QBL – A Software-Technical Approach for Supporting Competence-Based Learning

Matthias Then, Benjamin Wallenborn, Felix Fischman, Sebastian Lothary, Ramona Srbecky, Michael Winterhagen, Matthias Hemmje

FernUniversität in Hagen, Germany

Abstract

Qualifications-Based Learning (QBL) is a software-technical approach that takes up former research & development activities in the area of *Competence-Based Learning (CBL)* and proposes improvements. The QBL domain model was conceived with the goal to support standardized qualifications frameworks as well as institution-specific catalogues designed for internal study or training programs. Associations between qualifications from different frameworks/catalogues advance the emergence of comprehensive qualifications networks and thereby improve the cross-institutional comparability of qualifications-based programs and learning content. Furthermore, the informative value of personal qualifications profiles is increased. In addition to the domain model, QBL proposes an architectural model for the higher education sector and introduces it on the example of an exemplary institution's (FernUniversität in Hagen) IT-landscape. The application scenarios described in this paper refer to that environment. Several proof-of-concept prototypes were developed, for example, a plugin for the learning management system Moodle, a management component for qualifications frameworks, and an authoring toolkit for qualifications-based programs and learning content. As a consequence of the positive evaluation results, subsequent QBL-related projects and theses were initiated. This paper gives an overview of QBL concepts, prototypes, and current projects.

Keywords [1]

qualifications-based learning, competence-based learning, QBL, CBL, CBL software, competence model

1. Introduction

The development of the *Qualifications-Based Learning (QBL)* approach was triggered by the observation that the idea of *Competence-Based Learning (CBL)* is increasingly finding its way into the teaching/learning processes at educational institutions, which goes along with a growing demand for CBL software solutions. This demand is only partially met by existing approaches, models and software systems, there is still much need for improvements. QBL aims to contribute concepts and tools realizing CBL visions such as: creation of courses and study programs, design and implementation of teaching/ learning scenarios, availability of user profiles, and cross-institutional comparison of competence-related information such as *learning goals* and *access requirements* for courses, modules, study programs, and learning content. QBL takes up former research & development activities, proposes improvements and provides extensions.

The term QBL was introduced, because in CBL the term competence is not always clearly distinguished from other qualification types such as skill or knowledge. In the following, the term *Competence/Qualification (CQ)* will be used as a generic notation for qualifications of any kind.

CERC 2021: Collaborative European Research Conference, September 09–10, 2021, Cork, Ireland
[firstname.lastname]@fernuni-hagen.de

In 2014, when the development of QBL was initiated, CQ-based learning goals and access requirements of study programs, modules, courses, and learning content were usually described in form of free text. Regarding comparability, this is a serious problem, because such specifications leave too much room for interpretation. The usage of standardized *CQ Frameworks (CQF)* had not yet prevailed, not least because they were not sufficiently supported by learning platforms and other software systems. In addition, only a few universities and enterprises offered institution-specific CQ catalogues. Furthermore, the integrability of CQ-based concepts into higher education institution's IT landscapes and the data exchange between the involved software components was not yet sufficiently researched. From these observations, general *research goals* were derived:

RG1: investigation of CBL from a software developer's perspective. This includes: existing approaches, available software support, applicability in distributed architectures, data flows, and combinability with common exchange formats.

RG2: design of a general *QBL Model (QBLM)* that effectively supports CQFs and CQ-related data flows within a higher education institution's IT-landscape. The QBLM includes a domain model, an architectural model, and a service model.

RG3: development of proof-of-concept prototypes implementing the QBLM domain model.

RG4: realization of distributed QBL application scenarios within an exemplary institution's IT-landscape.

This paper outlines the current state of QBL concepts, prototypes, and currently running further developments. At the beginning (chapter 2), the results of the investigation required by RG1 are briefly summarized. After that (chapter 3), the modeling activities associated with RG2 are described, followed by (chapter 4) a summary of the prototypical solutions that emerged from RG3 and RG4. As a consequence of the positive evaluation results, subsequent QBL-related projects and theses were initiated, an overview is given in chapter 5.

2. State of the Art in Science and Technology

In the following, a few approaches, technologies, and software systems are described that can be regarded as basics and building blocks for QBL.

CQ-based comparison of study programs, modules, courses, and learning content demands an appropriate selection of available CQs. In this context, an overview of standardized CQFs, institution-specific CQ catalogues, and domain taxonomies in the area of information and communication technology has been obtained. The *European Qualifications Framework* [1], a widely recognized template for designing concrete, domain-specific CQFs, is of particular importance. Its proficiency levels can be regarded as an EU-wide basis for categorizing proficiency. Another milestone is the *European e-Competence Framework* [2], an implementation of the European Qualifications Framework focusing on the sector of information and communication technology.

A research & development project with similar objectives to QBL was *TENCompetence* (official title: Building the European Network for Lifelong Competence Development), it is described in [3], chapters 1, 18 und 19. Like QBL, the TEN-Competence approach recommends the usage of standardized formats, for example, its way of modeling study programs and learning content is strongly oriented towards *IMS Learning Design* [4]. CQ-related data then are added to these structures, an overview of the resulting domain model can be found in [5]. The concepts that emerged from TENCompetence, especially the mentioned domain model, are a suitable basis for QBL, so QBL is conceived as a further development of TENCompetence.

A useful means for conceiving distributed teaching/learning scenarios is the *IMS Learning Tools Interoperability (LTI)* specification [6] which can be applied to embed externally hosted, access-protected resources into courses provided by *Learning Management Systems (LMS)*. Such resources can be, for example, assignments, tests, videos, learning programs, or educational games. This is achieved on the basis of standardized services; the LTI basic services are: single sign on, tool launch, and return of outcomes.

In many cases, educational institutions use LMSs for designing and processing their teaching/learning scenarios, so a software solution is required that introduces QBL functionality into a common LMS. The decision on a suitable basis system was made in favor of *Moodle* [7] (Moodle: Modular Object-Oriented Dynamic Learning Environment), a popular, freely available open-source LMS. Moodle provides an easily extendable plugin architecture, capable APIs, and a large number of extension points. Furthermore, it already offers comprehensive support for CQ-based scenarios, even though there is still need for improvement. Another argument for Moodle is the fact that it is the default LMS at the *FernUniversität in Hagen (FUH)* [8], where most of our QBL application scenarios are located.

For research projects, FUH's productive IT environment is only available to a limited degree. For example, the central CQ management component has to be implemented separately from FUH's campus management system, so a temporary solution is required. Such prototypes can be implemented on the basis of the *Knowledge Management Ecosystem Portal (KM-EP)* [9], an educational ecosystem focusing on research and development activities in the areas of learning content creation and knowledge management. The further development of the KM-EP is accompanied by FUH's chair of *Multimedia and Internet Applications* [10].

In the following, concepts, models, and software solutions are presented that close the gap between state-of-the-art technologies and QBL requirements.

3. Conceptual Design of the QBL Model

This chapter introduces the QBLM and its components. As required by RG2, it includes a *Domain Model (QDM)*, an *architectural model*, and diverse *service distribution models*, each of them referring to a concrete application scenario.

3.1. QBLM Domain Model

The QDM is shown in the class diagram in Figure 1 (Fig. 54 in [11]). It can be regarded as a further development of the TENCompetence domain model [5]. The blue, red, orange, and gray colored elements have been derived from TENCompetence, modifications and extensions contributed by QBL are drawn in green color. Teaching/learning content is represented by blue classes, the orange ones stand for students' actions, goals, and received gradings. The major part of QBL-specific extensions refers to the competence model represented by the gray classes.

In compliance with TENCompetence and IMS Learning Design, QBL interprets online courses as *units of learning* composed of *learning activities* and *knowledge resources*. Each of these elements contributes to the course's CQ-based learning goal. Learning activities and knowledge resources are regarded as independent elements defining their own learning goals and access requirements, which facilitates both the modular design of CQ-based courses and the definition of processing sequences (i.e., CQ-based *learning paths*). For this reason, the QDM introduces the *Qualifications-Relevant Learning Element (QRLE)*. In Figure 1, it is visualized by an abstract class that is implemented by the blue classes, so 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 are represented by *CQ profiles* consisting of *CQ instances* (the actual CQs). To achieve maximum comparability of CQ-related data, QBL recommends consequent application of standardized CQFs and institution- or domain-specific CQ catalogues. On the other hand, the creation of innovative didactic scenarios often demands additional course-, module-, or program-specific catalogues, so the QDM has to be flexible enough to handle both variants. This is achieved by the class *CQ framework* and its associated elements: from a software-technical point of view, both standardized and individually created CQ catalogues are CQFs. To enable relations between CQs from different CQFs and connections to elements of domain taxonomies, a tagging mechanism was introduced via the classes *simple tag* and *semantic tagging object*.

CQ profiles are not only applied for specifying QRLE-related learning goals and access requirements, but also for describing 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 has to be bridged by appropriate CQ programs, which include suitable QRLEs such as courses, learning activities, and knowledge resources. The personal learning goal is achieved as soon as the actual profile is equal to the target profile.

In [11], chapters 3.2, 4.1, and 5.2, the QDM, its elements, and their added value are explained in detail. This also includes the description of advanced concepts that have not been mentioned here, e.g., the distinction between *CQ scopes* and CQ instances, the revised proficiency concept, and the definition of completion criteria for attesting CQs to students.

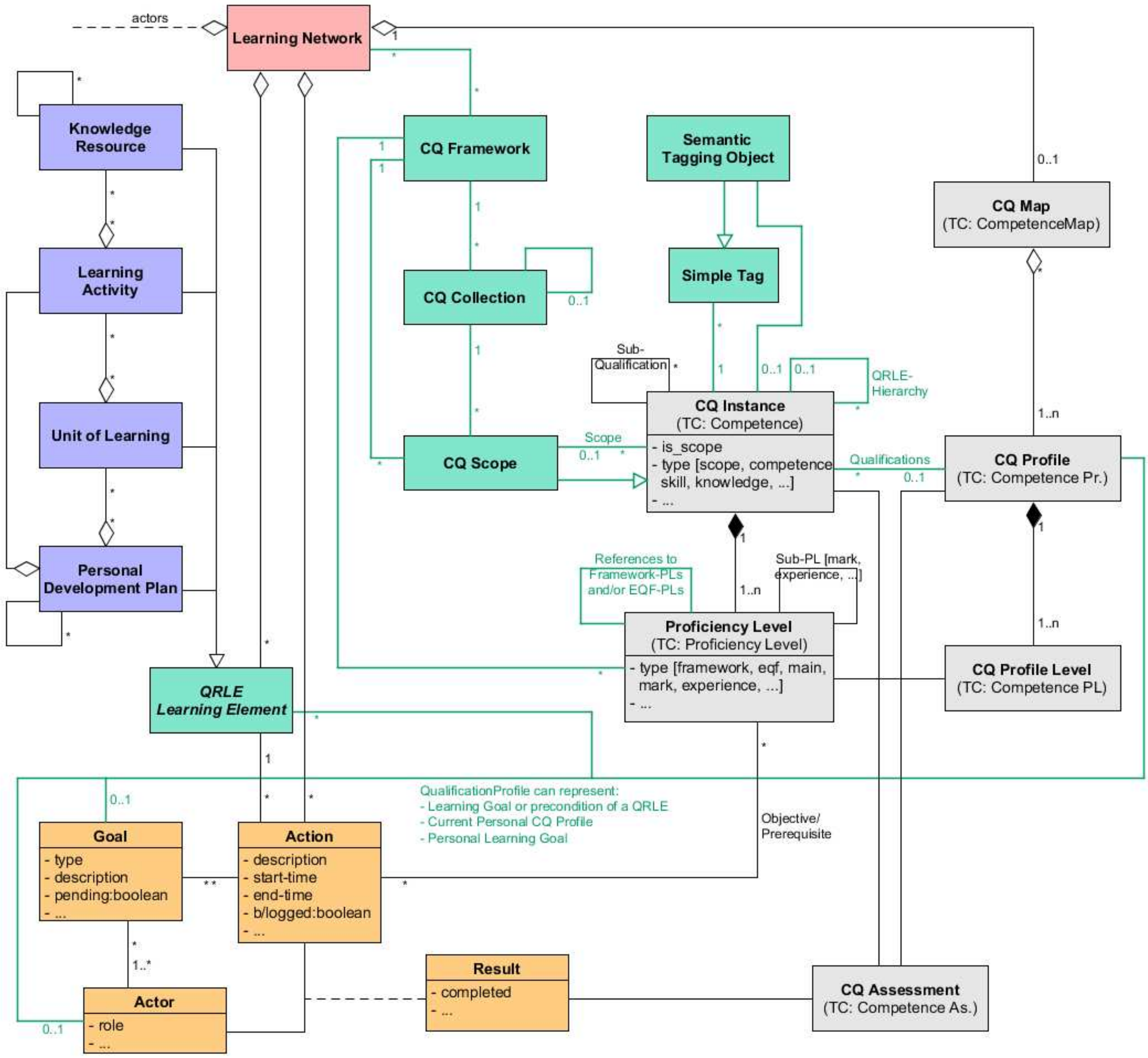

Figure 1: QBLM Domain Model (QDM, Fig. 54 in [11])

3.2. QBLM Architectural Model

The architectural model focuses on the introduction of QBL into a higher education institution's IT-landscape and identifies involved *application areas* and *software components*. Currently, it consists of a general version and a concrete implementation representing the system landscape at the FUH. In both cases, the as-is state was modeled at the beginning and later extended by additional components and relations required for QBL. The application scenarios outlined in the following refer to the FUH-specific version in the target state, which is displayed in Figure 2 (Fig. 64 in [11]). The yellow structures stand for application areas, software components are represented by green elements; the highlighted ones (light green, red border) and their interactions (red relations, dashed) are directly involved into our application scenarios. For more detailed information see [11], chapters 2.5 and 3.3.

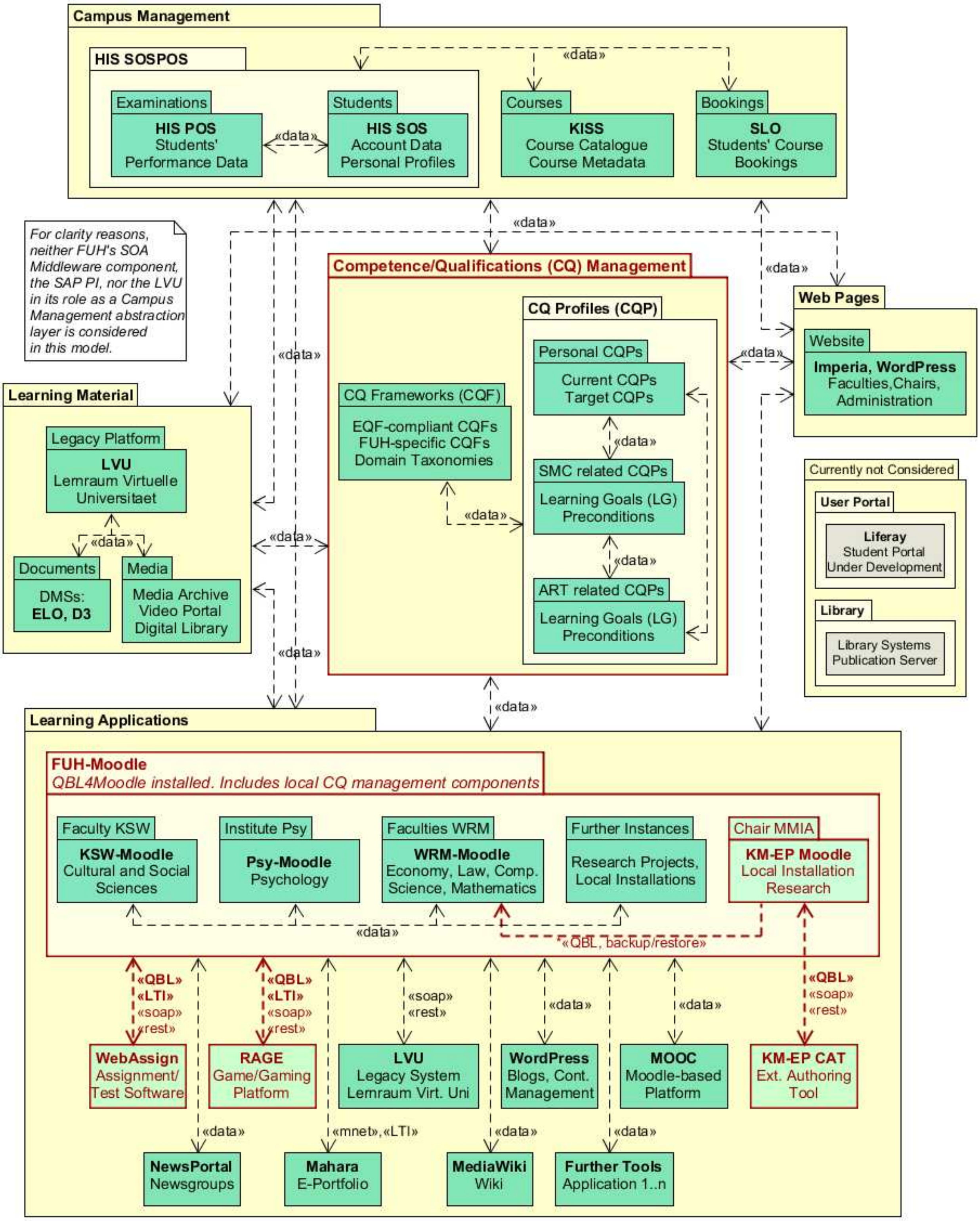

Figure 2: QBLM architectural model, FUH-specific version (Fig. 64 in [11])

In all visualizations of the architectural model, service-based interactions are indicated by relations labeled <data>. In some cases, especially when the relations are used in QBL application scenarios, they have been further concretized by service distribution models.

3.3. QBLM Application Scenarios and Service Distribution Models

In the following, these *Application Scenarios (AS)* are briefly outlined and links to the corresponding service distribution models and software prototypes are given.

AS1 is concerned with the creation of CQ-based, QDM-compliant courses. As demanded by RG3, the required QBL functionality is provided by an LMS extension, the LMS of choice is Moodle (see chapter 2). The resulting Moodle-plugin *QBL4Moodle* is documented in Then's PhD thesis [11], chapter 4.1, and in [12].

The idea of **AS2** is to extend the KM-EP with a CQ-based authoring toolkit that includes management components for QDM-compliant CQs/CQFs as well as tools for creating QRLEs with CQ-based learning goals and access requirements. Furthermore, an export mechanism for transferring such QRLEs to Moodle is required (this also demands QBL4Moodle extensions), as well as a study portal for students. AS2 is covered by Wallenborn's PhD thesis [9], for a service distribution model referring to Figure 2 see [11], chapter 3.3.5, Fig. 66.

AS3 focuses on the standardization of service-based interactions between e-learning systems, the goal is to connect Moodle and *WebAssign* [13], FUH's default application (besides Moodle) for exercises and tests, in an LTI- and QDM-compliant way. AS3 is called the *LTI-based Moodle-WebAssign-Integration*, detailed descriptions of this application scenario and the resulting software solution can be found in [11], chapters 3.3.6 and 4.2.

AS4 adapts the approach from AS3 for CQ-based *educational games*. Initial considerations and pre-projects for the *LTI-based Moodle-EduGame-Integration* are described in [11], chapters 3.3.7 and 4.3. Further developments are in progress.

4. Prototypical Implementations

In this chapter, a few prototypical implementations are outlined that emerged from the above-mentioned application scenarios.

The Moodle-plugin ***QBL4Moodle*** originally refers to AS1, but it also plays an important role in many other application scenarios. It is documented in [11], chap. 4.1, and in [12]. QBL4Moodle consequently uses the APIs and extension points offered by Moodle, modifications of the core code have been avoided as far as possible. The functional scope covers QBL basic functionalities such as:

- Support of standardized, institution-specific, and individually designed CQFs;
- Definition of CQ-based learning goals and access requirements for QRLEs;
- Assignment of achieved CQs to students;
- Realization of personal CQ profiles.

Subsequent application scenarios require extensions, for example, regarding the exchange of CQ-related data with the KM-EP, WebAssign, and educational games.

The ***CQ-based Authoring Toolkit for the KM-EP (KM-EP-CAT)*** was conceived and implemented in the context of application scenario AS2, for details see [9]. Figure 3 (Fig. 62 in [9]) gives an architectural overview of the KM-EP components that are involved into CQ-related processes and, therefore, were either introduced or modified in the context of AS2. In addition, the LMS Moodle is displayed, because it is the target system for course execution and therefore has to interact with the KM-EP-CAT. Before AS2 had been conceived, the KM-EP did not offer any support for CQ-based approaches, so a *CQ manager* for defining and modifying QDM-compliant CQs and CQFs was added, as well as a *CQ profile manager* for CQ profiles. The existing *course authoring tool* was extended with functionalities enabling authors to assign CQ profiles to courses and use them as learning goals and access requirements. Via QBL-specific web services provided by QBL4Moodle, such courses can be exported to Moodle where the teaching/learning process takes place. The QRLE types study program and module are supported by the *study program authoring tool*. It offers functionalities for creating/managing study programs and their associated modules which are usually composed of courses. All mentioned QRLE types can be associated with CQ-based learning goals and access requirements that are represented by CQ profiles created with the CQ profile manager. Besides these QRLE-related tools, a *study portal* is provided. In this portal, students can create and manage their accounts, enroll into programs, modules, and courses, and monitor their learning progress.

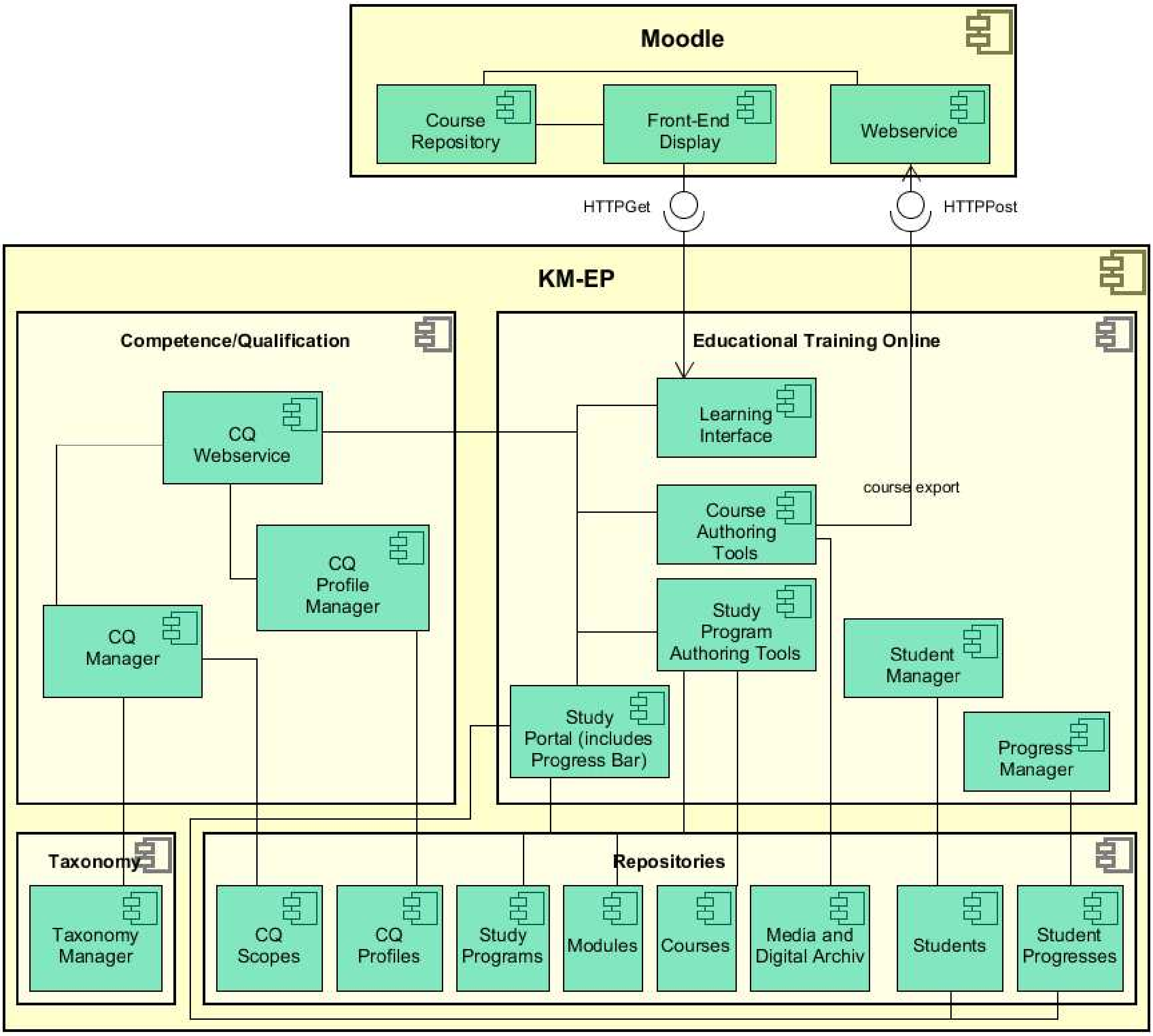

Figure 3: Application scenario AS2: CQ-related KM-EP components (Fig. 62 in [9])

The implementation of the ***LTI-based Moodle-WebAssign-Integration*** refers to application scenario AS3, Figure 4 (Fig. 68 in [11]) gives an overview of the involved components and their interactions. On Moodle-side, the activity type *external tool*, which is included by standard, enables course creators to seamlessly integrate access-protected exercises from WebAssign. Seamless means that before the requested resource is loaded, the logged in Moodle user is authenticated in WebAssign (LTI-service *single sign on*) and in the case of success, the resource is launched (LTI-service *tool launch*) within the Moodle course. If it is directly embedded or displayed in a separate window/tab, depends on Moodle-sided settings. Submitted attempts and received gradings (scores, comments, etc.) are stored in

WebAssign and via the LTI-service *return of outcomes*, the achieved scores can be transferred to Moodle where they are stored in the students' gradebooks.

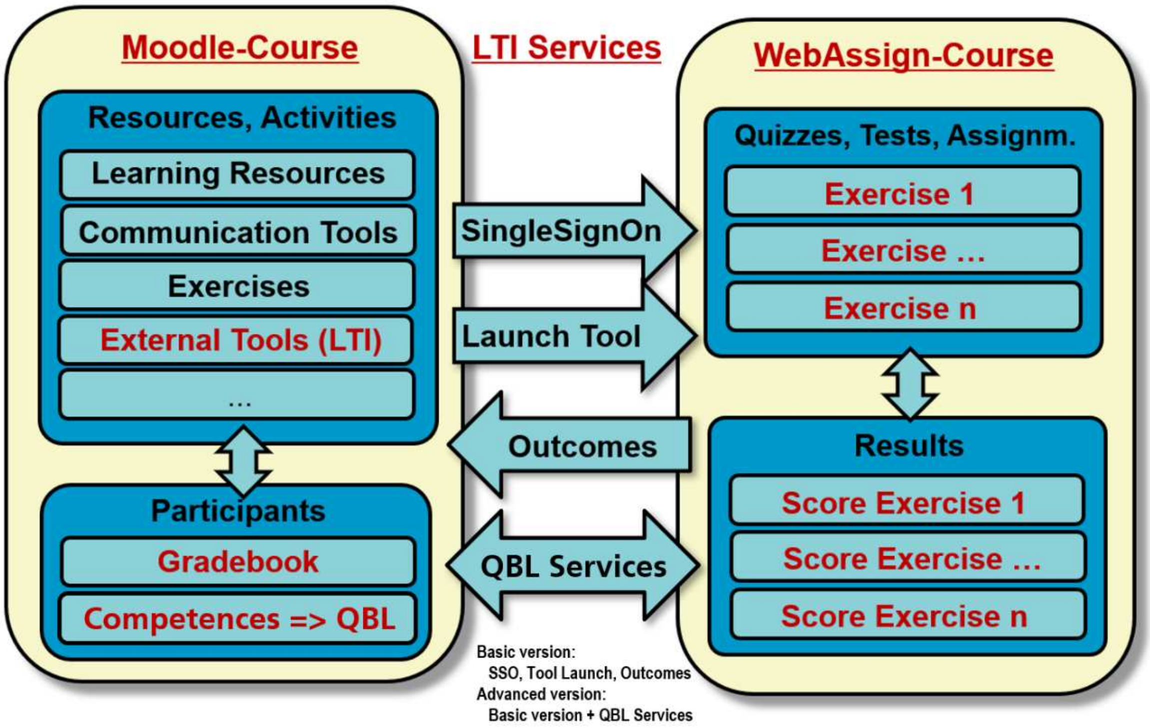

Figure 4: LTI-based Moodle-WebAssign-Integration (Fig. 68 in [11])

In this *basic version* of the LTI-based Moodle-WebAssign-Integration, achieved CQs can be attested to the students by configuring the involved external tools (more precisely: instances of activity type external tool) in a way that combines the configuration settings grade, activity completion, and competencies. As long as the scheduled CQ-based teaching/learning scenarios can be realized with the tools offered by standard Moodle, the basic version does not require any Moodle-sided extensions. QBL application scenarios can only be processed, if QBL4Moodle is installed on the involved Moodle instance. At the time AS3 was conceived, WebAssign did not yet support LTI, LTI-specific extensions were developed in the context of Then's PhD project, see [11] and [14]. The *advanced version* goes beyond the possibilities of the LTI outcomes service and allows a more extensive data exchange. To achieve this, QBL-specific extensions are required in both systems. Since winter semester 2017/18, the basic version of the LTI-based Moodle-WebAssign-Integration is an integral part of FUH's IT-landscape, it is applied for online exercises as well as for examinations. A user manual with the title "Einbinden von Übungssystem-Aufgaben als LTI-Tool in Moodle" can be found in [13], section "Handbücher für Kursbetreuer". A technical documentation describing the implementation is available in [15]. Furthermore, a QBL-related description is given in [11], chap. 4.2.

5. Evaluation, Outreach, and Impact

The QBLM, the QBL application scenarios, and the described software prototypes had been positively rated in several evaluation scenarios with participation of researchers, software developers, and course authors; see [11], chapter 5. As a consequence, subsequent PhD, master and bachelor theses were initiated that are concerned with improvements, extensions, and new application scenarios.

The LTI-based Moodle-EduGame-Integration (**AS4**) is continuously further developed, results are, for example, presented in [16,17]. Since then, a prototypical framework providing tools for different user groups and tasks was created. Developers are supported during game design and implementation; teachers can monitor and analyze students' gameplay performances. The latter is achieved by a gaming platform-sided component that tracks game events and transfers the tracked data back to the embedding LMS. Visualization tools offer different types of diagrams and statistics about the collected performance data. A mechanism for analyzing the tracked data, extracting the achieved CQs, and adding them to successful players' personal CQ profiles is currently in progress. The described tracking concept must already be considered during game development, the events to track and the corresponding CQs have to be specified via an associated API.

The objective of **AS5** is to design a concept for *dynamic learning paths in CQ-based courses* (more precisely: QDM-based courses) and implement it on the basis of Moodle/QBL4Moodle and the KM-EP-CAT. Initial outcomes are presented in [18]. In a first step, a course consisting of activities/resources with separate, QDM-compliant learning goals and access requirements has to be designed. The course can be created either directly in Moodle or in the KM-EP-CAT, from where it is later exported to Moodle. Learning goals and access requirements are adjusted in a way that allows the derivation of reasonable, student-specific processing sequences for each student. Such a personal processing sequence is called dynamic personalized learning path; it depends on the student's current state of attested CQs, which results from prior CQs and the learning progress within the course. Besides CQ-based learning content, this approach requires a personal CQ profile for each student, which has to be automatically updated each time the student achieves a new CQ. Depending on the current CQ profile, the course's activities and resources are either accessible/visible for the student or not. A central evaluation scenario for this approach is a course that conveys basic knowledge and skills in object-oriented programming. Based on a student's current CQ profile, it is decided what is displayed and what is hidden. Some topics, activities (e.g., quiz, assignment, test), and resources can be skipped, others have to be processed/assessed, and still others remain hidden because the required access requirements are not yet fulfilled. So, for each participant, an individually customized course is generated and continuously adapted to his/her learning progress. This way, the learning process for object-oriented programming becomes more effective and, furthermore, all achieved CQs can be considered for follow-up courses that use personal CQ profiles in a similar way.

AS6 emerged from the idea of using *IMS Learning Design (IMS-LD)* compliant *Didactical Structural Templates (DST)* for representing, processing, and exchanging CQ-based QRLEs. Like in AS5, the basis systems for prototypical implementations are the KM-EP-CAT and Moodle/QBL4Moodle. The current state of AS6 is presented in [19]. A DST describes the didactical structure of, for example, a Moodle course and is based on the e-learning standard IMS-LD enriched with QBL-specific concepts. By combining the QBL-approach with IMS-LD, it is possible to define learning goals and access requirements on every IMS-LD element. In [19] it has been shown that it is possible to realize a DST as a "classical" Moodle course, a Moodle course with gaming content, and a standalone applied game. A benefit of DSTs is that learners can choose/switch between different realizations of a specific DST whenever they want. Gamers will usually choose the standalone applied game, others might prefer the traditional approach with a Moodle quiz, still others might want to try both variants. For details about educational games and their interaction mechanisms with gaming platforms and LMSs see AS4. The AS4-approach can also be applied for DSTs, which has been proven in evaluation scenarios covering both web-based and non-web-based games.

AS7 is concerned with the application of QBL for continuous professional education and lifelong learning. Based on the concept described in [20,21], an approach for a *Semantic Qualification Web (SQW)* is designed. This application scenario was motivated by the rapid change of job profiles caused by the ongoing web digitization and globalization. Employees continuously have to gain new and renew existing CQs in order to maintain or improve their employability and to adapt their personal CQ profiles to the requirements of the labor market. Therefore, a goal-oriented (re-)qualification concept for up- and reskilling is needed that helps people to continuously develop and maintain the demanded CQs for their targeted jobs and to react to future job market needs. We propose the creation of an SQW that includes a knowledge base consisting of defined CQs, job profiles, and CQ-based QRLEs (programs, courses, learning content). On this basis, suitable learning paths optimized for the needs of the individual employee can be automatically generated and suggested, the user can make his/her own choice. The term SQW stands for the idea of a semantic network in the qualification sector on the basis of the QDM.

The initial concept is described in [20,21], further developments and proof-of-concept implementations are in progress and will be published soon.

6. Summary and Outlook

In this paper, the QBL approach was introduced and an overview of the current state of development was given. This includes the achievements from 2014-2019, which are summarized in the PhD theses of Wallenborn [9] and Then [11] (short version: [14]), as well as the follow-up developments made by[2] Fischman, Lothary, Srbecky, and Winterhagen since 2018/19. The QBL approach, the QBLM, and the described software prototypes are continuously improved and extended; further QBL-related projects and theses are scheduled.

7. References

[1] European Union, Recommendation of the European Parliament and of the Council of 23 April 2008 on the establishment of the European Qualifications Framework for lifelong learning, 2008, in: Official Journal of the European Union, pp. 1-7, 06 May 2008.
[2] European Committee for Standardization CEN, European e-Competence Framework (e-CF) version 3.0 - a common European Framework for ICT Professionals in all industry sectors, CWA 16234:2014 Part 1, 2014.
[3] R. Koper, Learning Network Services for Professional Development, Springer Berlin Heidelberg, 2009, ISBN 978-3-642-00977-8. DOI 10.1007/978-3-642-00978-5.
[4] R. Koper, C. Tattersall, Learning Design - A Handbook on Modelling and Delivering Networked Education and Training, Springer Berlin Heidelberg, ISBN 978-3-540-27360-8, 2006.
[5] R. Koper, The TENCompetence Domain Model - Version 1.1., 2008. URL: https://core.ac.uk/download/pdf/55533686.pdf, 18 June 2021.
[6] IMS Global Learning Consortium, Learning Tools Interoperability, 2021. URL: https://www.imsglobal.org/activity/learning-tools-interoperability, 18 June 2021.
[7] Moodle.org, Moodle open-source learning platform, 2021. URL: https://moodle.org, 18 Jun 2021.
[8] FernUniversität in Hagen, Germany's State Distance-Learning University, 2021. URL: https://www.fernuni-hagen.de, 18 June 2021.
[9] B. Wallenborn, Entwicklung einer innovativen Autorenumgebung für die universitäre Fernlehre, 2018, Dissertation, FernUniversität in Hagen. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001428, 18 Jun 2021. DOI: 10.18445/20180911-091907-0.
[10] Lehrgebiet MMIA, LGMMIA - Lehrgebiet Multimedia und Internetanwendungen, 2021. URL: http://www.lgmmia.fernuni-hagen.de, 18 June 2021.
[11] M. Then, Supporting Qualifications-Based Learning (QBL) in a Higher Education Institution's IT-Infrastructure, 2020, Dissertation, FernUniversität in Hagen. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001608, 18 June 2021. DOI: 10.18445/20200309-141118-0.
[12] M. Then, M. D. Hoang, M. Hemmje, A Moodle-based software solution for Qualifications-Based Learning (QBL), 2019, FernUniversität in Hagen. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001501, 18 June 2021. DOI: 10.18445/20190225-103757-0.
[13] I. Schulz-Gerlach, Hilfe/Handbücher - Online-Übungssystem, 2021. URL: https://online-uebungssystem.fernuni-hagen.de/hilfe/hilfe.html, 18 June 2021.
[14] M. Then, Ein Ansatz zur Softwaretechnischen Unterstützung des Qualifikationsbasierten Lernens (QBL) an Hochschulen, 2021, in: Hölldobler, S. (Ed.), Ausgezeichnete Informatikdissertationen 2020, Gesellschaft für Informatik. URL (preprint): https://www.researchgate.net/publication/349694936_Ein_Ansatz_zur_Softwaretechnischen_Unterstutzung_des_Qualifikationsbasierten_Lernens_QBL_an_Hochschulen, 18 June 2021.
[15] I. Schulz-Gerlach, Technische Doku zur LTI-Integration - Das Online-Übungssystem als LTI-Tool-Provider, 2021. URL:

[2] in alphabetical order

https://online-uebungssystem.fernuni-hagen.de/download/LTI_Moodle/LTI_TechnicalDoc.html, 18 June 2021.
[16] M. Winterhagen, M. Salman, M. Then, B. Wallenborn, T. Neuber, D. Heutelbeck, M. Fuchs, M. Hemmje, LTI-Connections Between Learning Management Systems and Gaming Platforms: Integrating a Serious-Game Prototype into Moodle Courses, 2020, in: Journal of Information Technology Research (JITR) 13(4), pp. 47-62, 2020, URL: https://www.researchgate.net/publication/346228285_LTI-Connections_Between_Learning_Management_Systems_and_Gaming_Platforms_Integrating_a_Serious-Game_Prototype_Into_Moodle_Courses. DOI: 10.4018/JITR.2020100104, 18 June 2021.
[17] R. Srbecky, M. Then, B. Wallenborn, M. Hemmje, Towards learning analytics in higher educational platforms in consideration of Qualifications-Based Learning, 2021, Edulearn 2021 - 13th International Conference on Education and New Learning Technologies, July 2021. DOI: 10.21125/edulearn.2021.1046.
[18] F. Fischman, H. Lersch, M. Winterhagen, B. Wallenborn, M. Fuchs, M. Then, M. Hemmje, Individualized Educational System Supporting Object Oriented Programming, 2020, in: Advances in Software Engineering, Education, and e-Learning - Proceedings from FECS'20, FCS'20, SERP'20, and EEE'20, Springer International Publishing, 2021. ISBN: 978-3-030-70872-6. DOI: 10.1007/978-3-030-70873-3.
[19] M. Winterhagen, M. D. Hoang, B. Wallenborn, D. Heutelbeck, M. Hemmje, Supporting Qualification Based Didactical Structural Templates for Multiple Learning Platforms, 2020, EEE'20 - The 19th International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, Las Vegas, USA, July 2020. URL: https://www.researchgate.net/publication/343255892_Supporting_Qualification_Based_Didactical_Structural_Templates_for_Multiple_Learning_Platforms, 18 June 2021.
[20] B. Humm, H. Bense, M. Fuchs, B. Gernhardt, M. Hemmje, T. Hoppe, L. Kaupp, S. Lothary, K.-U. Schäfer, B. Thull, T. Vogel, R. Wenning, Machine intelligence today: applications, methodology, and technology, 2021, in: Informatik Spektrum, 18. June 2021. DOI: 10.1007/s00287-021-01343-1.
[21] S. Lothary, W. Krasowski, B. Wallenborn, M. Then, M. Fuchs, M. Hemmje, Supporting the Representation of Personal Learning Paths for Goal-Oriented Professional Qualification, 2021, Edulearn 2021 - 13th International Conference on Education and New Learning Technologies, July 2021. DOI: 10.21125/edulearn.2021.0813.