A Prototype of an eLearning Platform in Support for Learning Analytics and Gamification

Conference Paper · September 2016

DOI: 10.1109/INCoS.2016.33

4 authors:

David Gañán Jiménez
Universitat Oberta de Catalunya

Santi Caballé
Universitat Oberta de Catalunya

Robert Clarisó
Universitat Oberta de Catalunya

Jordi Conesa
Universitat Oberta de Catalunya

A Prototype of an eLearning Platform in Support for Learning Analytics and Gamification

David Gañán, Santi Caballé, Robert Clarisó, Jordi Conesa
Department of IT, Multimedia, and Telecommunication, Open University of Catalonia,
Barcelona, Spain
{dganan, scaballe, rclariso, jconesac}@uoc.edu

Abstract— **This paper presents the implementation and prototyping of an innovative web-based eLearning platform, featuring Learning Analytics and Gamification called ICT-FLAG. A previous contribution presented the analysis and design of the platform. Following the design, this paper implements the platform and reports on the first experiences of connecting and integrating the platform with a real eLearning tool through an API. The purpose of this connection is to provide the tool with learning analytics and gamification-based services. The research reported in this paper is currently undertaken within the research project "Enhancing ICT education through Formative assessment, Learning Analytics and Gamification" (ICT-FLAG) funded by the Spanish Government.**

Keywords—eLearning platform, software development framework, web technologies, software analysis and design.

I. INTRODUCTION

The main goal of this work is to report the implementation and evaluation phases of the construction of an eLearning platform aiming to provide learning management systems (LMS) with innovative services in terms of learning analytics and gamification. The development of this platform is the main technological goal of a research project called ICT-FLAG, which is currently undertaken in the context of university degrees in the area of Information and Communication Technologies (ICT). In this context, the ICT-FLAG platform aims to design and build eLearning tools and services that can benefit students, lecturers, managers and academic coordinators. This impact will be mainly achieved through three innovative pedagogical and technological axes:

- Formative Assessment Tools (FAT) that can provide immediate feedback by means of automatic assessment [1][2].
- Learning Analytics (LA) as a service that monitors the activity and progress of the on-line teaching and learning processes supported by e-Learning systems and applications, combining this information with other sources of academic and historical information.
- Gamification as an incentive scheme in order to motivate students to practice more frequently, more collaboratively and increase their engagement in the learning experience [3][4][5].

In addition to dealing with multiple components and services, the ICT-FLAG platform needs to support a variety of virtual campus platforms, given that universities have a wide range of alternatives, from closed-source in-house solutions to open-source platforms, such as Moodle.

Considering the above goals of the ICT-FLAG project, a modular architecture with the following features is required:

- Customization. Simple selection of which components are deployed in each installation.
- Distribution. Enable the distribution of components along different installations.
- Extensibility. Enable the easy extension of an installation with new components.

This modular architecture also needs to consider some additional architectural and application features, required to ensure extensibility and the correct performance of the system, such as [6]: dependency injection, pluggability, distribution of modules, events / messages between modules, web support (and especially REST [7]), testing support, easy deployment and configuration, and portability.

In the current status of the project, the eLearning analytics functionalities of the system were already designed. The aim of this paper is to present the implementation details and issues, and the evaluation of the constructed part of the system.

The remainder of the paper is structured as follows: Section 2 reviews some previously work on the development of learning analytics and gamification systems, as well as summarizes previous work in the context of this research project. The details of the implementation of the system are presented in Section 3 while the corresponding evaluation and validation are explained in Section 4. Finally, Section 5 summarizes the main ideas and outlines the following steps in the development of the ICT-FLAG platform.

II. BACKGROUND

In this section, we discuss on the available tools and frameworks for applying LA and gamification in an educational context. Moreover, the state of the art in e-assessment tools is reviewed in relation to LA and gamification. Finally, we discuss how this previous work has influenced the requirements and the technical design of the ICT-FLAG platform.

A. Learning Analytics

Historically, Learning Analytics (LA) [8][9][10] appears from research about the processing, analysis and visualization of knowledge about the learning process. This knowledge may be gathered from large data sets of events at different levels of abstraction, such as the interactions of students with learning

978-1-5090-4124-4/16 $31.00 © 2016 IEEE
DOI 10.1109/INCoS.2016.33

management systems, other students and instructors throughout the learning process [11].

In this sense, LA is built upon previous works [10] in which the interactive processing of learning data has been exhaustively analyzed [12][13]. There are many results and applications in education, such as providing effective assessment and automatic feedback to students; supporting instructors by monitoring and predicting student performance; and modeling user profiles for content and process customization, according to the student needs, goals and individual skills.

Therefore, LA continues to evolve on interactive LA so as to apply them both in traditional eLearning systems and in new massive online course platforms (MOOCs) [13], where interaction data is generated at large or at very large scale. Large-scale data collection, processing and analysis using suitable models and reasoning can improve the use of existing eLearning systems, as well as improving the system themselves [14].

Recently, other relevant topic considered is the development of specifications/standards for providing a common data model for representing student interactions, which therefore allows for improving content interoperability [15]. Some of these approaches have addressed the problem of student performance [16]: i) The IEEE Standard for Learning Technology standards' family and ii) The Experience API (or xAPI or Tin Can API). The first provides a data model for tracking and exchanging information of student interactions with learning content [17] and a communication specification that allows the LMS to query collected information [18]. The second, the Experience API is a specification for collecting and managing eLearning activities, enabling the communication of eLearning information between different LMS and tools.

In this way, LA has addressed classical problems in eLearning, solving them at least partially [19]. However, there is a lack of integral solutions that can support all the previously mentioned aspects and incorporate them openly and transparently into current LMSs. Moreover, even though there are some common data models, their adoption for the e-Learning community is still scarce. As a consequence, each system tends to use its own model to represent student interactions, which hinders the construction of a LA model that manages information from different sources.

B. Gamification

"Gamification" is a term used to describe the use of game-based concepts and techniques outside recreational activities, with the goal of increasing the motivation of the participants and improving the results [20]. The benefits of gamification are an increase in motivation and engagement, which can be applied in education or work-related contexts [21]. In this sense, there are previous experiences where gamification techniques have been applied to both formal and informal learning scenarios [22][23].

On one hand, gamification offers an interesting collection of resources to increase learning engagement and motivation, in terms of achieved goals, such as reward mechanisms (scores, leaderboards, prices, achievements), progress logs ("to-do" and "done" tasks), social aspects (encouraging communication, reputation systems), feedback mechanisms and status notification (auditory or visual alerts) [22]. For example, a very simple gamification resource is an achievement system, a set of virtual badges that are awarded when a goal is achieved [24][25].

On the other hand, though promising, gamification is still an emerging discipline without a strong theoretical foundation. For instance, there are no commonly accepted theoretical frameworks or general principles on how to apply gamification strategies to a specific problem and there is little empirical validation [21]. Hence it is necessary to consider the context of a particular problem in order to choose the best strategy for applying gamification strategies [26].

C. E-Assessment systems and tools based on LA and gamification

Most Formative Assessment Tools (FATs) offer some type of reporting mechanisms to visualize student activity and performance. However, FATs use custom mechanisms to capture, store and report information. This level of customization allows for the definition of informative metrics that are well-suited to the learning process and the type of educational activities. On the other hand, it is hard to reuse these processes or to reuse the gathered information outside of the scope of a single FAT. For instance, it is hard to aggregate information from several tools along an academic program. In contrast, ICT-FLAG offers a generic solution that can be applied to a variety of tools with little implementation effort and can combine information from several sources.

LA can also consider additional information beyond that offered by a FAT, e.g. information provided by a LMS. Some well-known LMS, such as Moodle, have APIs that enable the interoperation with LA tools and the creation of plug-ins to embed LA functionalities [27]. LMS can also be used to incorporate gamification elements in the learning process [28], [29]. Finally, LA and gamification can be applied in the context of MOOCs [30][31][32]. However, existing LA tools are very restricted to the types of data that can be collected and analyzed, and usually the data visualization techniques are developed separately from the rest of the process of collection and analysis. Finally, the usability of these tools is limited, and because of its sophistication, is particularly complex for teachers and non-technical users to learn how to use and exploit them.

Regarding the inclusion of gamification in the current tools for e-assessment purposes, its use is still in its infancy and is based more on methodological aspects than technological ones. The inclusion of badges, for example, to reward the successful completion of an activity, does not correspond to a particular technological development [33].

D. Previous work

The ICT-FLAG system presented in this work is currently evolving. The decisions about the suitable technologies for the construction of the system were already discussed in [34]. The research concluded that the best technology for the construction of the system is Java EE (Java Enterprise Edition). Moreover, the analysis and design of the system was presented in [35] through the design methodology and the system's architectural view from different perspectives.

III. Implementation Details

This section explains the implementation details and issues of the data flow required for the LA features of the system. This data flow consists of a sequence of phases or steps that manipulate data, concretely:

- Data collection from a FAT.
- Data consistency checking.
- Data processing.
- Data storage.
- Data reporting and visualization.

Each of these steps is described in the next subsections.

A. Data collection from a FAT

The relevant data for LA is produced by FATs. These tools can provide information about the activity of a student while solving exercises as well as the assessment of the final student submission.

The system provides a REST API [7] that allows FATs to send data about assessment information. Currently, this API supports data for two types of elements: *assignments* and *exercises*. An *assignment* is a collection of exercises that a student should solve in order to receive a mark. *Exercises* are individual activities the student can execute inside the FAT, which can be a part of an assignment or can be simply for practice. The result of an exercise is an *outcome* ("right" – correct submission, "wrong" – incorrect submission, "timeout" - the student's submission could not be checked as it required too many resources or "error" – the student's solution triggered an error in the FAT) and optionally a *mark* which provides further information about the student's success or failure.

For each element, the system expects to obtain three different events or messages: *start, end and assessment*. The *start event* indicates when the student starts the assignment or exercise. The *end event* should be sent when the student ends the activity or the exercise (generally the submission of the exercise is considered its end). Finally, the *assessment event* allows for sending data about the mark obtained in that assignment or exercise. It is defined as a different event because the mark may not be available immediately at the end of the assignment or exercise. In the case of exercises, the assignment event is not required, provided that not all of FATs evaluate exercises with a mark (some of them only use an outcome, e.g. if the activity is a multiple-choice quiz).

The API expects other parameters about the context of the submitted information (See Table 1). Some of them are mandatory, such as the student or the tool that submits the information. Others are interesting in order to classify data but not mandatory (e.g., the subject or the program where the assignment or exercise belongs to).

Table 1. Types of parameters of the ICT-FLAG API.

name	**M/O**	**Description**
timestamp	O	The date and time when the event has occurred. If this parameter is null, the current date and time are chosen as timestamp
host	M	The URL of the ICT-FLAG system.
user	M	The username who delivers the assignment fact.
tool	M	The code of the FAT in the ICT-FLAG system
toolUUID	M	The unique identifier of the FAT
programCode	O	The code of the academic program where the assignment / exercise is performed.
subjectCode	O	The code of the subject where the assignment / exercise is applied.
semester	O	The code of the current semester in format YYYY-S, where YYYY is the year and S is the semester (1 or 2). Examples: 2014-2, 2015-1, 2015-2.
assignment / exercise	M	The code of the assignment or exercise as identified in the FAT. Exercises that do not belong to any assignment can skip this parameter (set null).
outcome	M	Outcome of the tool to the exercise submission. There is an enumeration called Outcome to provide this value.
score	M	The score of the assignment or exercise in a 0 to 10 scale.

In order to decrease the development effort needed to adapt FATs for communicating with the ICT-FLAG system, a helper library called ClientLib is provided, which facilitates sending and receiving data from the platform, making the implementation details transparent to the FAT authors.

B. Data consistency checking

Once the data arrives to the system through the REST API, it has to be checked for consistency errors that may make it useless or unreliable. These sanity checks are required because ICT-FLAG offers its services to a variety of FAT tools, whose developers may not be part of the ICT-FLAG project and unaware of how to properly use the API to submit data. These errors should be detected in order to avoid the corruption of the data in the repository.

Some sample errors are the following:

- The tool id does not exist or it is not provided.
- The action is not valid or not provided.
- The student username is not provided.
- The assignment identifier is not provided (in an assignment event).
- The exercise identifier is not provided (in an exercise event).
- The grade is not provided (in an assessment event).
- The grade value is invalid.

There are other potential errors or missing information that can be solved using some kind of fixes, which are explained in the following section.

Once the data is verified, it is stored in its raw format in a staging area, waiting its incorporation to the data warehouse.

C. Data processing

In order to effectively present the recovered information to the user, the data must be validated, integrated and extended with some metrics. In terms of Business Intelligence (BI), this step is known as an Extraction, Transformation and Load (ETL) process [36].

We consider two alternatives for implementation of the ETL process: immediate and deferred. The immediate approach consists of executing the ETL process each time a new API call arrives from a FAT. Such process incrementally updates the data repository (known as Data Warehouse or DW for short), with the new data. Then the reports show the data available from that DW (see Fig. 1).

Figure 1. Immediate ETL approach.

The deferred approach differs from the immediate in that the ETL process is not done immediately when the data is received, but asynchronously, for example each certain time interval predefined (see Fig. 2).

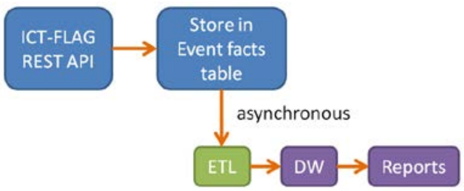

Figure 2. Deferred ETL approach.

On one hand, the immediate approach allows the reports to provide real time information, but the required resources may affect the response time for requests to the ICT-FLAG API. On the other hand, the deferred approach does not delay FAT requests but provide past data, which can be more or less problematic according to the delay time and the environment.

In order to benefit from the advantages of both options and minimize their disadvantages, a streaming solution is proposed. Requests sent from the FAT will be stored in a queue (avoiding delays for the processing of that data), and there will be a background thread processing all the items on that queue asynchronously (allowing data to reach the reports as soon as possible). The resulting approach is shown in Figure 3.

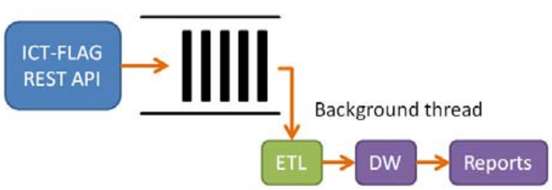

Figure 3. Proposed ETL approach.

Independently of when the ETL process is performed, it will consist of processing the data available one by one, applying some transformations, principally: *cleanup*, *fix-ups* and *calculations*. The *cleanup* consists of cleaning some data, such as unifying null values and empty strings. *Fix-ups* are changes that can be done on the data for fixing missing information. We consider two types of fix-ups: little or acceptable fix-ups, and big or flawed ones. Data with big or flawed fix-ups will not appear on reports in order to avoid adding noise to the results, but they can be consulted for example to evaluate the efficacy and to improve the FAT integration with the system. Finally, *calculations* are new data that can be inferred from the data being processed and other data already existing in the DW. The following table summarizes the most important transformations of each type.

Table 2. Main ETL transformations.

Cleanups
• Program is null or empty ➔ unify to null. • Subject is null or empty ➔ unify to null.
Acceptable fix-ups
• Date time null or empty or not valid ➔ use current time. • User does not exist ➔ the user is created. • Program does not exist ➔ set to null. • Subject does not exist ➔ the subject is created. • Semester is null or empty or not valid ➔ use the current semester.
Flawed fix-ups
• Outcome is null or empty or not valid in an exercise submission ➔ set outcome = 'error'. • Grade is outside the range 0..10 ➔ do nothing but mark it as wrong value.
Calculations
• *Course* and *course group* may be calculated if the subject and semester is informed, and the user is registered as a member of some course group for that subject and semester. • *Duration* of the assignment or exercise can be calculated as the time passed between the start and end events. • *Time from last attempt* can be calculated as the time passed between the end time of the last attempt and the start time of the new one (if any). • *Attempt number* is 1 if there is no previous attempt or the last attempt number plus 1 instead.

As we mentioned before, the expected sequence of events is: start – end – assessment, but the ETL process is also prepared to fix-up some situations where events do not occur or do not arrive in the expected order (for example, if the student does not submit an started exercise, or there is a communication problem, etc.). Some of these fix-ups are considered flawed and will not appear by default in reports. The following table summarizes the fix-ups applied to each scenario.

Table 3. Fix-ups applied to solve events ordering issues.

If action is 'start'
• No problem is reported. • A new instance of Exercise / Assignment Data is created. • EndDate is left null so it will not appear in reports yet.
If action is 'end'
• X is the last Exercise / Assignment Data matching with the new data, which has endDate = null. • If X exists ➔ X is updated with the new data. • If X does not exist ➔ a new row is created with the new data, startDate = endDate and will not appear by default in reports (as this is a flawed fix-up).
If action is 'assessment'

- X is the last Exercise / Assignment Data matching with the new data, which has grade = null.
- If X exists → X is updated with the new data.
- If X does not exist → a new row is created with the new data, startDate = endDate = assessmentDate and will not appear by default in reports (as this is a flawed fix-up).
- In any case, if endDate = null, the assessment date will be set as endDate, although this will not be considered as a flawed fix-up.

D. *Data storage*

The previous ETL process stores the processed data into the DW, whose structure (see Fig. 4) defines two main entities: one for storing exercises information (*ExerciseDW*), and another for storing information about assignments (*AssignmentDW*). All the related entities allow for contextualizing the data of different items: the program and subject of the assignment / exercise, the tool which generated the information, the student who did the exercise, etc. All this information will be also useful to categorize, filter or group information at the reporting level.

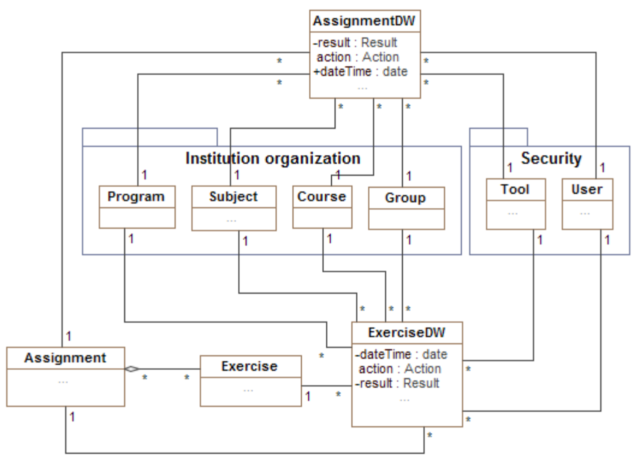

Figure 4. eLearning analytics DW structure.

E. *Data reporting and visualization*

The last phase in the dataflow sequence corresponds to the data visualization. In order to show the data to the user (typically program directors, lecturers, instructors or even students) in an effective way, a set of reports and dashboards should be created.

There are several options to define and implement reports, but we can summarize them in two categories: use a pre-built solution or 'make it your own'. From the huge quantity of options available in the market, we chose four candidates, two pre-built solutions (Pentaho Community Edition[1], or Pentaho CE for short, and BIRT[2]), and two diagram features for the custom solution (JSF Primefaces[3] and the D3[4] JavaScript library). Table 4 shows the advantages and drawbacks we identified in each solution in the context of the ICT-FLAG platform.

Pre-built solutions save a lot of implementation effort and facilitate integration and security measures because they provide their own security mechanisms. However, we discarded using Pentaho CE solution due to its difficulties to embed the reports into the ICT-FLAG application. BIRT looks like a nice reporting solution but seems more suitable for defining static reports than dynamic ones. In addition we found a lack in the documentation about how to embed BIRT reports into web applications. BIRT has been discarded due to these inconveniences.

In custom solutions all the reporting facilities (e.g., query parametrization, sorting, filtering, pagination, and so on.) should be fully implemented from scratch. Fortunately, there are some libraries that facilitate the visualization of the report such as *D3* and *Primerfaces JSF*. As we do not require so complicated types of charts we selected the *Primefaces JSF library*, principally because it is very easy to integrate into the Java EE application. Furthermore, should we require more complex graphs in the future, we can opt to use *D3* without having to change anything, given that they both are compatible.

Table 4. Advantages (+) and drawbacks (-) of each reporting solution.

Pentaho
+ It is one of the most used BI solutions.
+ Dashboards and reports can be defined quickly, without writing large amounts of code.
+ Provides a good quantity of predefined types of charts and tables.
- The Community Edition has many issues.
- Requires its own installation.
- Difficult to integrate transparently.
- Security permissions do not satisfy the project requirements (it is difficult to set application security filters).
BIRT
+ Reports can be defined graphically.
+ There are different types of charts and other report elements available.
- Difficult to integrate and apply application security filters.
- Embedding reports in web application is not trivial.
JSF Primefaces chart capabilities
+ Easily integrated in JavaEE applications.
- A limited number of charts and tables available (more can be adapted but with extra effort).
d3 (or other JavaScript libraries)
+ Huge number of available charts.
- More difficult to integrate (but possible).

Once the appropriate technology for the implementation of the reports has been decided, the content of each report should be defined. We considered two approaches to define reports:

- General reports with dynamic functionality.
- Static and specific reports.

General dynamic reports are better for users who intend to analyze data and find patterns. They allow for changing report parameters dynamically so that the user can 'play' with the data. Since not all users have enough knowledge to work with or understand dynamic reports, mostly static and specific

[1] http://community.pentaho.com/
[2] http://www.eclipse.org/birt/
[3] http://primefaces.org/
[4] https://d3js.org/

reports have been implemented. Even static, the implemented reports allow for some dynamic changes, such as filtering or sorting capabilities.

For the dynamic report option, two types of reports were defined: for exercises and for assignments. Each type of report shows data grouped by different concepts, and the user can change grouping concepts, filter by different properties and sort data results. The data charts provided automatically adapt their content when the user applies the new configuration (see Fig. 5).

Figure 5. Dynamic report.

On the other hand, four different perspectives are considered for the static reports:

- Exercise: considers the success rate, number of attempts and time spent on a given exercise.
- Activity: aggregates the exercise information and includes information on the grades.
- Course: aggregates the exercise and assignment information at the course level.
- Student: tracks the performance of individual students along the course.

A summary of the indicators shown in each report is defined in Table 5.

Is worth mentioning that both dynamic and static reports will be filtered by user role and his relationship with the institution, in order to show only the information that he has permission to visualize.

Table 5. Indicators of static reports.

Exercise perspective
• Number of students that have worked on the exercise (total / correct / incorrect / failure): amount + percentage • Results of students in their first attempt (or first N attempts, e.g. N=3) • Average number of attempts until getting a correct answer • Average time between attempts • Average time from start until end
Assignment perspective
• Grade distribution • Aggregate results at the exercise level (average for all exercises) • Show the N worst exercises in terms of: o Lowest success rate o Longest time o Highest number of attempts
Course / Subject
• Number of students • Aggregate information at the exercise and assignment level (averages) • Historical trends (i.e. show data for current and previous semesters) • Show the N worst exercises in terms of: o Lowest success rate o Longest time o Highest number of attempts
Students
• Show the N students with: o Highest failure rate o Less than K attempted exercises o Less that K correctly solved exercises

IV. Experimentation and Validation

A representative group of two testers formed by a lecturer (i.e., researcher in e-learning), and a skilled technician user from the Open University of Catalonia performed the integration of a FAT tool (called *VerilUOC* [37]) with the ICT-FLAG system. As part of the test, testers integrated the ClientLib library into the FAT tool and sent data from it to the system through its APIs. The results obtained were used to evaluate the ClientLib in terms of usability, effectiveness and the quality of the existing documentation.

Four indicators of interest were selected and presented to the ClientLib testers in a questionnaire. The testers had to choose a score in the five-point Likert scale. Also, three of these indicators had an additional field to collect open comments:

1. Easiness of integration of the ICT-FLAG ClientLib into your FAT. Score on scale 0-5 and open comments.
2. The number of events you can send to the ICT-FLAG system and how this fits with your FAT requirements. Score on scale 0-5 and open comments.

3. The amount of information you can send of each event to the ICT-FLAG system and how this fits with your FAT requirements. Score on scale 0-5 and open comments.
4. The usefulness of the available documentation of the ClientLib. Score on scale 0-5 and open comments.

Table 6 shows some basic statistics of the quantitative marks on the scale 0-5 scored by all testers for each of the four indicators of interest considered. Table 7 shows an extract of qualitative results from those indicators with open comments provided by the testers after the test in questionnaires.

Table 6. Mean (M) and Standard Deviation (SD) score for each indicator in the scale of 0-5.

Testers	Indicators of interest (0-5)				
	#1	#2	#3	#4	Total (M)
# Lecturer	3	5	4	5	4,25
# Technician	4	5	5	5	4,75
Total M(SD)	3,5 (0,7)	5 (0)	4,5 (0,7)	5 (0)	**4,5 (0,35)**

From the quantitative results, we can see that in general most of the scores have the highest value. The lowest results were obtained in the first question about the easiness to realize the integration. According to the tester's comments (see Table 7), this punctuation is related with a version conflict problem with some library dependencies, so this is an issue to be improved in future versions of the ClientLib.

Table 7. Excerpt of the questionnaires on the 4 commented indicators.

Indicators of interest with open comments	Tester's open comments (L:Lecturer, T:Technician)
Easiness of integration of the ICT-FLAG ClientLib into your FAT.	"There were some incompatibility problems with internal JAVA libraries (different versions)" (L) "Some problems can appear if there is any conflict between the version of the libraries referenced by the ClientLib and the FAT." (T)
The number of events you can send to the ICT-FLAG system and how this fits with your FAT requirements.	"The ICT-FLAG library has the sufficient events to collect information from the VerilUOC platform." (L) "The number of available events is enough, and can be extended in the future for adapting to new FAT requirements." (T)
The amount of information you can send of each event to the ICT-FLAG system and how this fits with your FAT requirements.	"The amount of information is very large. However, it is difficult to reduce the number of parameters without losing precision. Maybe a container class with reduce the complexity to call events." (L) "The amount of information is enough, and can be extended in the future for adapting to new FAT requirements." (T)
The usefulness of the available documentation of the ClientLib.	"Enough explanation. I will recommend to put some examples on future releases." (L) "The documentation is useful and correct." (T)

Testers also suggested some enhancements that will be considered in future versions of the ClientLib, such as the encapsulation of all the data in a container class to reduce the number of parameters of each call, or the addition of more examples in the documentation.

In summary, all the indicators scored on, or over, the average 0 – 5 (see Table 6). In particular, indicators #2, #3 and #4 were well scored by all testers who acknowledged that the ClientLib is in general useful for the integration of a FAT tool with the ICT-FLAG system.

V. Conclusions and Further Work

This paper presented the implementation details and issues of the development of a LA and gamification platform for eLearning called ICT-FLAG. All the steps of the data flow were presented, from the start of the event at the FAT to the data visualization, and the proposed solution for each step was described in detail.

We believe that the proposed solutions for efficient data flow ensure features, such as scalability, modularity or security, which are a primary concern in the project. Another relevant requirement for the system is the need to integrate the platform with a wide variety of eLearning tools such as automated tools for formative assessment or different learning management systems is also enforced by the definition of a client library that offers the ICT-FLAG API while hiding details about the platform. Other components such as the BI module allow ICT-FLAG to provide complex LA.

As ongoing work, we are about to testing the LA system developed in a variate of eLearning tools at larger scale. Furthermore, we currently work in the analysis and design of the gamification features of the platform, which will enable instructors to introduce gamification methodologies in their FAT with the minimal effort using our system. Finally, in order for our ICT-FLAG platform to ensure relevant features, such as data flow efficiency, scalability, modularity and security, which are a primary concern in our project, we plan to evaluate these quality features by software engineering practices.

Acknowledgments

This work was funded by the SMARTLEARN research group, and the Spanish Government through the project TIN2013-45303-P "ICT-FLAG" (Enhancing ICT education through Formative assessment, Learning Analytics and Gamification).

References

[1] Whitelock, D. (2007). Computer Assisted Formative Assessment : Supporting Students to Become More Re ective Learners. In 8th International Conference on Computer Based Learning in Science, CBLIS, pp. 492 - 503.

[2] Nicol, D., & Macfarlane-Dick, D. (2006). Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice. Studies in Higher Education, 31 (2), 199 - 218.

[3] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, pp. 9-15.

[4] Domínguez, A., Saenz-de-Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., & Martínez-Herráiz, J. J. (2013). Gamifying learning experiences: Practical implications and outcomes. Computers & Education, 63, 380-392.

[5] Amriani, A., Aji, A. F., Utomo, A. Y., & Junus, K. M. (2013). An empirical study of gamification impact on e-Learning environment. In

Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on, pp. 265-269.

[6] Yener, M., & Theedom, A. (2014). Professional Java EE Design Patterns. John Wiley & Sons.

[7] Mehta, B. (2014). RESTful Java Patterns and Best Practices. Packt Publishing Ltd.

[8] Siemens, G., & d Baker, R. S. (2012). Learning analytics and educational data mining: towards communication and collaboration. In Proceedings of the 2nd international conference on learning analytics and knowledge, pp. 252-254.

[9] Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. International Journal of Technology Enhanced Learning, 4(5-6), 304-317.

[10] Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40(6), 601-618.

[11] Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. Expert systems with applications, 33(1), 135-146.

[12] Fahy, P. J. (2006). Online and face-to-face group interaction processes compared using Bales' interaction process analysis (IPA). European Journal of Open, Distance and E-Learning, 9(1).

[13] Siemens, G. (2013). Massive open online courses: Innovation in education. Open educational resources: Innovation, research and practice, 5.

[14] Caballé, S., & Xhafa, F. (2013). Distributed-based massive processing of activity logs for efficient user modeling in a Virtual Campus. Cluster computing, 16(4), 829-844.

[15] Friesen, N. (2005). Interoperability and learning objects: An overview of e-learning standardization. Interdisciplinary Journal of Knowledge and Learning Objects, 1(1), 23-31.

[16] Del Blanco, Á., Serrano, Á., Freire, M., Martínez-Ortiz, I., & Fernández-Manjón, B. (2013). E-Learning standards and learning analytics. Can data collection be improved by using standard data models?. In Global Engineering Education Conference (EDUCON), pp. 1255-1261.

[17] IEEE Standard for Learning Technology--Data Model for Content Object Communication. IEEE Std 1484.11.1-2004 (R2010).

[18] IEEE Standard for Learning Technology-- ECMAScript Application Programming Interface for Content to Runtime Services Communication IEEE Std 1484.11.2-2003 (R2009).

[19] Isabel Guitart, Jordi Conesa. (2016). Creating University Analytical Information Systems: A Grand Challenge for Information Systems Research. In *Formative Assessment Learning Data Analytics and Gamification* (pp. 167-186). Elsevier. ISBN: 978-0-12-803637-2

[20] Salen, K., & Zimmerman, E. (2004). Rules of play: Game design fundamentals. Cambridge, MA: MIT Press.

[21] Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work?--a literature review of empirical studies on gamification. In System Sciences (HICSS), 2014 47th Hawaii International Conference on, pp. 3025-3034.

[22] Lee, J. J., & Hammer, J. (2011). Gamification in education: What, how, why bother?. Academic exchange quarterly, 15(2), 146.

[23] Borrás Gené, O., Martínez Núñez, M., y Fidalgo Blanco, Á. (2014). Gamification in MOOC: challenges, opportunities and proposals for advancing MOOC model. In F. J. García-Peñalvo (Ed.), Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 215-220.

[24] Denny, P. (2013). The effect of virtual achievements on student engagement. In Proceedings of the SIGCHI conference on human factors in computing systems, pp. 763-772..

[25] Hakulinen, L., Auvinen, T., & Korhonen, A. (2013). Empirical study on the effect of achievement badges in TRAKLA2 online learning environment. In Learning and Teaching in Computing and Engineering (LaTiCE), 2013, pp. 47-54.

[26] Groh, F. (2012). Gamification: State of the art definition and utilization. Institute of Media Informatics Ulm University, 39.

[27] Dimopoulos, I., Petropoulou, O., Boloudakis, M., & Retalis, S. (2013). Using Learning Analytics in Moodle for assessing students' performance.

[28] HENRICK, G. (2010). Gamification-What is it and What it is in Moodle.

[29] Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: a systematic mapping study. Educational Technology & Society, 18(3), 1-14.

[30] Greller, W., & Drachsler, H. (2012). Translating Learning into Numbers: A Generic Framework for Learning Analytics. Educational technology & society, 15(3), 42-57.

[31] Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2012). A reference model for learning analytics. International Journal of Technology Enhanced Learning, 4(5-6), 318-331.

[32] Gené, O. B., Núñez, M. M., & Blanco, Á. F. (2014). Gamification in MOOC: Challenges, opportunities and proposals for advancing MOOC model. In Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 215-220.

[33] Gibson, D., Ostashewski, N., Flintoff, K., Grant, S., & Knight, E. (2013). Digital badges in education. Education and Information Technologies, pp. 1-8. Sringer

[34] Gañan, D., Caballé, S., Clarisó, R., Conesa, C.(2016). Towards the Effective Software Development of an eLearning Platform Featuring Learning Analytics and Gamification. 4th International Workshop on Collaborative Enterprise Systems (COLLABES 2016). Accepted. To appear.

[35] Gañan, D., Caballé, S., Clarisó, R., Conesa, C.(2016). Analysis and Design of an eLearning Platform Featuring Learning Analytics and Gamification. 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS 2016). Accepted. To appear.

[36] Grossmann, W., & Rinderle-Ma, S. (2015). Fundamentals of Business Intelligence. Springer.

[37] Baneres, D., Clariso, R., Jorba, J., Serra, M., Experiences in Digital Circuit Design Courses: A Self-Study Platform for Learning Support. Learning Technologies, IEEE Transactions on, vol.7, no.3, pp.1-15, (2014) doi: 10.1109/TLT.2014.2320919