A Web Services-Based Application for LMS Data Extraction and Processing for Social Network Analysis

Julián Chaparro-Peláez(✉), Emiliano Acquila-Natale, Santiago Iglesias-Pradas, and Ignacio Suárez-Navas

Departamento de Ingeniería de Organización, Administración de Empresas y Estadística, Universidad Politécnica de Madrid, Madrid, Spain
{julian.chaparro,emiliano.acquila,s.iglesias}@upm.es,
i.suarezn@alumnos.upm.es

Abstract. The emergence of learning analytics as a discipline of its own has given way to a diverse subset of research fields offering very different approximations to the topic. One of the most recent and active approaches is social learning analytics, which focuses primarily on the application of social network analysis (SNA) techniques and visualizations to study and help understanding interactions in online courses as a key pillar of social construction of learning. However, and despite this interest, current tools for analysis and visualization are very limited for advanced social learning analytics, and SNA applications cannot directly process data from learning management systems. This paper presents a technical view of the design and implementation of a web services-based application that aims to overcome these limitations by extracting and processing educational data about forum interactions in online courses to generate the corresponding social graphs and enable advanced social network analysis on SNA software.

Keywords: Learning analytics · LMS · Social network analysis · Data extraction · Data processing

1 Introduction

In the past decade, widespread use of IT across organizations—both within the organization itself and across organizational boundaries—has led to the generation of a vast amount of data originating from business processes, known as big data. Big data can be defined as "high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization" [1], and the urgent need to manage these data has given birth to the emergence of disciplines such as business intelligence and business analytics. While business intelligence and business analytics have been constantly evolving in the past years [2], the application of analytics to the educational landscape has not been much of a concern for researchers and practitioners until recently, despite the high growth of Virtual Learning Environments (VLEs) and learning management systems (LMS) adoption for online and blended learning (e-learning and b-learning, respectively).

D. Palacios-Marqués et al. (Eds.): GIKA 2015, LNBIP 222, pp. 110–121, 2015.
DOI: 10.1007/978-3-319-22204-2_11

Despite early efforts from the educational data mining field—cf. Ventura & Romero [3,4]—only in the past five years has the educational community realized of the potential value of educational data analysis for the optimization of educational processes and, more specifically, for the improvement of learning. In this context, the celebration of the First Conference International Conference on Learning Analytics and Knowledge (LAK'11) formalizes the birth of a new discipline, learning analytics, which aims to translate the concepts present in business analytics to ICT-supported education [5]. Thus, a broad definition of the concept describes learning analytics as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs." [6].

Although early research on learning analytics focuses on the study of LMS logs for prediction of student success or at-risk student detection in online instruction, mostly relying on regression analysis—cf. [7]–, more recent research on this field has brought novel approaches that try to go beyond statistical analysis and offer other informative and decision-oriented perspectives, such as visualization–e.g., [8]—or social learning analytics–e.g., [9].

This study focuses on the latter, and more specifically on how to facilitate detailed social learning analytics from LMS data. In order to do so, a basic definition of the concept of social learning analytics will be provided in Section 2, along with a brief description of the most popular tools used for social learning analytics, and their advantages and disadvantages. From the description of these tools in section 2 we conclude that there is a need for development of new tools capable of LMS data extraction and processing for advanced social learning analytics. Section 3 will cover the design and implementation of such a tool—named GraphFES, a web services-based application—from a software engineering perspective, and will therefore describe the functional and software requirements, the system design process, system architecture, system implementation and operation. Finally, Section 4 will briefly illustrate the operation of the application with an example.

It must be emphasized again at this point that the main goal of this study is not to offer an in-depth study on social learning analytics, the use of social learning analytics tools, or the application of social network analysis to educational data, but rather to thoroughly describe the design and implementation of a tool that makes it possible to narrow the gap between educational data available in formal LMS and the tools that facilitate social learning analytics. Therefore, next section will only briefly cover and address relevant concepts that are necessary to comprehend the potential of GraphFES for education.

2 Social Learning Analytics

In formal learning environments, the learning process takes place generally in learning management systems (LMS). LMS allow students to develop individual, self-directed learning through access to learning resources—documents, videos, external links, etc.—and assessment instruments, such as quizzes or essays, but they also provide means to communicate with their peers and instructors to make up for the lack of physical contact and to make social construction of knowledge possible. This is especially critical in collaborative learning, where social learning is positioned at the center of the process. Social learning theory establishes that cognitive processes take place in a social context, by reciprocal interaction between behavior and controlling conditions, both individual and environmental [10], and that knowledge is created and constructed by the interactions of individuals in a given society [11]. This perspective brings attention to knowledge creation by participation and engagement in the discourse—e.g. communities of practice [12]. Course forums are an essential part of social learning in formal contexts—i.e. LMS—, and therefore understanding the dynamics of interactions in course forums is key to explain social learning.

Buckingham-Shum and Ferguson [13] make a distinction between inherently social analytics, which only makes sense in a collective context, and socialised analytics, as personal analytics that have new attributes in a collective context. Among inherently social analytics, they propose two strands of analytics focusing on interpersonal relationships—social network analytics—and language analysis for social construction of knowledge–discourse analytics–, respectively.

From this perspective, it is evident that explaining social learning from an analytics perspective requires novel approaches relying on social network analysis (SNA) of educational data. SNA helps understanding the social dynamics of a given group. In formal educational settings SNA focuses mainly in the interactions that take place in bounded spaces for collaboration, primarily course forums. The main uses of SNA of educational data include identification of relevant learning agents, such as at-risk students, knowledge brokers or influential students [9].

One of the most distinctive characteristics of SNA is that it may provide meaningful information in two different ways: analysis and visualization. Analysis focuses on calculation of SNA parameters and metrics for each node—see Freeman [14] for further information about centrality measures, and Hernández-García [15] (p. 156) for further information about SNA metrics for learning analytics—and network overall parameters. Visualization of social networks, on the other hand, provides graphical information to facilitate understanding of the different relationships among agents. The purpose of this study is to facilitate advanced SNA analysis and visualization of educational data, and thus the next section will give an overview of three popular tools to perform SNA of data from the leading open-source LMS, Moodle [16].

Tools for Social Learning Analytics

SNAPP is a web browser bookmark let that extracts information about message board activity from LMS, and then builds up the resulting social network in a Java applet. SNAPP displays social network graphs from LMS forum data by parsing forum contents—i.e. it does not use information from the LMS logs–and allows basic manipulation and filtering, as well as temporal representation and evolution of the graphs and graph export features. SNAPP also provides basic centrality SNA measures, such as degree, in- and out-degree, betweenness and eigenvector centrality, and network density. However, it requires advanced security permissions and lacks support for newer versions of Moodle.

Forum Graph is a Moodle's report plug-in that displays the resulting social graph from activity in a single forum. Forum graph is therefore integrated in Moodle and it is easy to install; nevertheless it does not allow filtering and graph manipulation, and it does not provide SNA metrics, limiting its suitability for advanced SNA.

Meerkat-ED is a Java application that can load forum and post information from Moodle backup files, and then extracts it and builds the resulting social graph. Meerkat-ED includes both social network analytics and discourse analytics capabilities. Regarding SNA, Meerkat-ED facilitates basic graph manipulation, capabilities to visualize temporal evolution and basic SNA metrics.

From the above, it is manifest that these tools, albeit useful for basic analysis, lack capabilities for advanced SNA: SNAPP and Forum Graph provide little information other than visualization of the network topology, but Meerkat-ED shows how external apps may improve analysis and visualization by separating the data layer from Moodle logs and the process and presentation layer done in the application.

Given these limitations, the next logical step is to consider the use of SNA specific software tools instead, such as Gephi, which do have the necessary functionalities in terms of visualization—including complex manipulation—and advanced analysis of SNA metrics. However, the way LMS store data is not adapted to the format used by SNA applications, and therefore data pre-processing is necessary. The following section presents our solution to this problem by using a web service-based application named GraphFES (Graph Forum Extraction Service) that extracts forum interaction data from Moodle logs and creates a graph that can be analyzed in Gephi, and gives a detailed explanation of the system design and implementation.

3 GraphFES: Design and Implementation

3.1 Software Requirements Specification and Project Design

Functional and Technical Requirements

The main objective of GraphFES is to extract data related to forum activity from Moodle logs and create the resulting social graph for SNA in Gephi. In order to make it independent of LMS changes, the first requirement is that GraphFES will be an external application. In Moodle, this means that it is necessary to create a local plug-in that implements external functions to extract database information. This ensures that structure and order of the LMS architecture is preserved. The local plug-in exposes these functions, but it also requires activation of web services in Moodle in order to make the functions accessible to external applications.

Once access to the database from an external application is possible, it is necessary to create said application, which must execute the queries to the database for data retrieval. After collection of all relevant data—only forum data—, the application must be able to process them and to create a social graph in a format that the SNA application—i.e. Gephi—can read; in this case, and in order to be able to use all of Gephi's potential, the format chosen is Gephi's native format: GEXF, an extensible markup language (XML). Apart from data extraction and processing, the application should also have a simple graphic user interface for user interaction.

Using web services to perform the database queries requires the use of HTTP requests to Moodle, otherwise data extraction would not be possible. Because data is transmitted in JSON, it is necessary that the application can handle JSON.

There are three additional desirable characteristics in the final system: 1) compatibility with different versions of the LMS, to avoid falling in the same problems as SNAPP; 2) client platform independence, so that it may be used in different operating systems; and 3) high processing speed, for increased performance.

Project Design

After identification of the main system requirements, the next step is to address the software project design. By reviewing the literature on software engineering, different design options are considered. The simpler one is the waterfall model [17]; the waterfall model structures the different stages of the software development in a sequential way. While the waterfall model is a good choice for small projects, it does not allow changes during the project and prototypes of the system are not available until the latest stages. Prototyping [18] helps overcoming these limitations, by defining the initial specifications for the creation of a prototype; the prototype is a basic product that includes only the essential characteristics of the final product. Prototypes give an idea of how the final product will look like, and allow revision of requirements for next iterations. While such an approach might seem adequate to the development of GraphFES, it might increase the duration of the project because GraphFES comprises different modules. The spiral model [19] is a generalization of prototyping that can shorten project duration, incorporating the sequential stages from the waterfall model but with a series of evolutionary deliverables that further refine the prototype after each iteration. Unfortunately, the spiral model is suitable for more complex projects, adding too much work overhead to small projects. Finally, iterative and incremental development [20] is the pillar of the Unified Process and other agile methodologies. Iterative and incremental development uses short iteration cycles that minimize risks in software development. The iterative and incremental development model is suitable to the design of GraphFES because it facilitates development of its different components along different iterations, with different system prototypes that make it easier to evaluate functional and technical requirements, and introduce changes in case it is necessary.

GraphFES comprises different components but, since they are dependent, they should be implemented in a given order. The collective application model [21], useful for learning analytics purposes [22], has a cyclical nature that suits both the iterative and incremental development model and GraphFES, comprising of three stages: information gathering—select and capture—, information processing—aggregate and process—and information presentation—display. Therefore, the first development iteration must cover the local Moodle extension because it handles data extraction and delivery to the application. The second iteration should then cover the web application that collects the data and processes them to create the graphs. A final third iteration would cover system requirements revision and prototype changes. During the software implementation process, however, it was detected that from Moodle version 2.6 onward the log system had been changed, and therefore a new iteration was included to make the necessary changes to ensure compatibility of GraphFES with different versions of Moodle. The introduction of this new iteration confirms the suitability of the iterative and incremental model, which allowed redefinition of requirements and introduction of new characteristics seamlessly, and with low impact for subsequent iterations.

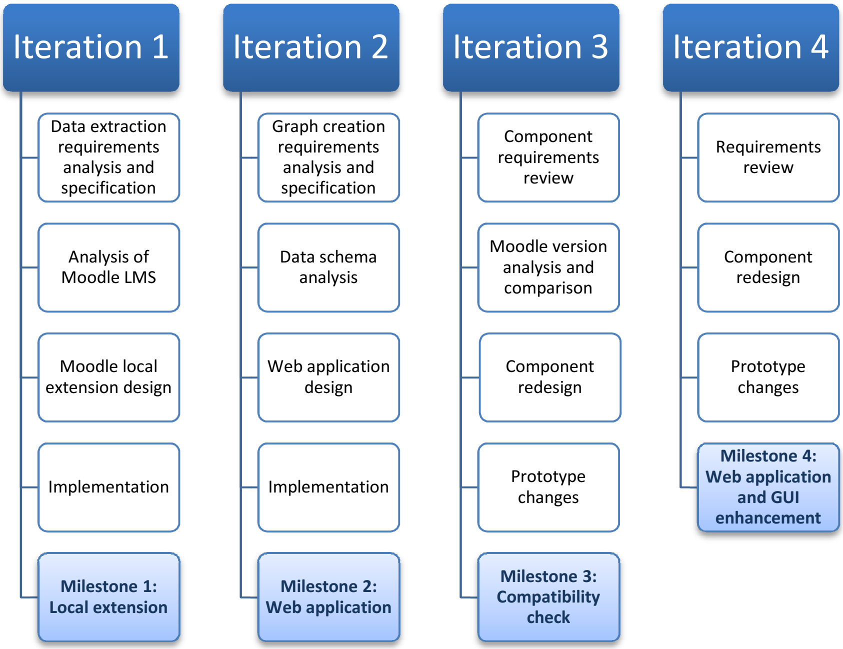

Fig. 1. GraphFES project design

3.2 System Architecture

Upon the project design explained in the previous section and summarized in Fig. 1, it is necessary to describe GraphFES architecture, comprising of two main components: Moodle local plug-in and web application (Fig. 2).

The local plug-in enables implementation of the two external functions that access Moodle database logs, both the old (*Legacy Log*) and new (*Standard Log*). These functions, together with other pre-installed functions, establish a communication using the REST protocol with the web application through a web service. The web application makes the data request to the LMS; this request is handled via the local plug-in, which gives a response with the necessary data using JSON. After data reception, the web application processes the data and generates the resulting social graphs in GEXF format, making it readily available for SNA using Gephi.

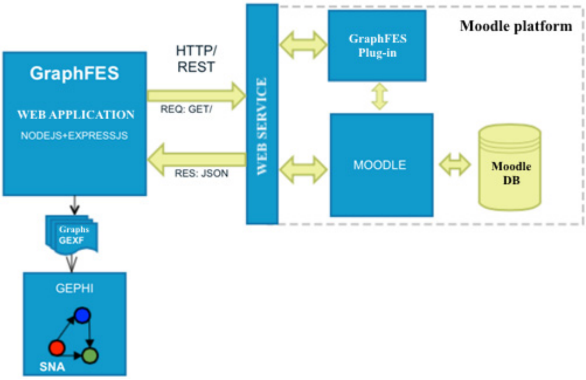

Fig. 2. GraphFES system architecture

3.3 System Implementation

Technology

Development of the local plug-in requires the use of PHP programming language —the same as Moodle, because it has to be included in an external folder in the LMS. There are two main components of the local plug-in: a file called *services.php*, that declares the external functions of the web service, and a file called *externallib.php*, which implements the external functions and performs the database query.

The web application, on the other hand, was developed in a cross-platform language—*Node.js* with the *Expressjs* framework—to ensure that it would operate in different systems, and both from a local or remote host. Node.js also improves application speed and performance because it is an event-oriented programming language that allows a high number of concurrent connections. Two additional JavaScript libraries were needed for graph generation after data processing: *elementtree*, for GEXF generation, and *Sigmajs*, for graph pre-visualization—pre-visualization was introduced in iteration 4 as a graphical user interface enhancement.

Use Case Scenario

A use case describes system operation in a specific scenario, and how the different actors interact with the system. The use of GraphFES is oriented to instructors and course administrators, and therefore the following use case scenario analyzes the interaction of a teacher with GraphFES:

1. A teacher with a valid user in Moodle LMS accesses GraphFES. GraphFES then prompts for logon credentials: username, password, URL of the Moodle LMS and name of the web service.
2. After authentication, the web interface of GraphFES shows a list of the available courses that the teacher may analyze.
3. The teacher chooses one course and GraphFES starts graph generation.

4. Once the data is processed, the graphs in GEXF format are available for pre-visualization in GraphFES and for download and SNA in Gephi.

The use case presents very few variations. For instance, GraphFES should show a system error in case the authentication data is incorrect. The following section gives a higher degree of detail of the use case scenario proposed in this section by analyzing the sequential operation of GraphFES.

Sequential Operation of GraphFES
A sequence diagram gives a temporal and dynamic representation of the interactions between the different components of a software system, and it highlights the functionality of each component in the process. Therefore, the sequence diagram facilitates understanding of what functions to implement on each component so that the whole process gives a successful output.

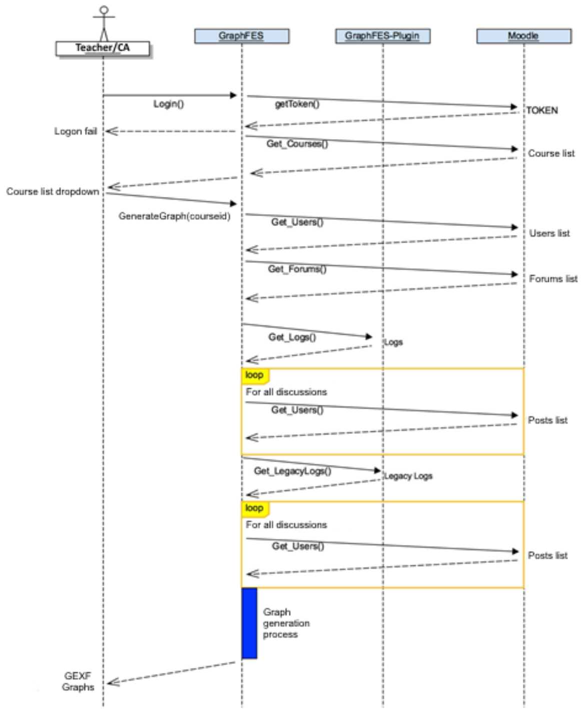

Fig. 3. Sequential operation of GraphFES

For a better understanding of GraphFES operation, the local plug-in and the web application have an independent representation in Fig. 3. Hence, the diagram shows separate calls to the plug-in external functions so that it is easier to differentiate the role of each of GraphFES's components. However, it should be noted that the communication is established between the web application and the web service deployed in Moodle.

The process depicted in Fig. 3, which summarizes GraphFES operation, can be explained in the following sequential stages:

1. The user (most likely, a teacher or course administrator) introduces logon credentials, including URL of Moodle and name of the web service (Note: the web service has to be activated in Moodle; web service activation in Moodle is a straightforward process, but it is omitted in this study due to length limitation. We refer the reader to the following URL for further information: https://docs.moodle.org/29/en/Using_web_services).
2. The web application requests the user's token.
3. If logon credentials are correct, Moodle returns the token that the system will use for the following petitions; else, GraphFES shows a logon error message.
4. Using the token, the web application requests the list of courses available for that user in Moodle.
5. Moodle returns the list of courses in JSON format. The list may be empty if the user cannot access any course or no course exists in the Moodle LMS.
6. The web application shows the list of courses for the user to choose one.
7. The user selects a course and triggers the graph generation process.
8. The web application requests the list of users enrolled in the course and the list of course forums.
9. Moodle returns the lists of users and forums in JSON format.
10. The web application requests course logs for the forum module.
11. Moodle returns the record from the Standard Log in JSON format.
12. For each discussion thread, the web application requests all the posts included in that thread.
13. Moodle returns the list of posts included in each thread in JSON format.
14. Steps 10-13 are repeated for the Legacy Log.
15. After collection of all data, the graph generation process starts.
16. When graph generation has ended, three graph files are stored on disk, in order to make them available to the user. GraphFES generates three different graphs, known as *Views, Replies* and *Messages.* As mentioned in the introductory section, the analysis of the resulting graphs from GraphFES processing is out of the scope of this paper. We refer to [15] for more information about these three types of graphs and how to use them for SNA in Gephi.
17. The web application shows a table with the different graphs generated, and allows the user to download them for SNA in Gephi or make a Force Atlas pre-visualization on the web browser.

System Testing

Mandatory testing follows the development of GraphFES, including three different tests: plug-in tests, web application tests and integration tests. Plug-in tests comprise intensive correct access and log data extraction checks, detection of incorrect input parameters and exception launching as response. Web application tests include error reporting after introduction of erroneous access data, correct GEXF checks, generation of graphs with no logs, no course forums, no posts and no post views, generation of graphs with both Standard and Legacy Log data and correct graph pre-visualization. Finally, integration tests include real course data graph generation and performance tests. For reference, generation of the three graphs in a course with 18 users, 159 messages and 1033 message views takes approximately 3.5 seconds to process in an Intel's Core i5, 2.4 GHz computer, while generation of the three graphs with vast amounts of data (126 users, 9241 messages, 41550 message views) takes approximately 82 seconds.

4 GraphFES: An Illustrated Example

Finally, this section illustrates the operation of GraphFES from the user interface perspective, and shows the results of the process using real data—anonymized—from a computer-supported collaborative learning course (CSCL) with strong focus on teamwork and intensive use of forums. For instance, Fig. 4 depicts the web application interface for authentication, course selection, program execution summary and pre-visualization, while Fig. 5 displays log records as generated by the Moodle reporting module (top) and the visualization of the three graphs generated by GraphFES in Gephi.

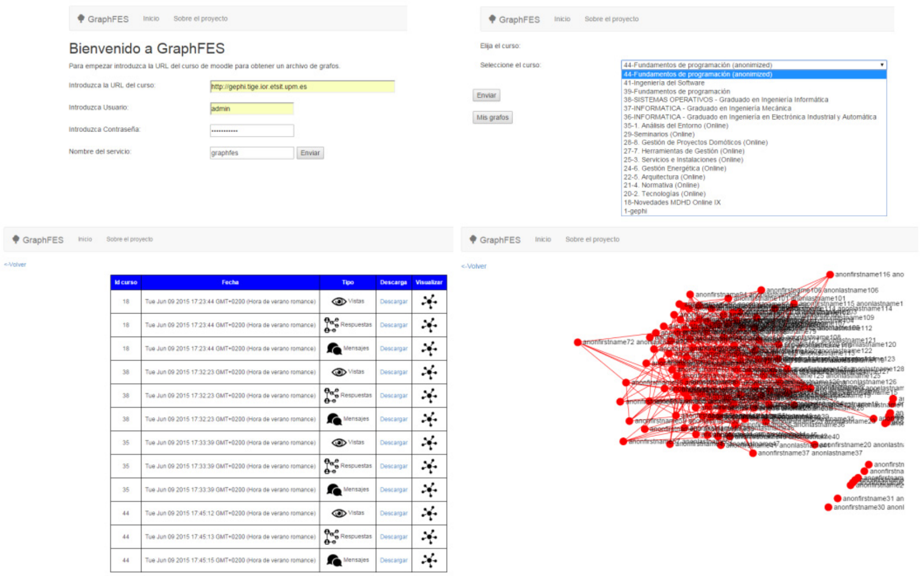

Fig. 4. Left to right, and top-down: GraphFES logon, course selection, graph generation summary and graph pre-visualization screens

Fig. 5. Top: one of the most active teamwork forums and Moodle forum activity report. Bottom: Views, Replies and Messages graphs of that forum in Gephi, with the corresponding SNA metrics returned by Gephi.

5 Conclusion

This paper highlights the need for new solutions and approaches in the emergent field of learning analytics and, more specifically, social learning analytics. In order to do so, the study stresses the current limitations of tools for SNA of educational data, and proposes a new type of tool able to circumvent such limitations: GraphFES, a web service based Moodle data extraction and graph generation tool for SNA in Gephi. Throughout the paper, we have explained the software design process and the technical and architectural principles for the development of such tool. We believe that the implementation of GraphFES has superior potential to that of existing tools and we hope that its future use will accelerate the development of in-depth and advanced social learning analytics.

References

1. Beyer, M.A., Laney, D.: The Importance of 'Big Data': A Definition. Gartner (2012)
2. Chen, H., Chiang, R.H.L., Storey, V.C.: Business intelligence and analytics:from big data to big impact. MIS Quarterly **36**(4), 1165–1188 (2012)
3. Romero, C., Ventura, S.: Educational data mining: A survey from 1995 to 2005. Expert Systems with Applications **33**, 135–146 (2007)
4. Romero, C., Ventura, S.: Educational Data Mining: A Review of the State of the Art. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **40**(6), 601–618 (2010)

5. van Barneveld, A., Arnold, K.E., Campbell, J.P.: Analytics in Higher Education: Establishing a Common Language. EDUCAUSE ELI Paper **1**, 1–11 (2012)
6. Long, P., Siemens, G.: Penetrating the Fog: Analytics in Learning and Education. EDUCAUSE Review **46**(5), 31–40 (2011)
7. Agudo-Peregrina, Á.F., Iglesias-Pradas, S., Conde-González, M.Á., Hernández-García, Á.: Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. Computers in Human Behavior **31**, 542–550 (2014)
8. Gómez-Aguilar, D.A., Hernández-García, Á., García-Peñalvo, F.J., Therón, R.: Tap into visual analysis of customization of grouping of activities in eLearning. Computers in Human Behavior **47**, 60–67 (2015)
9. Hernández-García, Á., González-González, I., Jiménez-Zarco, A.I., Chaparro-Peláez, J.: Applying social learning analytics to message boards in online distance learning: A case study. Computers in Human Behavior **47**, 68–80 (2015)
10. Bandura, A.: Social learning theory. General Learning Press, New York (1971)
11. Berger, P.L., Luckman, T.: The Social Construction of Reality: A Treatise in the Sociology of Knowledge. Penguin Books, Harmondsworth (1967)
12. Lave, J., Wenger, E.: Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, Cambridge (1991)
13. Buckingham-Shum, S., Ferguson, R.: Social Learning Analytics. Journal of Educational Technology & Society **15**(3), 3–26 (2012)
14. Freeman, L.C.: Centrality in social networks: Conceptual clarification. Social Networks **1**(3), 215–239 (1979)
15. Hernández-García, Á.: Usare Gephi per visualizzare la partecipazione nei corsi online: un approccio di Social Learning Analytics. Tecnologie Didattiche **22**(3), 148–156 (2014)
16. Edutechnica: LMS Data – Spring 2015 Updates. (2015). http://edutechnica.com/2015/03/08/lms-data-spring-2015-updates
17. Royce, W.: Managing the development of large software systems. In: Proceedings of IEEE WESCON, pp. 1–9 (1970)
18. Brooks Jr., F.P.: No Silver Bullet Essence and Accidents of Software Engineering. Computer **20**(4), 10–19 (1987)
19. Boehm, B.W.: A spiral model of software development and enhancement. Computer **21**(5), 61–72 (1988)
20. Larman, C., Basili, V.R.: Iterative and incremental developments: A Brief History. Computer **36**(6), 47–56 (2003)
21. Dron, J., Anderson, T.: On the design of collective applications. In: Proceedings of the 2009 International Conference on Computational Science and Engineering, vol. 4, pp. 368–374 (2009)
22. Rayón, Á.: SCALA: Supporting competency assessment through learning analytics. PhD Dissertation. University of Deusto (2015)