

Learning Analytics in Augmented Reality

Blueprint for an AR / xAPI Framework

Joshua Secretan, Dr Fridolin Wild, Will Guest

Performance Augmentation Lab

Oxford Brookes University

Oxford Brookes

cct-pal@brookes.ac.uk

Abstract — Learning Analytics in flat web platforms is rather commonplace, but recording learner performance from immersive real-world experience still poses a challenge. With this study, we break ground and show that Learning Analytics for Augmented Reality are possible and we outline how an according system can be implemented. We describe a novel framework for competency-based workplace Learning Analytics and document the required components with regards to software, hardware, and their configuration. Thereby, we discuss key design decisions and lessons learned from the implementation of the framework, drawing from the experience of the EU project Learning Analytics in Augmented Reality (LAAR), which provided the environment and context for this work. Most notably, this included testing the technologies in realistic scenarios in vocational training of stage technicians. Finally, we conclude the paper with a summary of the work done and an outlook on possible extensions for researchers wishing to build on our work.

Keywords—Augmented reality, Learning Analytics, Experience API, Learning Locker, ESCO

I. INTRODUCTION

Technology is constantly changing how we work and live, unfortunately with education and training often trailing behind technological innovation incepted in other areas. At the moment, the umbrella term of ‘Industry 4.0’ is often used to characterize the current wave of automation promising to reinvent the way we produce. In the context of this, labour market studies predict significant disruption, for example, with [1] estimating 47% of total US employment is at risk of redundancy. Further looking studies like [2] more provokingly conclude that the “full automation of labor (all human jobs)” may happen on a horizon of just over a hundred years. Even if looking at methodologically more conservative studies, such as [3], who find that “on average across the 21 OECD countries, 9% of jobs are automatable”, the truth is probably somewhere in the middle and certainly dependent on political will and organizational change management. Gone are the days when professions transcended family generations. Those not lucky enough to specialize, the core of our society, the working class, may need to re-train multiple times over their career life.

The answer to this challenging context and the technological opportunity emerging from it, in our view is performance augmentation. In practice, we suggest the

education and training sector needs to tackle two core problems to make this opportunity a success. First, *better AR and wearable systems for learning, teaching, training* are possible and need to be implemented. Augmented Reality using, e.g., projection or smart glasses, allows embedding information directly in its application context in the real world and using that for learning and training. The merit of such techniques is outlined by [4] who, through analysis of 78 studies “investigate how Augmented Reality and sensor technology can be used to capture expert performance in such a way that the captured performance can be used to train apprentices”. Dedicated hardware solutions now exist, such as the Microsoft HoloLens, the MagicLeap, etc. Mobile phones are finally catching up with the capabilities of these fringe devices. SDKs and toolkits supporting more rapid development facilitate an explosion of AR learning and training applications. Second, we need better *Learning Analytics for real-time and real-world interaction*. The adoption of technology-enhanced learning in the last decades brought with it the ability to collect learners’ activity traces to help improve competence development, an established field called Learning Analytics. So far, however, Learning Analytics only scratch the surface of what they are capable of, not yet connecting up with the new immersive technological possibilities of VR/AR/MR. Contributing to resolve this shortcoming, ground-breaking work by [5] discusses the idea of Immersive Analytics. Moreover, [6] demonstrate how data can be collected from wearable technology. Unanswered questions remaining, however, are what traces and what data should be actually recorded, and what logic operates to perform when interrogating the data collected.

In this contribution, we propose a novel framework to help remediate this situation, implementing an analytics framework using the Experience API (xAPI) on top of an existing AR training solution. To serve other developers, the paper highlights design decisions and documents lessons learned.

This paper begins by describing the suggested framework, its terminology and normative principles. The next section details the xAPI profile created to support this project and collecting AR experiences. The third section discusses how to log an AR experience and how to connect this to a skills database. There is then a brief look at how collected data can be analyzed and displayed. The paper then ends with a look at the lessons learnt, the conclusions and future work.

II. FRAMEWORK

Before turning to the system components of the framework, we first like to clarify the relation of learning activities (and learner action) to competence and performance, as these are the integral concepts required for the analysis of learning. Moreover, as we will show, this will also help with the definition of the data models needed to perform AR Learning Analytics.

An ‘*action*’ is the smallest possible step that can be taken by a user in a ‘*learning activity*’. ‘Locate the hammer’. ‘Read the text’. ‘Watch the video’. Users leave digital traces in the system when performing the actions of a learning activity, facilitating the logging of what a trainee or learner is doing to a central or decentralized analytics storage endpoint. Clearly, performing an action does not mean it will always be motivated by and executed with competence, but hardly any trainable competence develops without engaging in actions.

In our understanding, ‘*competence*’ is merely a potential for action and as such is not directly measurable [7]. Competence becomes visible when learners perform. Moreover, competence assertions require some or other form of validation. For example, ETTE, the European Theatre Technicians Education standard, defines competence as “the *proven* ability to use skills and underpinning knowledge and attitudes” [8]. When assessing competence, we interpolate from a person’s action performance.

In the proposed framework, a *competency taxonomy* defines how we group and name the specific skills, knowledge, abilities, and other characteristics of a particular occupation. Following the proposal in [9], we use ‘*competence*’ as the generic state, and ‘*competency*’ to denote a particular skill. Typically, there is a non-trivial relation between actions to be taken as part of a learning activity and the specific competencies they motivate developing. Think of changing the oil in a car: A user may have the skills to do this, or even the knowledge of why oil changes matter, but for a mechanic to be judged competent it takes a much deeper understanding of an engine to diagnose when things go wrong or how oil issues contribute to wicked problems.

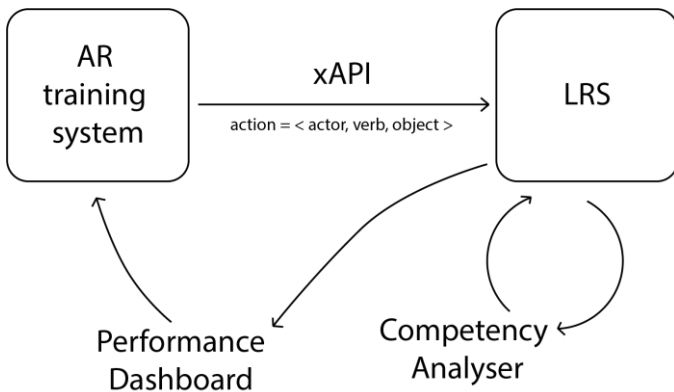


Fig. 1. Component Architecture

Now, last but not least, *performance* refers to the analytics. Performance is competence in action, the sum of demonstrated competencies. The crux for Learning Analytics is often that actions are visible, but it is not clear to which competency context they belong. Only when we have information about whether an action is a demonstration of a particular competency, then it becomes possible to extract meaningful performance analytics. An easy way to consider how the three definitions combine is: Performance = Action + Competence.

Thereby, the *AR training system* shall represent both the hardware layer and the software application. In our implementation, this is a Microsoft HoloLens and a modified WEKIT application [19].

The *Experience API (xAPI) interface* is a software layer which must implement the xAPI specification [10] to allow statements to be formed and sent. The Experience API is both successor project and standard complement to the widely used Shareable Content Object Reference Model (SCORM) standard [11]. The xAPI provides a normative set of services for storing and retrieving statements about learner activity. These packets are logged in the form <Actor, Verb, Object>. An example is “John found the exit”. Further fields can be added to these statements, like ‘result’, ‘context’, ‘authority’ and ‘attachments’.

The *Learning Record Store (LRS)* then acts as the streaming database for xAPI statements. In our case, we deployed the Open Source project LearningLocker [12].

The *Competency Analyzer (CA)* is the logic component server-side. This should be unique for each new AR training system, tailored to be able to understand and process the statements. Finally, *Performance Dashboard (PD)* allows the user to view their progress. It visualizes the outputs of the competence analyzer. This typically is either a component of the AR training app or of the dashboard provided by the LRS. In our case, the learning locker provides such a dashboard.

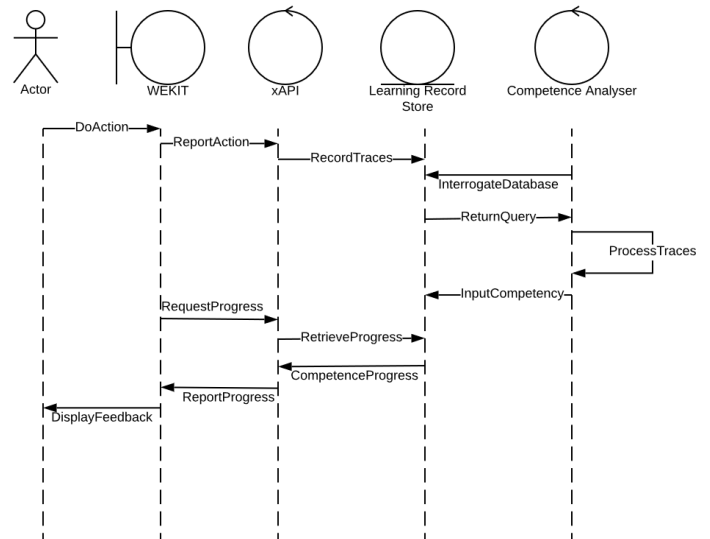


Fig. 2. Sequence diagram of component interaction.

Figure 2 shows the order of interaction and process names for a session within the framework. In this scenario a user performs a single action (although many are actually required). This action is reported to the xAPI layer running within the modified WEKIT app [19]. xAPI statements containing the traces of information gatherable are then created and stored in the LRS. At a high frequency the competence analyzer should then interrogate the LRS, processing these traces and comparing them to performance, allowing a measure of competence to be calculated. This is then asserted back into the LRS in form of further xAPI statements. TinCan [13], the Rustici Software implementation of the xAPI specification, was used to send statements to the LRS. Modifications were made to the APIs networking code which is documented in [14].

III. XAPI PROFILE

To support the recording of a user’s actions in the real world, a vocabulary is needed to describe them. In the xAPI, a ‘vocabulary profile’ describes a collection of verbs combined to define a single type of use case. The serious games profile [13] and the video profile [14] served as best practice orientation examples.

All vocabularies are ideally application profiles that re-use existing standard vocabularies as much as possible, featuring only those verbs that are needed and aligning them to a well-formed collection, while adding missing verbs that are application-specific.

Verbs fall into groups and three different types of verbs can be distinguished, see Table 1 [15].

Area	Description
<i>Tool & Tech specific verbs</i>	Information about user engagement on app level (app started, quit, ...)
<i>Domain-Specific Verbs & Objects</i>	Information about real-world and digital content engagement
<i>Custom Progressions & Patterns</i>	Information about learner achievement (learning experiences completed, failed, ...)

Table 1. Vocabulary types (extracted from [15])

A simple xAPI profile was developed to facilitate the capturing of data in these settings, bootstrapped from existing vocabularies. The following verbs have been used:

Group	Verb	Source	Description
<i>Tools & Tech</i>	Initialized	adlnet.gov	Learning experience loaded and configured (using, e.g. calibration marker for spatial anchor recalibration)
	Launched	adlnet.gov	Learning experience

			started
	Abandoned	adlnet.gov	User quit / app crashed
	Exited	adlnet.gov	User left intentionally
<i>Domain-specific (AR media)</i>	Viewed	tincanapi.com	Record user viewed the content (video, image, label, animation)
	Listened	activitystrea.ms	A record that audio was listened to
<i>Domain-specific (real-world handling & movement)</i>	Table 2+3	WEKIT ESCO/ETTE	Action predicates as described in ARLEM spec
<i>Progression</i>	Evaluated	tincanapi.com	The teacher evaluated learner statement
	Rejected	activitystrea.ms	The teacher rejected the learner claim
	Accepted	activitystrea.ms	The teacher accepted the learner claim
<i>Progression</i>	Attempted	adlnet.gov	Learner did not finish the experience, activity aborted/abandoned
	Failed	adlnet.gov	Competency not achieved
	Passed	adlnet.gov	Competency achieved
	Skipped	tincanapi.com	User jumped over the experience
	Completed	activitystrea.ms	Learner finished the experience

Table 2. xAPI verbs

Moreover, for the domain-specific parts, a taxonomy for the hands-on instruction is needed, for example, the WEKIT taxonomy, or the ESCO extract of verbs from concrete, instructable skills. Verbs like ‘adapt’ from ‘adapt to artists’ creative demands’ are dropped, as they are too abstract.

A single verb is repeated for multiple skills. A stage technician should be able to ‘assemble the rehearsal set’ and ‘assemble the rehearsal set’. The same verb ‘assemble’ transfers to many other occupations. A piano maker must be able to ‘assemble musical instrument parts’.

WEKIT verbs of handling and movement for Industry 4.0 maintenance		
Allow	Lift	Place
Point	Locate	Plug
Assemble	Lower	Rotate CCW
Close	Lubricate	Rotate CW

Cut	Measure	Screw
Disassemble	Open	Unfasten
Drill	Pack	Unpack
Forbid	Paint	Unplug
Inspect	Pick	Unscrew

Table 3. Modified from [16]

ESCO verbs for stage technician essential skills		
adjust	assemble	assess
de-rig	dismantle	distribute
draw	ensure	fit up
focus	follow	handle
hang	install	keep up
light	manage	mark
modify	operate	pack
plot	prepare	prevent
provide	run	set up
technically design	understand	use
work		

Table 4. ESCO verbs extracted from stage technician essential skills [17].

IV. LOGGING AR LEARNING EXPERIENCES WITH XAPI

There should be two forms: Low-level logic for interpreting the individual action behaviour of a user. This should link single actions to outcomes. Higher-level logic than can be used to either reveal (where information extraction is possible) or add (were not) the competency analysis on a larger scale.

Linking of behaviour statements to connected competencies is, however, a complex problem. This is both a conceptual as well as a technical problem. Conceptually, the difficulty lies in the validation of assertions of competence, i.e., figuring out when the learning activity is actually successful and competence of the learner can be assumed. This typically requires assessment in one form or the other. Ideally, there is an automated system that can detect from the

behaviour statements the successful demonstration of a competency. Technically, the challenge is to model the linking to the competency context for those behaviour statements that are part of the training, while linking the competency as an outcome where assertive statements are added.

ESCO, the European database for Skills/Competences, qualifications and Occupations, contains competency taxonomies for each occupation (currently 2,942 occupations). The occupation ‘stage technician’ contains 43 essential competencies and 31 optional ones. For example, there is a competency named ‘assemble truss connections’, which has the defined uniform resource identifier (URI):

<http://data.europa.eu/esco/skill/e73c5d59-5e54-4238-bd2a-9aded8083107>

Our tested truss assembly learning activity consists of eleven action steps, several of the steps with multiple augmentations. Starting/exiting the activity, action steps, as well as all augmentations are logged with their own triple statements. To connect now these <actor, verb, object> statement to the competency ‘assemble truss connections’, we add the URI from above to the context section. This allows logging the competency which the learner activity is serving.

```

"context": {
  "contextActivities": {
    "parent": [
      {
        "objectType": "Activity",
        "id": "http://data.europa.eu/esco/skill/e73c5d59-5e54-4238-bd2a-9aded8083107"
      }
    ],
    "grouping": [
      {
        "objectType": "Activity",
        "id": "http://data.europa.eu/esco/occupation/acb6b99a-6c13-482a-81b1-8b5614e0153e"
      }
    ]
  }
},

```

Clearly, a single statement does not make up the full competency demonstration. It is the sequence rather and their successful completion that makes up the competency demonstration.

There are the following options for lifting the activity and behaviour statements up to the level that they assert competency demonstration statements: self-assessment, peer-assessment, teacher-assessment, machine-assessment, or no assessment. All of these act slightly differently in regards to how the competency demonstrations are validated, but they all share that the logging of the action behaviour is separate from the assertion of competency.

So, in the example, when the activity is successfully finished, it becomes possible to insert a progression statement that contains the competency URI as a result, logging that the learner ‘passed’ or ‘attempted’ or ‘failed’ to demonstrate that competency.

```

"result": {
  "extensions": {
    "http://data.europa.eu/esco/skill/e73c5d59-5e54-4238-bd2a-9aded8083107": "passed"
  }
},

```

Depending on the type of assessment, validation can either be self-asserted (guided by the app, self-certification), teacher assessed (separate app used for observation/inspection of activity and validation of successful outcomes), or automatically evaluated based on defined performance criteria (e.g. using system functionality or queries over logs to check for repetition, precision, etc.).

We recommend to automatically insert 'completed' at the end of the activity, complemented by 'passed' or 'failed' if using a teacher assessment app or automated assessment with an intelligent system. A simple example of machine validation is a learner action of the type 'locate the X', where the user has to identify a specific part or location, and, using ARLEM's stare gaze activation with a locate trigger, the assertion of the validated completion of actually having found the X becomes possible.

V. DASHBOARD & COMPETENCE ANALYSER

The competency analyzer then can look like the one depicted in Fig.2, allowing to configure queries for extracting data about specific learner behaviour.

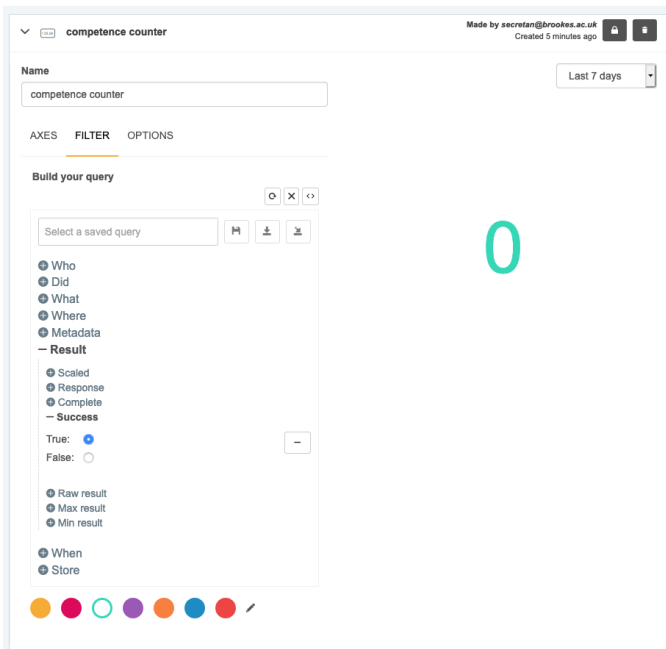


Fig.2. Query interface for creating stored competency queries.

The statements in the LRS can be visualized back to the learner by using a performance dashboard, such as the one depicted in Fig.3.

VI. LESSONS LEARNED

The implementation verifies that the framework works with the flexibility required as with regard to the vocabulary choice. It provided us with feedback on the key obstacles in adoption. The primary concerns we identified thereby are: how to assess whether the learner has actually learned something, what evidence can be brought forward to back up such claim, and which competency taxonomy should be used.

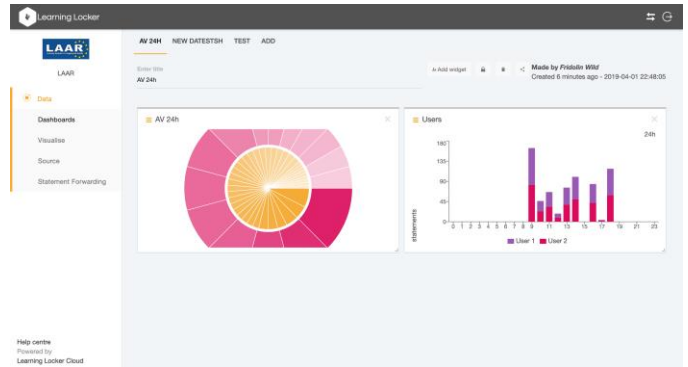


Fig.3. Performance dashboard.

The first issue about validating behaviour relates to the way AR training exercises are built. It is paramount to create learning activities in a way that they allow logging of behaviour implicitly and explicitly. Explicitly means deploying triggers (voice, gaze, air-taps, sensor values), as the trigger can be utilized to drop the progress statements to the endpoint.

It is possible, however, to use implicit behaviour validation as well, inferring that previous action steps must have been performed, or even performed correctly if subsequent ones allow inferring such conclusion. For example, if a second action step uses an explicit trigger on a marker target on an object inside of a box, and the first action step is the instruction to open the box, then the visibility of the fiducial marker allows inferring that the box must have been opened.

There is, however, no generic way of determining how statements about competency can be alleviated from simple observational behavioural traces. This is always up to the specific learning exercise, the assessment regime (high stakes differ significantly with respect to proctoring requirements from low stakes exams, for example). The framework is open to different assessment methods and allows to process raw data about simple behavioural traces with queries or even machine learning.

As for the nature of evidence, several insights could be gained from the implementation. By its very nature, any AR capable device must have a camera feed. This enables capture of user results, which can later be reviewed. The xAPI *attachments.fileURL* field combined with an external server provides storage. Through this, evidence can be collected and processed. While of course, machine learning and AI solutions can be utilized to investigate the evidence, it is also possible to complement the learner app with a trainer app, where assessors or peers validate compliance with expectation. In that case, two devices were necessary, both apps running simultaneously or asynchronously.

As with regard to the competency taxonomy to be used, we have provided in the framework a way to interface with any that is able to provide unique resource identifiers on competency level. In the end, the choice of the taxonomy is left to the instructor. Taxonomies are typically made for specific purposes. For example, ESCO focuses on

comparability across the EU labour market and education and training, setting emphasis on standardization, even across professions. It provides an API. For stage technicians, ETE (European Theatre Technicians Education) provides a fairly more detailed analogue description of skills, used in many institutes to test students against. Both influenced each other.

VII. CONCLUSION AND FUTURE WORK

Reflection is a fundamental working principle in learning, and, therefore, any human performance augmentation must embrace reflective practice in the sense of the concept proposed in [18], which distinguishes the types in-action and on-action. Learning Analytics are the contemporarily most prominent method to facilitate reflection-in-action (during, in situ, as an act of problem-solving) and reflection-on-action (after, possibly in a different space, as an act of creative sense-making of the past).

Within this contribution, we have proposed a framework for Augmented Reality Learning Analytics needed to foster such types of reflection. We have outlined the key concepts of action, learning activity, competence, competency, and performance, and described all involved technical components – from AR training system, xAPI, LRS, Competence Analyzer, and Analytics Dashboard. We have reviewed existing vocabulary profiles and proposed a suitable selection of verbs for tracking learning performance in AR, which fall into the three groups of tools & tech-specific, progression, and domain-specific. For the latter, we reported two domain-specific vocabularies, one for more general maintenance handling and movement, the other for stage technicians. We have documented how xAPI statements need to be linked with the competency taxonomy.

With all this, we show that the technology is ready for putting Learning Analytics into Augmented Reality. The implementation serves as verification that such LA-AR system can actually be implemented, but also as a basis for the user test we briefly report in Section 5.

We see the need for future work, refining, in particular, the domain-specific vocabulary, best as a community of practice. Moreover, more work is needed for automated validation of competency assertions based on simple behavioural traces using a database, AI, and machine learning technology. This would help to improve the way we can track whether the learner did something with the required precision, practice, or other. This may or may not feedback to the profiles emerging.

ACKNOWLEDGEMENT

The work behind this paper was co-funded by the Erasmus+ program of the European Union in the LAAR project (grant no. 2017-1-LI01-KA202-000087) and the ARETE project (grant no. 856533). We would especially like to thank our collaborators in LAAR, namely: Peter Sommerauer (I-SMART), Chris Van Goethem (STEPP), Oliver Müller and Leonard Maxim (IT University of Copenhagen), and, last but not least, Tommy Neumann of VPLT.

REFERENCES

- [1] Carl Benedikt Frey and Michael A. Osborne (2013): The Future of Employment: How Susceptible are Jobs to Computerisation?
- [2] Kaja Grace, John Salvatier, Allan Daoe, Baobao Zhang, and Owain Evans (2017): When Will AI Exceed Human Performance? Evidence from AI Experts, *JAIR* 62(2018): 729-754
- [3] Arntz, Melanie; Gregory, Terry; Zierahn, Ulrich (2016): The Risk of Automation for Jobs in OECD Countries, *OECD SEM Working Papers*, No. 189
- [4] Bibeg Hang Limbu, Halszka Jarodzka, Roland Klemke, Marcus Specht (2019): Using sensors and augmented reality to train apprentices using recoded expert performance: A systematic literature review, *Educational Research Review*, Vol.25(2018):1-22.
- [5] Tom Chandler, Maxime Coreil, Tobias Czauderna, Tim Dwyer, Jaroslaw Glowacki, Cagatay Goncu, Mattias Klapperstueck, Karsten Klein, Kim Marriott, Falk Schreiber and Elliot Wilson (2015): Immersive Analytics, In: *Big Data Visual Analytics (BIDA'15)*, IEEE.
- [6] Ralf Klamma, Rizwan Ali and István Koren (2018): Immersive Community Analytics for Wearable Enhanced Learning, In: *Learning and Collaboration Technologies*, Springer.
- [7] Wild, F. (2016): *Learning Analytics in R with SNA, LSA, and MPIA*, Springer.
- [8] Christian A. Buschhoff, Koen Deveux, Willem Groenewoud, Anders Larsson, Ulf Nielsen, Rainer Münz, Harald Prieß, Bastiaan Schoof, Hugo van Uum, Rob Acket, Marc Vandermeulen, Chris Van Goethem (2018): *European Theatre Technicians Education*, DTHG.
- [9] Tina Tedorescu (2006): Competence Versus Competency - What is the Difference?, In: *Performance Improvement* 45(10).
- [10] The Advanced Distributed Learning Initiative (2019): *Shareable Content Object Reference Model (SCORM)*, online at: <https://scorm.com/>, last access: 2019-07-22
- [11] The Advanced Distributed Learning Initiative (2017): *xAPI Specification*, <https://github.com/adlnet/xAPI-Spec/>, last a.: 2019-07-22
- [12] ht2labs (2019): *Learning Locker*, <http://learninglocker.net/>, last access: 2019-07-22
- [13] Rustici (2019): *TinCan for C#*, <https://github.com/RusticiSoftware/TinCan.NET>, last acc.: 2019-07-26
- [14] J. Secretan (2018), *TinCan for Hololens*, <https://github.com/SecretanHazCode/TinCan.NET>, last acc.: 2019-07-26
- [15] e-ucm (2017): *Serious games xAPI profile*, <https://github.com/e-ucm/xapi-seriousgames/blob/master/xAPI%20Profile.md>, last a.: 2019-07-26
- [16] xAPI Video Community of Practice (2019): *Video*, <https://liveaspankaj.gitbooks.io/xapi-video-profile/content/>, last access: 2019-07-26
- [17] Tscheulin, A. (2018): *Using xAPI Profiles to Turn Unique Business Logic into Actionable Data*, online at <https://www.yetanalytics.com/blog/using-xapi-profiles-to-turn-unique-business-logic-into-actionable-data>, last access: 2019-07-22
- [18] Schön, D.A. (1983): *The reflective practitioner*. Basic Books New York.
- [19] Kaj Helin, Jaakko Karjalainen, Alla Vovk (2018): *D4.5 AR-visualization System and AR-based Experience Re-enactment and Learning System d4.5* (Fig. 20, p. 22)
- [20] European Commission (2019): *European Skills Competences Qualifications and Occupations (ESCO)* Available: <https://ec.europa.eu/esco/portal/home>. last access: 2019-07-22.