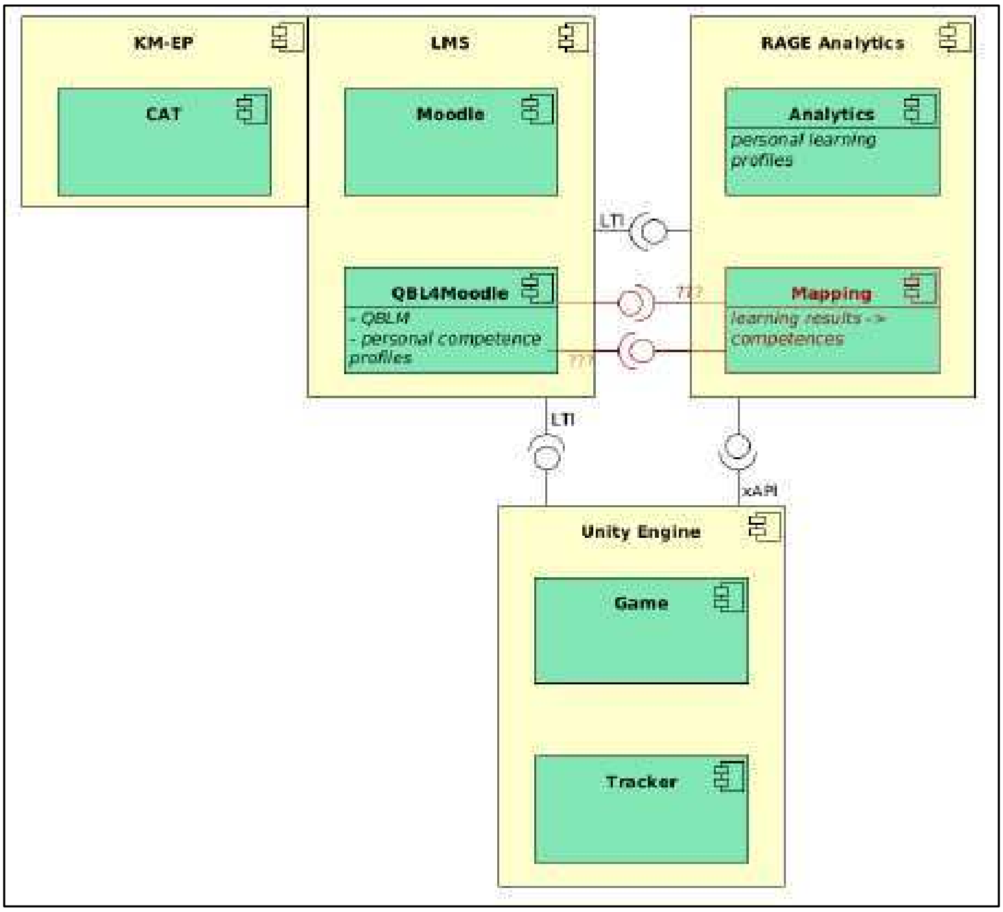


Figure 2: overall system architecture

In the following, the conceptual task assignments and use cases, as shown in Figure 3, will be addressed. This assignment is for the user roles administrator, developer, teacher, and learner. On the LMS and Analytics Engine side, the administrators are making configurational settings. This includes the setup of an External Tool [26] (via the LTI interface) on the side of the LMS to integrate the EduGame or the Analytics Engine. Also, the administrators shall perform the configuration of the Game Analytics on the Analytics Engine side. This includes assigning role permissions and managing Analytics Engine services. Developers shall create the EduGame or adapt an existing EduGame and then connect it to the Analytics Engine and from the Analytics Engine to the CQP of the LMS. Teachers should manage a course in the LMS. The creation of the course should be done in a simplified manner using a CAT and then export the course to the LMS. In the LMS, the teachers should be able to see the learners' CQP, which also considers the results of the EduGame. In the Analytics Engine, which they can call up directly from the LMS, they should be able to monitor the results of the learners. The learners should access the EduGame via a course in the LMS. The learners must be uniquely identified in the EduGame through the LMS. The learning results should be visible in the Analytics Engine or the individual CQP.

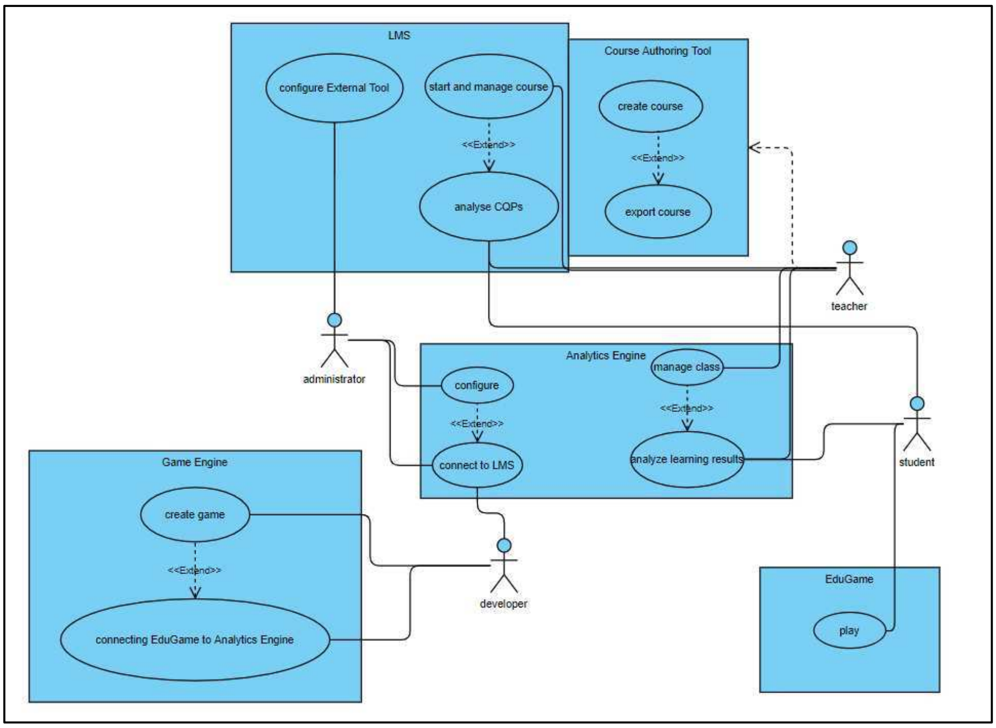


Figure 3: Use Cases

3.2. Concept for a Game Event Framework

With this Game Event Framework (GEF), game events should be defined and mapped (see Figure 2). A game event is subject to 0 to n conditions in the form of other game events. For example, event B could only occur if event A has passed before. In each game event, the player can choose from 1 to n game decisions. These decisions must be included in the dependencies of events to each other. Event B, for example, could only occur if the player had made decision A. A level is assigned to each decision in the model to capture the varying complexity of game decisions. To connect the GEF with the CQ Framework, game events must be set CQ (class Qualification in QBLM), and the game decisions resp. their levels with Proficiency Levels be linked. The GEF contains a game event. When the learner runs through this event, he can choose between three Choices. It is assumed that going through the Game Event extends CQ 1 from the CQ Framework. Therefore, the game decisions that can be made while going through the game event one should differ in their complexity and should each be assigned to one of the levels of CQ 1.

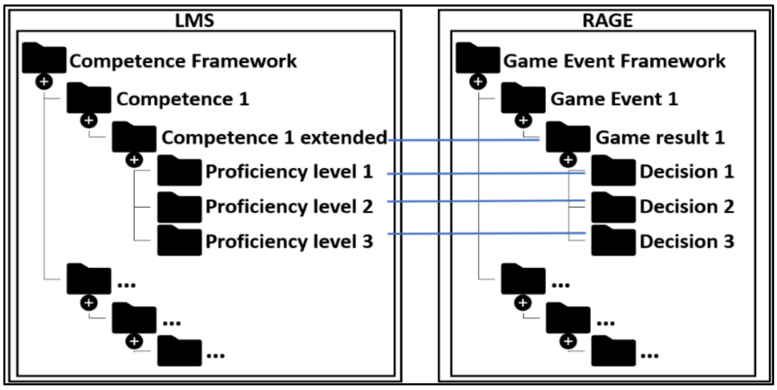


Figure 4: Mapping of the Competence Framework and GEF

Next, the GEF to be implemented within RAGE Analytics is classified in the QBLM Service Distribution Model (QSM) [32]. 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. Based on the State of development, the implementation of the GEF is assessed. A standard Moodle requires the extensions EduGameConnector and QBL4Moodle. The EduGameConnector has only been conceptually designed so far [32]. This extension is responsible for the interaction with the RAGE Analytics environment (also: Gaming Analytics Suite, GAS) and the P-CAP. The LTI interface is used to authenticate and call RAGE Analytics from Moodle. The tracker, which sends game data to the analytics server, should inform the EduGameConnector about starting games in the future. The EduGameConnector should also be able to retrieve input traces and player-specific data stored in RAGE Analytics. [32] also envisions the EduGameConnector communicating with the P-CAP. The Domain Model Asset included in the P-CAP enables the creation of game-specific CFs. The Competency Assessment Asset provides the CQ States in which achieved CQs are stored. The design of game events in which CQs can be achieved is also done via the Competency Assessment Asset. The EduGameConnector should be able to access the CQ States and retrieve the CQ Frameworks from the P-CAP.

The P-CAP does not use RAGE Analytics components to date. The RAGE Ecosystem does not yet have access to the CQ Frameworks that QBL4Moodle enables. The GEF is equivalent in position to the P-CAP in the QSM. Both models allow the definition of game events in which CQs can be achieved. However, the P-CAP does not yet use RAGE Analytics components to track and store game data. The development status of the overall system consisting of Moodle and the RAGE Ecosystem does not allow for implementing the GEF in its envisioned form within this work scope. The lack of access to CQ Frameworks through RAGE Analytics prevents mapping game events with QBLM-based CQs. The yet-to-be-implemented EduGameConnector receives game data, without which no CQP is possible. Moreover, QBL4Moodle does not yet enable CQPs for individual users [32].

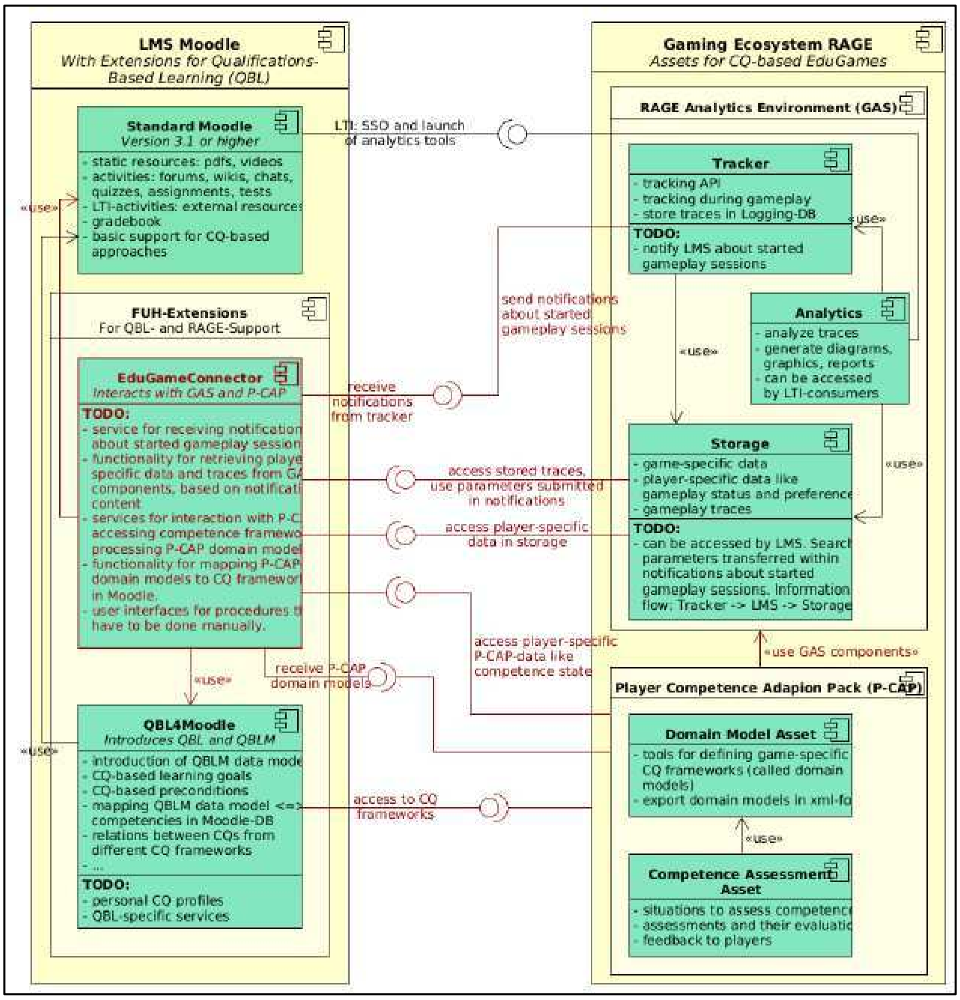

LMS Moodle
With Extensions for Qualifications-Based Learning (QBL)
Standard Moodle
Version 3.1 or higher
- static resources: pdfs, videos
- activities: forums, wikis, chats, quizzes, assignments, tests
- LTI-activities: external resources
- gradebook
- basic support for CQ-based approaches
«use»
FUH-Extensions
For QBL- and RAGE-Support
EduGameConnector
Interacts with GAS and P-CAP
TODO:
- service for receiving notification about started gameplay session
- functionality for retrieving playe specific data and traces from GA components, based on notificati content
- services for interaction with P-C accessing competence framewo processing P-CAP domain model
- functionality for mapping P-CAP domain models to CQ framewor in Moodle.
- user interfaces for procedures t have to be done manually.
«use»
QBL4Moodle
Introduces QBL and QBLM
- introduction of QBLM data mode
- CQ-based learning goals
- CQ-based preconditions
- mapping QBLM data model <=> competencies in Moodle-DB
- relations between CQs from different CQ frameworks
- ...
TODO:
- personal CQ profiles
- QBL-specific services
«use»
LTI: SSO and launch of analytics tools
receive notifications from tracker
send notifications about started gameplay sessions
access stored traces, use parameters submitted in notifications
access player-specific data in storage
access player-specific P-CAP-data like competence state
receive P-CAP domain models
access to CQ frameworks
Gaming Ecosystem RAGE
Assets for CQ-based EduGames
RAGE Analytics Environment (GAS)
Tracker
- tracking API
- tracking during gameplay
- store traces in Logging-DB
TODO:
- notify LMS about started gameplay sessions
«use»
Analytics
- analyze traces
- generate diagrams, graphics, reports
- can be accessed by LTI-consumers
«use»
Storage
- game-specific data
- player-specific data like gameplay status and preference
- gameplay traces
TODO:
- can be accessed by LMS. Search parameters transferred within notifications about started gameplay sessions. Information flow: Tracker -> LMS -> Storage
«use»
«use GAS components»
Player Competence Adapion Pack (P-CAP)
Domain Model Asset
- tools for defining game-specific CQ frameworks (called domain models)
- export domain models in xml-fo
«use»
Competence Assessment Asset
- situations to assess competence
- assessments and their evaluatic
- feedback to players

Figure 5: QBLM QSM [32]

3.3. Concept for an adapted System Design

The design will be adapted to map between CQs, and game events required for RQs to occur in the following subsection. 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 CQ Framework by RAGE Analytics prevents mapping game events with QBLM-based qualifications. 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 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 [17]. The mapping and subsequent analysis will not take place within the RAGE Analytics environment as initially intended. Instead, the game events are transferred to the Analytics Engine via an in-game tracker (see Figure 6).

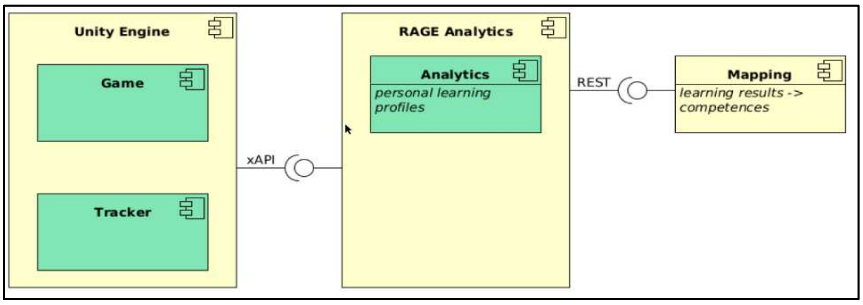


Figure 6: adapted system architecture

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 Proficiency Levels 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 [23]. In a CSV file, the data is stored in text form separated by commas and not in column form [23]. 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. Immediately after the retrieval, the received JSON 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.

The identified game situations from the prototype of [28] can be mapped with standard interactions from the xAPI Serious Games (xAPI-SG) Profile [13]. The xAPI-SG Profile represents a standard tracking format for Serious Games. It was developed to collect general and game-independent information from Serious Games [13]. Only the identified game event Answer Questions are suitable for mapping to CQ. The other events, Open Game and Start Game do not depend on player decisions. The answer questions event can be transferred, e.g., via the xAPI Game Object Completable. Answering multiple-choice questions can be sent to RAGE Analytics via the tracker's Completed method. Correctly answered questions mean a call to the Completed method with the parameter *success = true*, while incorrectly answered questions result in a call with the parameter *success = false*.

4. Proof of Concept Implementation

In the following, the Proof of Concept (PoC) implementation and the results of the SDP will be described. First, the installation and configuration of RAGE Analytics will be defined. Afterward, the integration of a tracker for RAGE Analytics into the SGP will be addressed and followed by describing the retrieval of the game events and the automated mapping between the game events and the CQ model.

4.1. Installation and Configuration of RAGE Analytics

The user must have a Linux operating system, a Docker version, and Docker Compose installed to install RAGE Analytics. With the help of Docker, it is possible to separate the infrastructure from the application so that the software can be deployed faster. For this, Docker provides the ability to package and run applications in what is called a container. A container contains everything needed to run the application [8]. Multi-container Docker applications can be defined and executed using the Docker Compose tool [9]. In a Compose file, the application's services can be configured and then started via a single command. The A2 module [12] is used to manage permissions for users and connected services. Users create their account themselves and are assigned a role by "root". Next to the predefined roles, own roles can be added. The "root" password is created during the RAGE Analytics installation. In this prototype implementation, the users "developer," "teacher," and "student" were created and assigned to the appropriate roles. The Representational State Transfer (REST) API [17] is used to access and process the data received from the Unity game. The teachers' account is used for this purpose.

It must be ensured that the teachers' account has the appropriate access to the REST API endpoints. The REST API only provides access to aggregated match data in the form of a simplified JSON file. Elasticsearch [14] provides all the received data. Elasticsearch is an open-source distributed search engine and analytics engine for data. Raw data can be received from various sources, parsed, normalized, enriched, and then indexed in Elasticsearch. The data is stored as JSON documents, and Elasticsearch stores it in what is called an inverted index. In an inverted index, each word that occurs at least once in the documents is stored to indicate its occurrence [14]. This data structure enables a fast full-text search. This work is limited to the use of the REST API since the available data is elementary, and the feasibility should be shown. Elasticsearch is also freely accessible without further security protection and would have to be secured for productive use.

4.2. Tracker Integration of RAGE Analytics into the EduGame

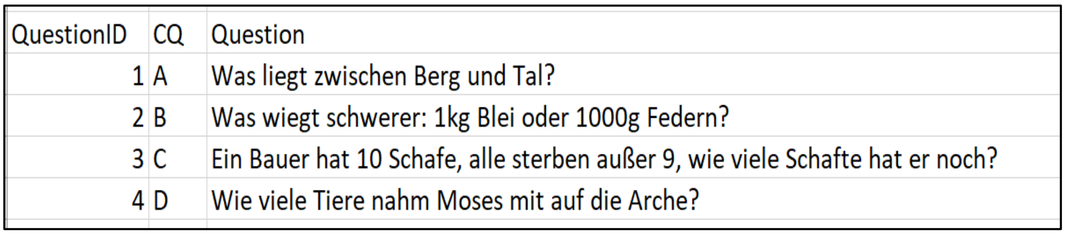
To send game data from the SGP to RAGE Analytics, a tracker must first be integrated into the game. Therefore, the unity tracker from the RAGE project is used. An empty game object is created in the Unity editor in the first scene (here: MainMenu), renamed Tracker, and linked with the script Tracker.cs from the cloned directory. The game object Tracker can now be used to define parameters for the transfer of the game data. The tracker remains in the further scenes of the game.

For the extension of the Unity scripts, statements from xAPI-SG Profile [13] are used. The game logic is mapped via scripts in the Assets/Scripts folder. The Update() method has been added to the script to capture that the MainMenu is opened. When the player returns to the MainMenu after a game round, this action must not be captured again. As mentioned above, tracking is done using predefined xAPI statements. For tracking, a so-called Accessible object can be used. This is done in the code via the method Accessed(). The game logic for answering the multiple-choice questions is in the Multiple Choice Questions class. A Completable object describes whether the question was answered correctly or incorrectly. The script's implementation is done by the method Completed(). It is possible to create custom statements within the script if xAPI Profile statements are not enough, but for this PoC implementation, custom statements had not been required.

4.3. Automated Mapping of the Data

Game data submitted to RAGE Analytics is not accessed directly through Elasticsearch but through the REST API. Required for authentication and retrieval of learning activity results are the REST API endpoints /API/login and /API/activities/:activityId/results. Access to the REST API is tested using the Postman API [30] development environment. Postman is an API development environment and available for popular operating systems. Postman covers the entire API development process by providing test servers, API monitoring, automated documentation, and management of test scenarios and data through its own API [30]. A POST request generates a token that is needed for further authentication. The account of the "teacher" is used since this person has access to the learning activity. The generated token is used for a GET request, which creates a JSON object containing the aggregated game data. This means that the number of entered screens (xAPI-Verb accessed) and the evaluation of answered questions according to true and false (xAPI-Verb completed) are displayed cumulatively.

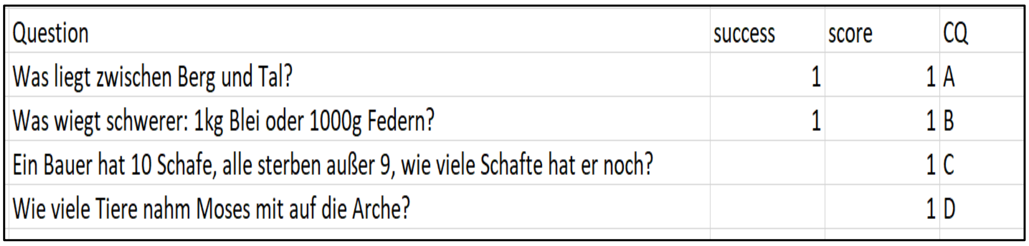
The next step is to automate the retrieval of the data using a PHP script. The goal is to output the data in a CSV file, as this can be easily read in and further processed by Moodle. It should also be possible to map game events (here: multiple-choice questions) and CQs. A simple mapping of CQs to multiple-choice questions will occur in another CSV file shown in Figure 7. The CSV file with the contained mapping is provided to the PHP script before execution.



QuestionID	CQ	Question
1	A	Was liegt zwischen Berg und Tal?
2	B	Was wiegt schwerer: 1kg Blei oder 1000g Federn?
3	C	Ein Bauer hat 10 Schafe, alle sterben außer 9, wie viele Schafte hat er noch?
4	D	Wie viele Tiere nahm Moses mit auf die Arche?

Figure 7: Assignment of CQs to questions

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 a PHP script (see Figure 8).



Question	success	score	CQ
Was liegt zwischen Berg und Tal?	1	1	A
Was wiegt schwerer: 1kg Blei oder 1000g Federn?	1	1	B
Ein Bauer hat 10 Schafe, alle sterben außer 9, wie viele Schafte hat er noch?		1	C
Wie viele Tiere nahm Moses mit auf die Arche?		1	D

Figure 8: Output of game results with assigned CQs

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. The functionality of the script is briefly explained below. The login function returns the token for authentication. Via getRequest-function, a GET request is made at the endpoint URL. Authentication is done with the passed parameter token. The getResult-function is the heart of the script. The script first executes the two explained functions and writes the JSON Object to the CSV file file.csv. It is essential to mention that a multiple-choice question is only in the JSON Object if the player has answered it (correctly or incorrectly). Afterward, the CSV file mapping.csv is read, the (answered) questions in file.csv are read via the questionID. The correct answers are identified, and the corresponding CQ is also written to the file.csv file. With the functions described so far, it is possible to query the data via the REST API by manually executing the PHP script. 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.

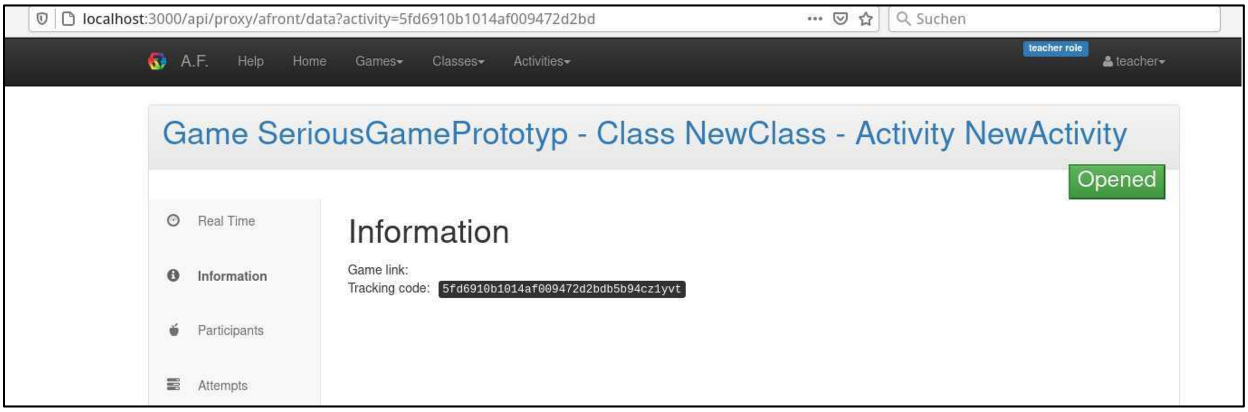
5. Initial Evaluation

In the form of a CW [38], an initial evaluation of the PoC implementation has been accomplished by domain experts in computer science education. The evaluation's main goal was to estimate the productive capacity of the implementation and orientate the future development. During the CW, the following four tasks have been accomplished and evaluated:

1. Configuration of RAGE Analytics
2. Playing the EduGame
3. Evaluation of the EduGame outcomes visualization in RAGE Analytics
4. Mapping game events and CQ model

All four tasks are derived from the RQs mentioned in the first section of this paper. In the following four tasks, their goals and the evaluation results of the EP will be addressed.

The goal of the first task is to create the prerequisites in RAGE Analytics for the successful connection and evaluation of an educational game. This includes settings as well as the visualization of the data. First, the system components of RAGE and the related services have been configured and evaluated. Then, the prerequisites for the connection and evaluation of an EduGame could be created. Finally, a learning activity was successfully created, and a tracking code for the Unity game was generated, as shown in Figure 9, with it. Upon configuration, a default dashboard is displayed in RAGE Analytics to visualize the data received from the tracker. The dashboard can be customized with the "developer" role using Kibana [22]. Using "Edit" in Kibana, the dashboard is customized and saved to include only the AccesibleAccesedAllperTargetId, CompletablesCompletedperTarget, and xAPIVerbsActivity panels. This is done to be able to visualize the Accessibles (screens entered), Completables (questions answered) and xAPIVerbs (accessed, completed, testverb) used in the game according to the frequency of their use.



localhost:3000/api/proxy/afront/data?activity=5fd6910b1014af009472d2bd

Game SeriousGamePrototyp - Class NewClass - Activity NewActivity

Information

Game link:
Tracking code: `5fd6910b1014af009472d2bdb5b94cz1yvt`

Figure 9: RAGE Analytics Frontend: newly created Learning Acitivity and corresponding Tracking Code

The second subtask aimed to run through the Unity game, which was extended by a tracker, without errors or aborts. Since the integration of the EduGame in Moodle as an External Tool was not implemented as part of the PoC implementation, the EduGame is started directly from the Unity environment. For this, the corresponding project is selected with the "Learner" role in Unity. The game is started by the user via "Play" and the players must enter their named manually (see Figure 10).

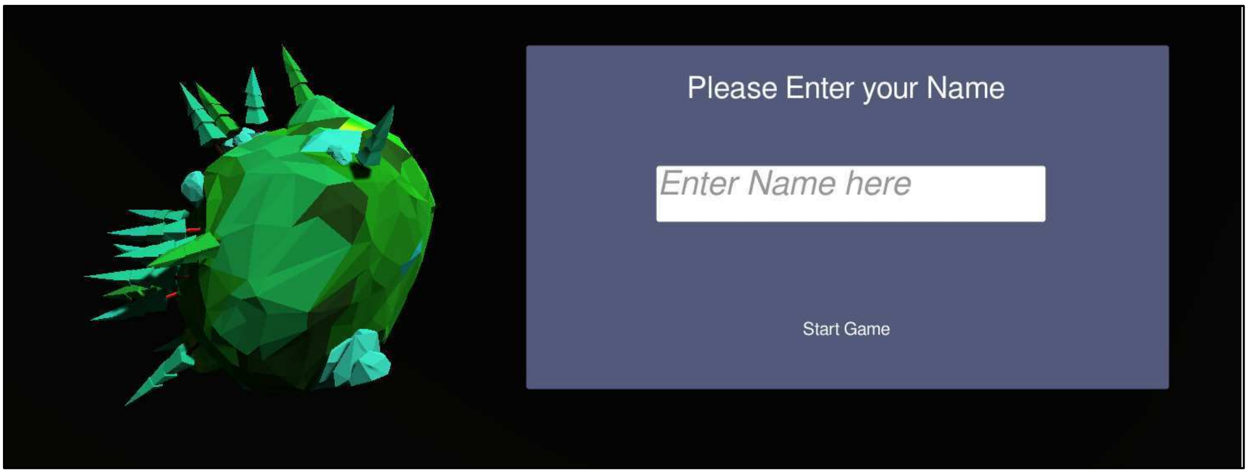


Figure 10: EduGame start screen

The game area includes a terrain, a display of raw materials (Wood, Stone) and three buttons for the construction of buildings (Woodfarm, Stonefarm, Tavern). The learner creates a first building with a left click on the Woodfarm button and another left click on the terrain (See Figure 11).

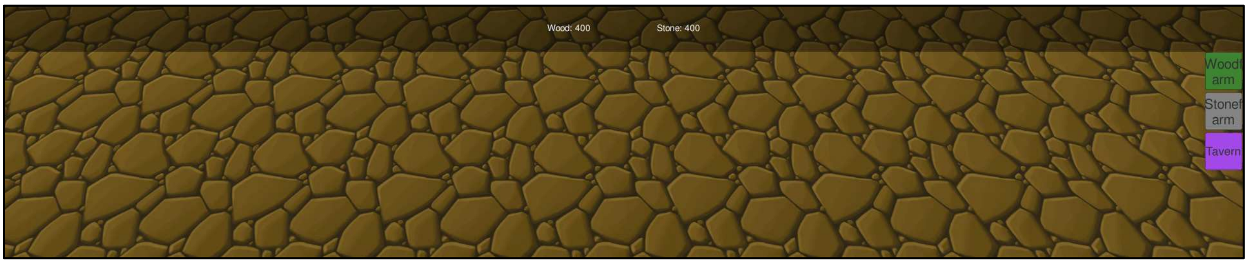
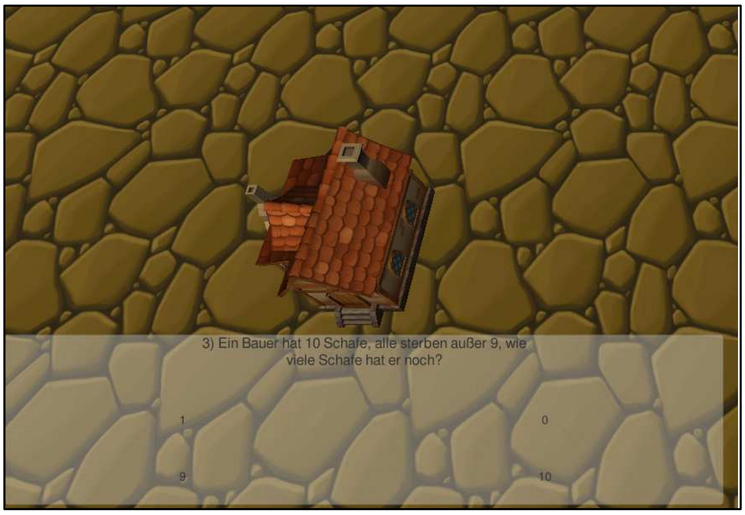


Figure 11: empty game interface of the SGP

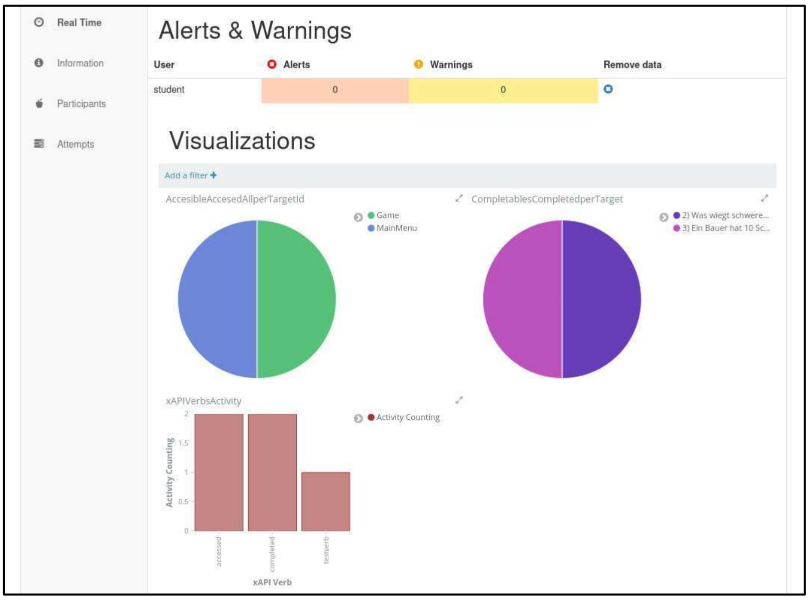
The construction has cost raw materials, the new stock is displayed. After a left click on the Woodfarm a multiple-choice question appears. The learners answer the multiple-choice question (see Figure 12) by left clicking with the answer and receives a "Correct" as confirmation for the correct answer or "wrong" for an incorrect answer. If the answer is correct, the raw material Wood is continuously increased by two units. After a left click on the Woodfarm another multiple-choice question appears. The learners can end the game by clicking the pause button in the toolbar. The game has been played and evaluated if any errors or misbehaviors occurred. No errors or terminations were detected.



3) Ein Bauer hat 10 Schafe, alle sterben außer 9, wie viele Schafe hat er noch?

Figure 12: SGP game interface with a multiple-choice question displayed

The aim of the third subtask was to successfully evaluate the game data transmitted by the Unity game in the RAGE Analytics frontend. As a result, the dashboard in RAGE Analytics correctly maps student's game results (See Figure 13).



Alerts & Warnings

User	Alerts	Warnings	Remove data
student	0	0	○

Visualizations

Figure 13: RAGE Analytics dashboard

The object of the fourth subtask is to obtain a CSV file with the achieved CQs after a game run. The mapping between game events (here: multiple choice questions) and CQs must be defined previously. The mapping between the multiple-choice questions and the CQs done via the CSV file is transferred to the results file. Together with the game results obtained via the REST API, the game's CQs can be read from the results file. 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. These will be implemented and considered in future work.

6. Conclusion and Future Work

In this paper, a mechanism for automated mapping for CQs gained in an EduGame and a player-specific evaluation of EduGame events have 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 PoC implementation and the evaluation had been described. In the PoC implementation, a tracker was included in the Unity game, and the game scripts were adapted so that game events could be sent to RAGE Analytics. With the dashboard in RAGE Analytics, the game events could be successfully displayed. Therefore, RQ1 "Can player individual evaluations of EduGame events be achieved via Analytics Engines be achieve?" is addressed.

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. Therefore, RQ2 "Can a model be designed to map learning game events and CQ in a CQ model?" is addressed.

Based on the evaluation, improvements and renewals were identified. The renewals and improvements found can be directly implemented as improvements from the CW. One of them is the easier usability of the RAGE Analytics dashboard. An improvement identified in the CW is that the dashboard in RAGE Analytics should be configurable to fit the game events used in the EduGame. Currently, the interface is only understandable for expert users and not for teaching staff. For this purpose, a configuration tool could be considered, which would allow configuring the dashboard in RAGE Analytics easily. A second direct 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. Also, a follow-up effort could address user authentication between the Moodle, RAGE Analytics, and Unity Engine systems. It would be desirable if a user login to Moodle were sufficient to access RAGE Analytics and legitimize themselves in the Unity game.

Future investigation and evaluations need to be done in the case of the implementation of the EduGameConnector and P-CAP. Also, evaluations and concepts regarding the usage of CbKST based algorithms for the analysis of CQs and the certification of CQs. Since right now the prototype does not analyze the in-game behavior regarding CQs. Therefore, the projects Adaptive Learning with Knowledge Spaces (ALEKS) [4] and Enhanced Learning Experience and Knowledge Transfer (ELEKTRA) [15] are used as references and analyzed to what extent the algorithms and concepts used there can be used.

7. References

[1] ADL Initiative U.S. Department of Defense: Project xAPI github, 2013. URL: https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-About.md#partone.

[2] ADL Initiative U.S. Department of Defense: Project xAPI, 2020. URL: https://adlnet.gov/projects/xapi/#resources.

[3] D. Albert, J. Lukas. Knowledge Spaces, Lawrence Erlbaum Associates, Mahwah, 1999.

[4] ALEKS: ALEKS 2021. URL: https://www.aleks.com/.

[5] S. Axon: Unity at 10: For better- or worse- game development has never been easier, 2016. URL: https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier.

[6] J. C. Beck: The videogame that teaches business strategy better than professors, 2015. URL: https://qz.com/526534/the-video-game-that-teaches-business-strategy-better-than-professors/.

[7] C. Dalsgaard, Social software: E-learning beyond learning management systems in European Journal of Open, Distance and E-Learning, volume 9, 2016.

[8] Docker: Docker 2021. URL: https://docs.docker.com/get-started/overview/.

[9] Docker Compose: Docker Compose, 2021. URL: https://docs.docker.com/compose/.

[10] R. Dillet: Unity CEO says half of all games are built on Unity, 2018. URL: https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/?guccounter=1&guce_referrer=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw&guce_referrer_sig=AQAAAJzvctT-NWpjAyurVPYHBip0WcEuFQRo61nbUIsB7Yvr172go-9Chw4oIiaxmpNssMmFa_n-Y9hE2kbpPpje.

[11] RAGE Analytics Overview: RAGE Analytics Overview, 2017. URL: https://github.com/e-ucm/rage-analytics/wiki/Rage-analytics-Overview.

[12] RAGE A2: RAGE A2 overview, 2018. URL: https://github.com/e-ucm/rage-analytics/wiki/A2-Overview.

[13] RAGE xAPI SG Profile: RAGE xAPI SG Profile, 2019. URL: , https://github.com/e-ucm/rage-analytics/wiki/xAPI-SG-Profile.

[14] Elasticsearch: Elasticsearch – was ist Elasticsearch?, 2021. URL: https://www.elastic.co/de/what-is/elasticsearch.

[15] ELEKTRA: ELEKTRA, 2021. URL: , https://cordis.europa.eu/project/id/027986.

[16] FernUniversität in Hagen: FernUniversität in Hagen, Germany's State Distance-Learning University, 2021. URL: https://www.fernuni-hagen.de.

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

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

[19] Game-Verband der deutschen Games-Branche e.V.: Jahresreport 2020, 2020. URL: https://www.game.de/wp-content/uploads/2020/08/game-Jahresreport-2020.pdf.

[20] J. Heller, B. Mayer, D. Albert, Competence-based Knowledge Structures for Personalised Learning in 1st International ELeGI Cinference on Advanced Technology for Enhanced Learning, Vico Equese, Italy, pp.8. ffhal-00190482f, 2005.

[21] IMS Global Learning Consortium: Learning Tools Interoperability, 2021. URL: https://www.imsglobal.org/activity/learning-tools-interoperability.

[22] Kibana: Kibana, 2021. URL: https://www.elastic.co/de/kibana.

[23] Microsoft: Outlook CSV, 2019. URL: https://support.microsoft.com/de-de/office/erstellen-oder-bearbeiten-von-csv-dateien-zum-importieren-in-outlook-4518d70d-8fe9-46ad-94fa-1494247193c7#:~:text=Eine%20CSV%2DDatei%20(durch%20Kommas,CSV%2DDateien%20durch%20Kommas%20getrennt,14.4.2021.

[24] B. Medler, Generations of game analytics, achievements and high scores in Journal for Computer Game Culture, Eludamos, version 3(2), p.177-194, 2003.

[25] Moodle.org: Moodle open-source learning platform, 2021. URL: https://moodle.org.

[26] Moodle.org: External Tool, 2021. URL: https://docs.moodle.org/311/en/External_tool.

[27] J.F. Nunamaker, M. Chen, T.D. Purdin, System development in Information System Research in Twenty-Third Annual Hawaii International Conference on System Sciences, pp. 631-640 vol.3, doi: 10.1109/HICSS.1990.205401, 1990.

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

[29] A. Nussbaumer, M. Maurer, S. Malicet, C. M. Steiner, D. Albert, A novel approach and software component for supporting competence-based learning with serious games in INTED2019 conference, 2018.

[30] Postman: Postman, 2021. URL: https://www.postman.com/.

[31] E. Soares de Lima, B. Fejió, A. Furtado, Player Behavior Modeling for Interactive Storytelling in Games in SBC Proceedings of SBGames, 2016.

[32] M. Then, Supporting Qualifications-Based Learning (QBL) in a Higher Education Institution's IT-Infrastructure, PhD thesis, FernUniversität in Hagen, 2020. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001608, 18 June 2021. DOI: 10.18445/20200309-141118-0.

[33] M. Then, B. Wallenborn, M. Fuchs, M. Hemmje, Towards a Domain Model for Integrating Competency Frameworks into Learning Platforms in Formamente – International Research Journal on Digital Future, issue 2016/3-4, p. 15-49, 2016.

[34] Unity: Unity Analytics, 2021. URL: https://docs.unity3d.com/Manual/UnityAnalytics.html.

[35] B. Vu, Taxonomy Management System Supporting Crowd-based Taxonomy Generation, PhD thesis, FernUniversität in Hagen, 2020. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001613, 18 Jun 2021. DOI: 10.18445/20200404-144028-0.

[36] B. Wallenborn, Entwicklung einer innovativen Autorenumgebung für die universitäre Fernlehre, PhD thesis, FernUniversität in Hagen, 2018. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001428, 18 Jun 2021. DOI: 10.18445/20180911-091907-0.

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