A scalable architecture for one-stop evaluation of serious games

Iván J. Pérez-Colado[1[0000-0002-1250-106X]], Víctor M. Pérez-Colado[1[0000-0001-9778-8129]], Iván Martínez-Ortiz[1[0000-0001-6595-5690]], Manuel Freire[1[0000-0003-4596-3823]] and Baltasar Fernández-Manjón[1[0000-0002-8200-6216]]

[1] Department of Software Engineering and Artificial Intelligence, Complutense University of Madrid, C/ Profesor José García Santesmases, 9. 28040 Madrid, Spain

Abstract. Evaluating a serious game is a time-consuming task. However, good evaluations are necessary to improve the effectiveness of serious games, and to prove this effectiveness to stakeholders. Computer support of evaluations requires addressing several problems, including security, privacy protection, data collection from both questionnaires and in-game activities, data analysis, and management of the experimental workflow. We describe improvements to the Simva architecture to add scalability and a bridge to exploratory data science to our one-stop serious games evaluation platform. Simva supports evaluations ranging from small-scale pilots to full-fledged validations with complex experimental designs. The improvements described in this paper greatly increase ease of deployment, interoperation with existing authentication infrastructure, and scalability of Simva, and can be readily applied to tools with similar goals.

Keywords: Serious games, evaluation, learning analytics.

1 Introduction

Serious games are currently expensive to evaluate. Additionally, traditional questionnaire-based evaluations are limited in scope, and require substantial effort to yield additional insights. Our evaluation platform, Simva, streamlines evaluations by bringing all required steps under a single roof. We present a new architecture for Simva that makes it easier to install and use; and which allows extracting more insights from collected data by interoperating with common data science tools, such as Jupyter Notebooks.

The next section of the paper describes how serious games are generally evaluated and includes examples of common experimental setups for evaluating serious games. Then, we present improvements to Simva to simplify validation by automating all steps, and simultaneously simplifying the roles of the system administrators which install and integrate Simva and the researchers that use it to run their experiments and look at the resulting data. Finally, we describe our conclusions and planned future work.

2 Evaluating Serious Games

Serious games are often evaluated with pre-post experiments, in which participants fill in questionnaires both before (pre) and after (post) playing the game [1]. Improvements in questionnaire scores are attributed to in-game learning, and when statistically significant, result in games that are formally evaluated as effective.

To perform a pre-post test, experimenters need to prepare, distribute among participants, and score both the pre and post versions of the questionnaire. They also need to ensure that each post-test is paired with its corresponding pre-test. This can be automated through online forms, such as LimeSurvey; but online forms are generally not designed to support this workflow, and even when programmable, require significant effort to do so.

Experimental designs often include the use of control groups. For example, members of a control group could be asked to play a second game, of equal duration but unrelated to the serious game under study. Improvements in pre-post scores from control group participants allow learning effects from the tests themselves to be measured and quantified. In a more complex approach, a counter-balanced experimental approach would include two sequences of experiments for each of two groups: participants in the first group would be requested to fill in a pre-questionnaire, play the intervention game, fill in a post-questionnaire, perform the control activity, and then fill in a final questionnaire. Those in the second group would reverse the order of the control activity and the intervention game. The advantage of this approach is that all participants end up performing all activities, avoiding unfairness. As a final example, a recall experiment measures test-score variations several days or weeks apart from the interventions. Logistically, they are harder to carry out, since additional experimental sessions must be scheduled in addition to the main experiment where the game is played.

In a recent literature review on serious games to address bullying and cyberbullying [2], 45% of the 42 publications describing experiments used paired pre and post-tests, 90% of the experiments used a single session of gameplay (as opposed to several), and 42% of the 26 games used control groups for their experiments. While not a representative sample of all serious games, these numbers hint at the difficulty of carrying out more complex experiments

2.1 Collecting interaction data to improve evaluations

Interactions between the players and serious games can also be collected and analyzed, either in real-time or after the session is through. Several standards for reporting these interactions are currently in use. General interactions between the learning system and learners can be formatted and sent to a server for analysis using standard formats, such as ADL's xAPI statements [3] or IMS' Caliper events [4]. Using game-specific vocabularies to describe these interactions decreases ambiguity by using standard definitions of most game-relevant concepts; this is the purpose of the xAPI Serious Game profile, proposed in [5].

Game Learning Analytics is the combination of Game Analytics and Learning Analytics and is dedicated to gaining insights from such interactions [6]. To use GLA in an evaluation, playthrough data for each participant must be linkable with their pre and post-test responses.

3 The Simva Approach

In [7], we described the main functionality of the initial version of Simva, which integrated LimeSurvey, an open-source survey management system, with RAGE Analytics, a serious game analytics platform developed for the RAGE H2020 project [8]. By managing identifiers and handling pre-post questionnaires, in addition to collecting user interaction data, this version of Simva addressed many of the requirements for experiments such as those described in [2]. Indeed, based on our experience managing large experiments [9], we have identified multiple tasks that usually are problematic and cumbersome in evaluations, and sought to address them with Simva.

Improvements to Simva further streamlined the process, adding support for more complex experiments [10]. In addition, we identified the opportunity to integrate evaluation into game authoring, so that game authors using tools that support Simva (uAdventure) could enjoy built-in validation support [7]. However, those authors would still need to install the Simva platform to be able to enjoy these advantages.

3.1 Integrating Simva

Our goal is to simplify game evaluation. The rest of this work motivates and describes the architectural changes needed to make Simva (and evaluation tools in general) easier to install, scale and integrate, both for small (local installation) and large (cloud/institutional) deployments.

Fig. 1. illustrates the new architecture of Simva, with the main modules as rounded boxes. Modules are currently distributed as docker containers and configured to interoperate together through docker-compose (docs.docker.com/compose). Docker containers generally encapsulate or all dependencies of a service and can be seen as lightweight virtual machines. Use of these containers strikes a balance between ease of testing in single-workstation scenarios and scalability in larger installations: many public clouds, such as those offered by Amazon, Microsoft and Google, support kubernetes containers. However, they are harder to work with locally; and the greatest leap, architecture-wise, is from a monolithic to a distributed system – once a system has been split up into collaborating containers, changing the type of containers is relatively straightforward.

In addition to cloud support, containers such as docker have several advantages for development. First, it is possible to use trusted and tested containers for common functionality with minimal or no modification, thus avoiding costly implementations and gaining access to free maintenance in the form of bugfixes and newer versions. In this sense, Kafka and Kafka-Connect, Minio, KeyCloak, Traefik and LimeSurvey (depicted in Fig. 1) are existing open-source projects that we did not need to reimple-

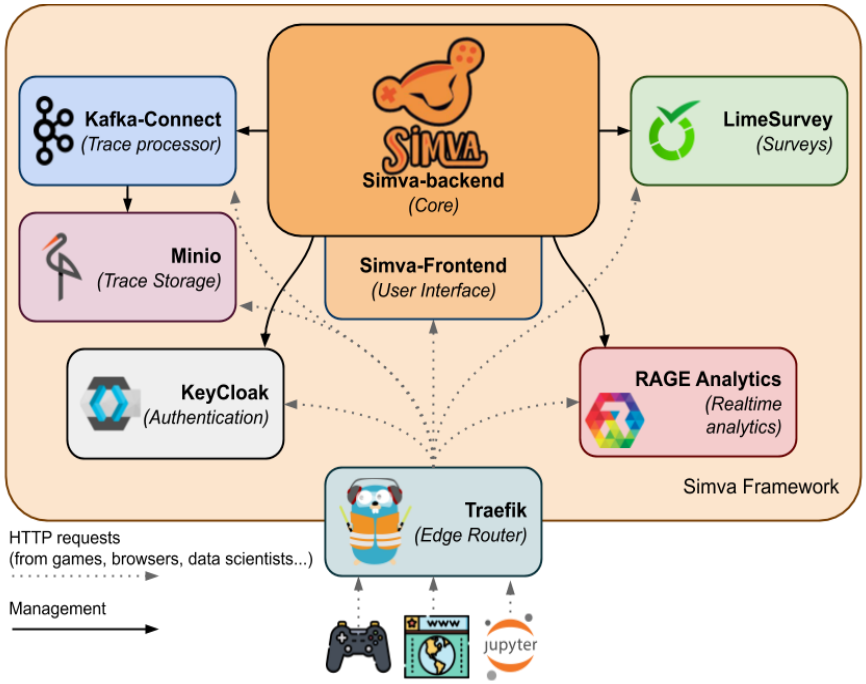

Fig. 1. Overview of the main components of the *Simva* Framework. *Traefik* redirects requests to each service, while *Simva-Backend* is the core that manages all other components.

ment, potentially saving several years of development efforts. Additionally, as these actively maintained projects are updated, we will have the chance to upgrade to newer versions with minimal additional effort on our part. We are the authors of RAGE Analytics, but it is also an open-source project distributed as a collection of docker containers. An additional advantage is that changes to one container are, by the very nature of containers, isolated from each other; so that use of incompatible library versions from within different containers can never be an issue.

However, bringing together multiple systems does have a cost. In particular, users rightfully expect to be spared from the internal complexity of the system. To this end, Simva's modules are hidden behind an application gateway (Traefik). Single-Sign-On (SSO), provided by KeyCloak, allows the whole platform to appear as a single coherent web application to the outside world, with 3 main entry-points, depicted at the bottom of Fig. 1:

- Web access to the front-end, after suitable authentication, allowing studies to be configured, managing groups of participants and their activities, and accessing collected data once through a browser once it is available
- Ingress of xAPI-Serious Game statements describing how players interact with games, to be stored for later analysis; and, in the case of activities where real-time analysis is enabled, to be made available as dashboards that display real-time analytics based on player activity.

- Authenticated access to stored experimental results for access from external data-science tools, such as Jupyter Notebooks.

To achieve interoperation between all modules required some effort, mostly related to enabling and configuring SSO and streamlining communication between them. For example, LimeSurvey was modified to use a SAML plugin to interoperate with KeyCloak for SSO purposes; and we made additional changes to its RemoteControl API to better manage surveys from the Simva backend. Minio, KeyCloak and Kafka-connect were similarly customized to interoperate better.

The move to a containerized architecture has brought the following advantages to Simva:

- Easy deployment. Containers are extensively used in cloud deployments, covering scalability; and docker containers are lightweight enough to be deployed locally for testing or smaller deployments, with minimal dependencies.
- KeyCloak supports major SSO technologies such as OpenID or SAML, allowing integration of Simva into existing institutional authentication systems, so that users need not memorize additional passwords.
- Traefik hides the inner complexity of the system, and as a fully featured application gateway, can protect against API abuse via throttling and other configurable policies.
- Simva can quickly incorporate upstream improvements to its component modules; for example, should newer versions of LimeSurvey be released, minimal effort would be required to incorporate them into Simva, as changes would be limited to that particular container.

3.2 Workflow of a simple evaluation

To illustrate the comparative ease of evaluating a serious game with Simva, we now briefly explore the steps involved, assuming we are interested in evaluating a game with participants from a local high school and have an unrelated serious game available for use as a control group.

First, the experimenters would need to download and launch the installation script (github.com/e-ucm/simva-infra), following the instructions included in the repository. Given an environment where docker and docker-compose can be installed, this script downloads, configures and connects all relevant containers, launching a fully func-

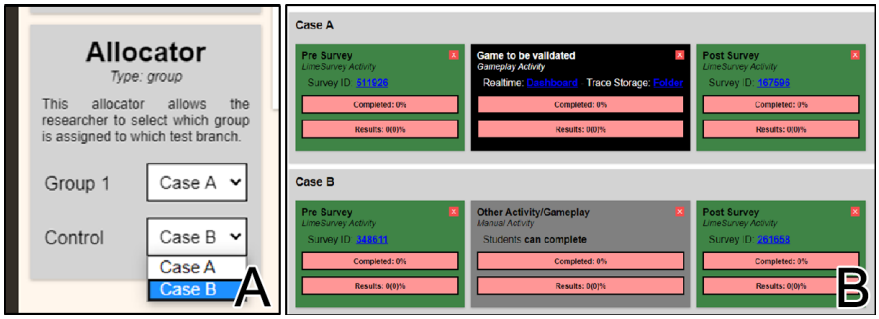

Fig. 2. Researchers can create multiple test branches and assign them to participants (A). Each branch can be assigned different sequences of activities (B) – in this case, the bottom-right row is the control group, while the top-right group plays the game under study.

tional Simva platform locally. Experimenters would next need to enter into Simva and configure the participant groups for the study; this can be as easy as requesting Simva to generate two sets of random tokens, one set for the intervention group and another for the control group. Experimenter would then download the token sets as printable PDF files, where tokens can be torn off to give them physically to each participant according to group; for a more in-depth discussion of token use to provide anonymization, see [11]. Studies are configured by determining participant groupings and assigning a sequence of activities to each group, as illustrated in Fig. 2. Examples of activities include playing a serious game or filling in a questionnaire. Fig. 3 illustrates how Simva users interact with the questionnaire module, provided by LimeSurvey but accessible through SSO as if it were built in. For this evaluation, experimenters could author two surveys, one for use as a pre-test and another for use as a post-test.

As participants log into the system through links (using their experimenter-provided token), they would be requested to fill in the questionnaire, to play the game that corresponds to their grouping only once the pre-test is complete, and, only after finishing the game, to fill in the post-test. Interaction between games and Simva is optional, and performed through an HTTP API; but if enabled, games can query whether participants with particular tokens have finished required activities before allowing the player to play; and can report player progress to Simva.

Finally, once an experiment is finished, data for each participant is available for download. In this case, experimenters would determine a scoring procedure for tests, and measure score increase from pre-test to post-test in both the experimental and control groups. Survey data can be downloaded as CSV or JSON files, while game interactions are available as JSON using xAPI-SG structure.

3.3 Cloud storage and access to GLA

As the amount of data to be analyzed increases, data-science tools are moving away from locally stored datasets to online ones. To enable the use of tools intended to analyze online datasets, the updated Simva architecture uses an S3-compatible storage. Amazon's Simple Storage Service (S3) is a de-facto standard for cloud storage; we use the open-source, S3-compatible Minio cloud-storage server for trace storage in Simva. This allows us to support the many data-science packages and libraries

Fig. 3. Screenshots illustrating the process of creating a new survey in *Simva*, using the *LimeSurvey* module: (A) launching the survey module to create a new survey; (B) survey authoring environment; and (C) choosing an existing survey to use.

that are already compatible with AWS S3, due to the frequent choice of S3 to host data lakes

Fig. 4 illustrates how Simva handles storage. Incoming traces from game interactions are queued into Kafka and stored by Minio for later retrieval. Use of Kafka, a scalable fault-tolerant queue, ensures that traces will not be lost, even in the presence of potential processing delays on the part of different modules of Simva, most notably complex analyses in RAGE Analytics. Simva includes a trace reallocator process that

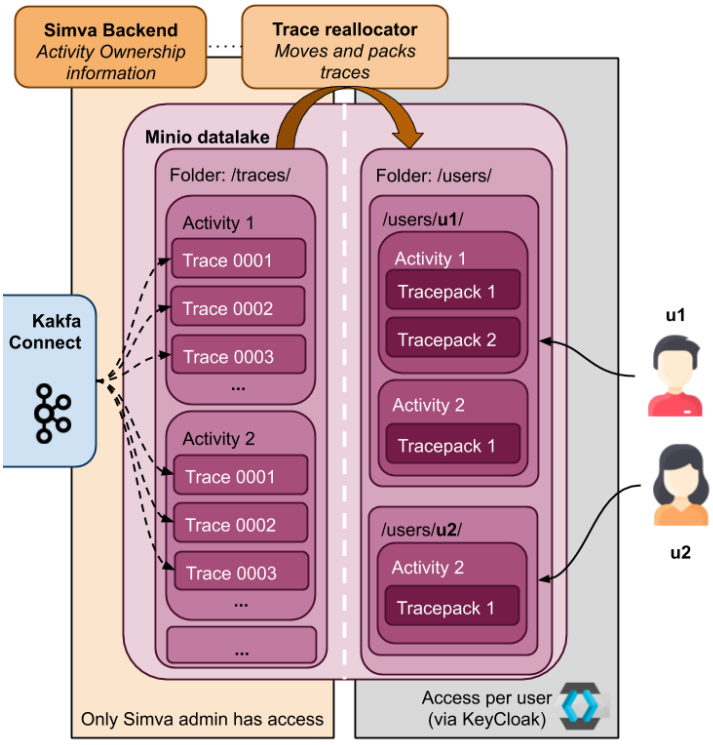

Fig. 4. Incoming xAPI traces are stored in Minio per-activity; and transformed using a trace reallocator to enable per-user access.

periodically builds downloadable versions of per-user traces. This allows traces to be stored per-activity but retrieved per-user with suitable access controls.

Access to interaction traces is targeted at external data science tools such as Jupyter Notebook, which have plugins that support S3 storages (see Fig. . 5). Jupyter allows quick generation and testing of hypotheses, for example segmenting users according to their in-game actions to determine the impact of player behaviors on learning. During preliminary tests of games, such rapid iterations are key to forming insights of what works and what does not, and to build better learning analytics models (LAMs; see [12]) for the final dashboards to be used in future versions of that game. For example, teachers often desire simple and understandable dashboards to

see what is going on during gameplay sessions, particularly if the games are used as a class activity. To create these dashboards requires significant time and effort to build a data analysis pipeline; if real-time updates are required, this would require stream-analysis technologies such as Apache Spark, Apache Storm, or Kafka Connect, which are hard to prototype with. A usual approach is therefore to start by applying a data science approach to perform coarse-grained analysis, establishing and testing hypothesis in quick succession. Once finished, it is much simpler to go to a data-engineering approach that applies streaming technologies to build real-time dashboards.

Once the game is mature, game learning analytics can be used to predict knowledge gain without using pre-post games at all [13]. Indeed, the use of data-science tools with serious games is most often related to prediction tasks, according to a recent literature review [14].

4 Conclusions and future work

Streamlining the evaluation of serious games can greatly reduce the efforts required to perform such evaluations, lowering the cost and risk associated with serious game deployment. We have described improvements to the Simva serious game validation platform that further lower the associated costs, by moving Simva to a containerized architecture based on solid open-source modules, and by adopting cloud storage for GLA data, which is readily accessible from existing cloud-based data science tools.

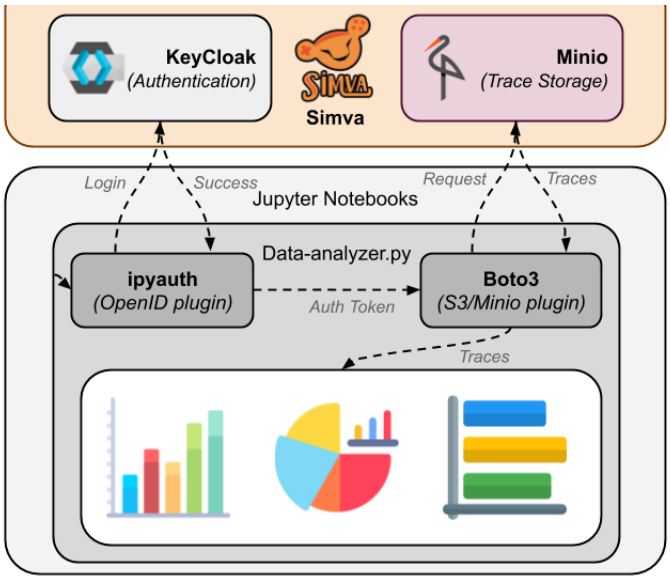

Fig. 5. Researchers using Jupyter Notebooks or other S3-compatible data-science tools can access the traces stored in Simva (Minio) by authenticating against KeyCloak and using the returned access token to fetch the data from Minio.

We described how a typical evaluation experiment for a serious game with the updated Simva architecture would work, highlighting the main advantages it provides to experimenters interested in evaluating or improving such a game. As a working tool, the new version of Simva will be used to illustrate the evaluation workflow during the following semesters in several graduate and post-graduate courses on Serious Games and E-Learning Technologies in the Complutense University in Madrid, Spain.

However, development of Simva is still ongoing. In particular, we are working on adding support for IMS' Learning Tools Interoperability (LTI) [15], to enable integration of Simva-managed activities into LTI-supporting LMSs such as Moodle, and to facilitate the development of Simva-compatible plugins and games. Simva would therefore act as both an LTI Tool Consumer, to host/interoperate with games and plugins; and as a Tool Provider, to be accessible from LMSs as an additional activity. This would remove two large hurdles for running experiments with serious games in LMS-equipped classrooms: it would remove the need to pre-configure games with Simva activity-ids (as the games would receive this information via LTI); and it would allow launching of Simva-managed experimental activities from within the LMS as one more task, allowing one-click launching of activities by relying on LMS-supplied information for authentication and group configuration.

Although the present work has been focused on applying Simva to evaluate games, it can also be used to evaluate any other educational tool that can be integrated as an activity, for example with IMS LTI; and more so if it can generate xAPI statements, such as H5P or Articulate content.

Acknowledgements

This work has been partially funded by Regional Government of Madrid (eMadrid P2018/TCS4307), by the Ministry of Education (TIN2017-89238-R), by the European Commission (Erasmus+ IMPRESS 2017-1-NL01-KA203-035259) and by the Telefonica-Complutense Chair on Digital Education and Serious Games.

References

1. Boyle, E.A., Hainey, T., Connolly, T.M., Gray, G., Earp, J., Ott, M., Lim, T., Ninaus, M., Ribeiro, C., Pereira, J.: An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. Comput. Educ. 94, 178–192 (2016). https://doi.org/10.1016/j.compedu.2015.11.003.
2. Calvo-Morata, A., Alonso-Fernández, C., Freire, M., Martínez-Ortiz, I., Fernández-Manjón, B.: Serious games to prevent and detect bullying and cyberbullying: a systematic serious games and literature review. Comput. Educ. (accepted), (2020).
3. Advanced Distributed Learning Initiative: Experience API, https://github.com/adlnet/xAPI-Spec/releases/tag/xAPI-1.0.3, last accessed 2020/07/10.

4. IMS Global Learning Consortium Inc.: Caliper Analytics Specification, version 1.1, https://www.imsglobal.org/sites/default/files/caliper/v1p1/caliper-spec-v1p1/caliper-spec-v1p1.html, last accessed 2020/07/10.
5. Serrano-Laguna, Á., Martínez-Ortiz, I., Haag, J., Regan, D., Johnson, A., Fernández-Manjón, B.: Applying standards to systematize learning analytics in serious games. Comput. Stand. Interfaces. 50, 116–123 (2017). https://doi.org/10.1016/j.csi.2016.09.014.
6. Freire, M., Serrano-Laguna, Á., Iglesias, B.M., Martínez-Ortiz, I., Moreno-Ger, P., Fernández-Manjón, B.: Game Learning Analytics: Learning Analytics for Serious Games. In: Learning, Design, and Technology. pp. 1–29. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-17727-4_21-1.
7. José, P.I., Manuel, P.V., Iván, M., Manuel, F.: Simplifiying Serious Games Authoring and Validation with uAdventure and SIMVA. 106–108 (2020). https://doi.org/10.1109/ICALT49669.2020.00039.
8. RAGE Consortium: RAGE Realizing an Applied Gaming Eco-System, http://rageproject.eu/, last accessed 2020/07/10.
9. Calvo-Morata, A., Rotaru, D.C., Alonso-Fernandez, C., Freire-Moran, M., Martinez-Ortiz, I., Fernandez-Manjon, B.: Validation of a Cyberbullying Serious Game Using Game Analytics. IEEE Trans. Learn. Technol. 13, 186–197 (2020). https://doi.org/10.1109/TLT.2018.2879354.
10. Alonso-Fernandez, C., Perez-Colado, I., Calvo-Morata, A., Freire, M., Martinez-Ortiz, I., Fernandez-Manjon, B.: Applications of Simva to simplify serious games validation and deployment. IEEE Rev. Iberoam. Tecnol. del Aprendiz. 1–1 (2020). https://doi.org/10.1109/RITA.2020.3008117.
11. Alonso-Fernández, C., Perez-Colado, I.J., Calvo-Morata, A., Freire, M., Martinez-Ortiz, I., Fernández-Manjón, B.: Using Simva to evaluate serious games and collect game learning analytics data. In: LASI Spain 2019: Learning Analytics in Higher Education (2019).
12. Perez-Colado, I., Alonso-Fernandez, C., Freire, M., Martinez-Ortiz, I., Fernandez-Manjon, B.: Game learning analytics is not informagic! In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 1729–1737. IEEE (2018). https://doi.org/10.1109/EDUCON.2018.8363443.
13. Alonso-Fernández, C., Caballero Roldán, R., Freire, M., Martinez-Ortiz, I., Fernández-Manjón, B.: Predicting students' knowledge after playing a serious game based on learning analytics data: A case study (in press). J. Comput. Assist. Learn. (2019). https://doi.org/10.1111/jcal.12405.
14. Alonso-Fernández, C., Calvo-Morata, A., Freire, M., Martínez-Ortiz, I., Fernández-Manjón, B.: Applications of data science to game learning analytics data: A systematic literature review. Comput. Educ. 141, 103612 (2019). https://doi.org/10.1016/j.compedu.2019.103612.
15. IMS Global Learning Consortium, I.: Learning Tools Interoperability Core Specification, Version 1.3, https://www.imsglobal.org/spec/lti/v1p3/, last accessed 2020/07/10.