Enki: A Pedagogical Services Aggregator for Learning Programming Languages

José Carlos Paiva
CRACS & INESC-Porto LA &
DCC - Faculty of Sciences,
University of Porto
Porto, Portugal
up201200272@fc.up.pt

José Paulo Leal
CRACS & INESC-Porto LA &
DCC - Faculty of Sciences,
University of Porto
Porto, Portugal
zp@dcc.fc.up.pt

Ricardo Queirós
CRACS & INESC-Porto LA &
DI/ESEIG/IPP
Porto, Portugal
ricardoqueiros@eseig.ipp.pt

ABSTRACT

This paper presents Enki, a web-based IDE that integrates several pedagogical tools designed to engage students in learning programming languages. Enki achieves this goal (1) by sequencing educational resources, either expository or evaluative, (2) by using gamification services to entice students to solve activities, (3) by promoting social interaction and (4) by helping students with activities, providing feedback on submitted solutions. The paper describes Enki, its concept and architecture, details its design and implementation, and covers also its validation.

General Terms

Languages, Design, Experimentation

Keywords

E-Learning, Integration, Gamification, Educational Resources, Sequencing, Exercises, Programming Languages

1. INTRODUCTION

Engaging students with the subject is the constant challenge of every teacher. When the subject is computer programming this means conciliating many different things. To start with, it means creating appealing contents to explain programming concepts. It means adapting this content to the needs and preferences of individual students. It means selecting exercises that cover all the syllabus and make sure that they are graded, with the right amount of feedback, timely so that every student stays on track. It means also encouraging the students to work as a group, learning from each other, both by competing and by collaborating. Last but not least, it means selecting the right tools for teaching, and at the same time preparing the students to work with the tools of programming.

The approach presented in this paper sought inspiration in computer programming tools to design an effective web based learning environment. The result combines content presentation, automatic assessment, social and game-like features. Being a web tool it enables students to start learning how to program without having to install fairly complex tools, such as editors and compilers. Nevertheless, it prepares them to, later on, use more sophisticated tools such as Integrated Development Environments (IDE).

Enki is a web-based learning environment with an IDE inspired graphic user interface that integrates several kinds of tools. These tools include a Gamification Service (GS) to provide gamification features to students, an Educational Resources Sequencing Service (ERSS) to offer different learning paths, an Evaluator Engine (EE) to give automatic feedback to students' solutions, an Exercise Creator (EC) to allow teachers to create exercises and a Learning Objects Repository (LOR) to store those exercises. Apart from these tools, Enki also promotes social collaboration and can be integrated in an ecosystem of e-learning systems based on a Learning Management System (LMS).

The remainder of this paper is organized as follows. Section 2 reviews related work on the gamification and social collaboration in e-learning, sequencing of educational resources, automatic evaluation of exercises and interoperability. Section 3 introduces Enki, its architecture and graphical user interface. Section 4 describes its validation in a programming course. Finally, Section 5 summarizes the contributions of this research.

2. RELATED WORK

To the best of the authors' knowledge, there is no tool in the literature that provides all the features mentioned in the previous section, integrated in Enki. So, this section surveys systems with some of these features.

2.1 Gamification On E-Learning

Gamification aims to engage users by applying game principles (points, progression, competition), in non-game contexts. Gamification is currently being applied to e-learning environments with relative success, as it helps students to remain focused and thus to fulfill course goals.

In this context, the most widely used approach is to empower LMS with game mechanics such as badges, achievements, leaderboards in order to boost engagement and to improve knowledge retention. Some of the notable examples are Academy LMS, Axonify and Matrix. Despite the success of this approach, the concept of loser may adversely affect the motivation of students [27].

ITiCSE '16, July 09-13, 2016, Arequipa, Peru

DOI: http://dx.doi.org/10.1145/2899415.2899441

Peer 2 Peer University (P2PU) [1] is another environment defined as a social computing platform that promotes peer-created and peer-led online learning environments. In P2PU, learners can join, complete and leave challenges at any time. They can also earn badges, associated with learning tasks and courses, which are based on Mozilla Open Badges framework[1]. P2PU also promotes social collaboration by allowing any stakeholder to create a course.

2.2 Sequencing of Educational Resources

Most of the learning environments offer the same learning paths and resources to students, regardless of their prior knowledge, goals, progress or preferences which has a negative impact on their motivation [26].

Multibook [9] is a web-based adaptive hypermedia learning system for multimedia and communication technology, developed by the Technical University of Darmstadt and the University of Hagen. Multibook aims to offer different lessons to different users, by storing a huge number of compiled lessons or by dynamically generating lessons for each user. It uses four dimensions for each user. Firstly, Multibook fills the user's profile with his demands and preferences. The Multibook system also keeps track of the information that had been found and learned by users as well as the additional materials requested while users work with the system. The knowledge base of Multibook consists of two distinct concept spaces – Concept Space and Media Brick Space. Concept Space contains a networked model of learning topics and uses knowledge management approaches. Media Brick Space is used to store atomic information units (media bricks) of different multimedia formats which are interconnected via rhetoric relations. Media bricks use IEEE's LOM scheme and are described and treated as learning objects.

2.3 Social Collaboration On E-Learning

One of the main reasons that leads students to online course dropout is the lack of social interaction. Most students play a passive role on learning environments, often reduced to consult content provided by teachers which leads to an unstimulating environment [24]. The natural characteristics of online social networks, such as content sharing and comments, promote an active and stimulating learning environment[18].

SCALE (Supporting Collaboration and Adaptation in a Learning Environment) [11] is a web-based educational environment with learning and assessment content. It enables students to (1) work on individual or group activities proposed with respect to their knowledge, (2) participate in self-assessment, peer-assessment or collaborative-assessment activities, (3) work with embedded educational environments, (4) use synchronous and asynchronous communication tools and (5) have access to feedback components. Three studies [11] shown that this tool facilitates and supports learning and assessment.

2.4 Automatic Evaluation Of Exercises

The evaluation of exercises takes a large amount of time to teachers, and thus, many universities have invested in the development of automated assessment systems [3, 21]. Several of these systems are only prepared for the assessment of programming assignments [13, 20, 17, 5] and differ on the extra features that they provide, such as multi-programming language support, evaluation type (static or dynamic), feedback, interoperability, learning context, security and plagiarism. However, there is also some work on automatic evaluation of other types of exercises such as UML, Mathematics and Physics [21, 4, 12].

Mooshak [17] is a web-based system to handle programming contests. It acts as a full contest manager and as an automatic judge for programming contests. Mooshak supports submissions of exercises written in several programming languages. The standard way of to evaluate a program is to compile it and then execute it with test cases input files, comparing the obtained output with the expected result (black-box approach). It also deals with non-determinism using special correctors, which are invoked after each test case execution. The feedback provided by this system consists of error status (e.g. wrong answer, compilation error, execution errors).

2.5 Interoperability

Most of the learning institutions have already adopted a Learning Management System (LMS) to organize and share their course resources, to deliver assignments and/or to report the performance of the students [8]. So, interoperability among e-learning content and components is increasingly becoming the key to the success of any e-learning environment.

Many approaches to couple LMS with other applications have been proposed, since defining LMS from scratch based on service-oriented architectures [2, 6], including web services layers within the LMS infrastructure [25, 7] or providing support for interoperability specifications [16].

The latter approach is primarily based on IMS specifications, namely the LTI (Learning Tools Interoperability) specification that facilitates the integration between LMS and external applications. The TSUGI framework [2] is a recent proposal to simplify the implementation of LTI tools.

3. ENKI

This section describes Enki, a web-based IDE for learning programming languages using gamification features. Enki blends assessment and learning, presenting content, from hypertext to video, as well as exercises, in an adaptive and engaging way.

The IDE makes use of gamification to engage students in the learning process, interacting with gamification services to support the creation of leaderboards, reward students for their achievements, among others. It also integrates a service for sequencing educational resources to provide different learning paces according to students' capabilities. The exercises and assessment are, typically, programming exercises. The system that hosts Enki includes also interfaces for teachers to author and manage both exercises and content, as well as to browse assessment results and student profiles.

The next subsections present the architecture of Enki and its main components, and describe its implementation.

3.1 Architecture

Enki is a part of the Mooshak 2.0, the new version of Mooshak [17], a web environment for automated assessment in computer science, both in competitive and pedagogical learning. The new version is a complete re-implementation

[1]http://openbadges.org/

[2]http://csev.github.io/tsugi/

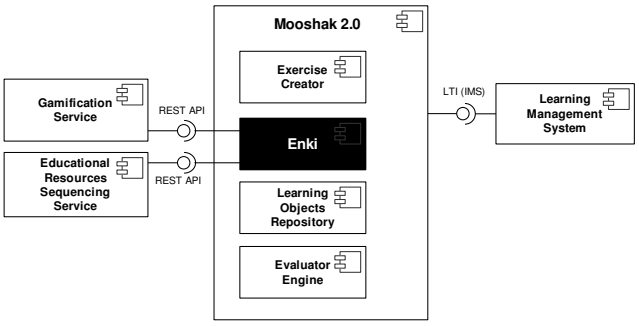

Figure 1: Components diagram of the network of Enki where Mooshak 2.0 acts as a tool provider for an LMS

of the code base with a wider variety of user interfaces for different use cases. It has interoperability features that enable it to interact with other e-learning tools such as LMSs.

Enki takes advantage of Mooshak 2.0 to have a pivotal role in a network of e-learning systems, coordinating the communication with all external components as depicted by the UML components diagram in Figure 1.

An important task for building the network of Enki is the choice of the systems that would play each role. The next sub-subsections describe the types of systems that compose the network presented in Figure 1, and introduce the selected system(s) for each of the components.

3.1.1 Learning Management System (LMS)

An LMS is a software application for the administration, documentation, tracking and reporting; used in training programs, classrooms and online events [10]. Typically it is used by two types of users' groups: learners and teachers. The learners can use the LMS to plan their learning experience and to collaborate with their colleagues; the teachers can deliver educational content and track, analyze and report the learner evolution within an organization.

Nowadays, an LMS plays a central role in any eLearning architecture. Still, the LMS cannot afford to be isolated from other systems in an educational institution. Thus, the potential for interoperability is an important, although frequently overlooked, aspect of an LMS system [15].

The purpose of Enki is to integrate an e-learning ecosystem based on an LMS. For this, Enki benefits from the interoperability mechanisms inherited from Mooshak 2.0 to provide authentication directly from the LMS and to submit exercises grades to the LMS, using the Learning Tools Interoperability (LTI) specification.

Although the majority of the LMSs support LTI communication [23], only Moodle and Sakai are able to fully integrate with Enki. Blackboard LMS is also capable of running Enki but it cannot receive grading results.

3.1.2 Gamification Service (GS)

A Game-Backend-as-a-Service (GBaaS), which is abbreviated here as Gamification Service (GS), is a subset of a Backend-as-a-Service (BaaS) - a cloud computing service model acting as a middleware component that allows developers to connect their applications to cloud services via application programming interfaces (API) and software development kits (SDK) - that includes cross-platform solutions for the typical game concepts. These GBaaS that can leverage on their authentication services and massive user base already provide gamification features. However, gamification services that rely on external authentication are not adequate for a network of e-learning systems which already operates on a single sign-on ecosystem.

Since there was no service fulfilling the requirements of the network of Enki, a new gamification service was developed. This service – called Odin – [14] is inspired in the Google Play Game Service (GPGS) but with a different approach regarding authentication. Institutions, rather than end-users, are the ones that require authentication. The communication with Odin is made through its REST API, similar to the GPGS API[3].

3.1.3 Educational Resources Sequencing Service (ERSS)

The ERSS selected was Seqins [22]. Seqins is a sequencing tool of digital educational resources that includes a flexible sequencing model that fosters students to learn at different rhythms. Enki feds Seqins, through its REST API, with precedence among content units, assessment results and students' progress and Seqins provides an XML representation of the resources to present to the current student.

3.1.4 Evaluator Engine (EE)

The purpose of an EE is to mark and grade exercises. In this network, an EE should perform four tasks: (1) receive a reference to the exercise, an attempt to solve it (a program) and a reference to the student submitting the attempt, (2) load the exercise from the LOR (possibly itself) using the given reference, (3) compile the solution and run the tests, related to the exercise, against the attempt of the student and (4) produce an evaluation report with the classification, feedback and, possibly, corrections.

The EE system is provided by Mooshak 2.0. As in its previous version, the main feature is the automatic evaluation of exercises, adding support for different exercise types and better feedback. For the Enki purposes, this evaluator suffered some minor changes to be less strict.

3.1.5 Exercise Creator (EC)

An EC must allow teachers to create a complete exercise package, containing a statement, a solution, tests, skeletons, and a manifest file describing the contents of the package. This package must follow the same package specification as the LOR for programming exercises.

This kind of tool is offered by Mooshak 2.0, which also stores its exercises on its own repository.

3.1.6 Learning Objects Repository (LOR)

A Learning Objects Repository (LOR) is a system that stores educational resources and enables educators to share, manage and use them. These resources (or Learning Objects) are small, self-contained and reusable educational units which, typically, have additional metadata to catalog and search them. The system that plays the role of a LOR in the network of Enki is also Mooshak 2.0.

3.2 Graphical User Interface

Enki was developed using Google Web Toolkit (GWT), an open source software development framework that allows a

[3]https://developers.google.com/games/services/web/api

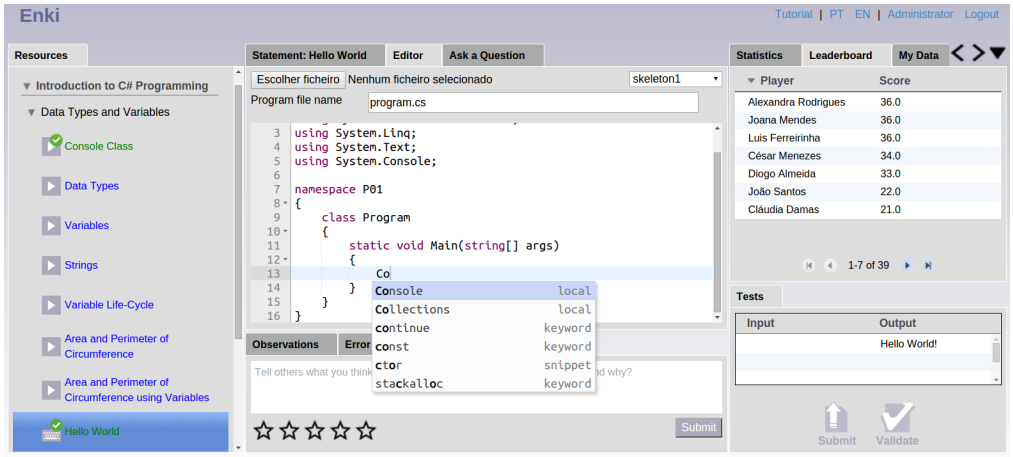

Figure 2: Interface of Enki for students

fast development of AJAX applications in Java. The GWT code is organized in two main packages, the server, which includes the service implementations triggered by the user interface, and the client, that includes the user interfaces.

A distinctive feature of Enki is its student interface, presented in Figure 2, which emulates an integrated development environment (IDE). It was designed for responsiveness, to be used in different resolutions and devices as well as to resize the browser window, keeping the panels' proportions or adjusting them to a better look and feel (in case of smaller resolutions). Students can rearrange panels and tabs to their needs with the drag-and-drop and resize features provided by the Enki's interface. These features rely on two free and open-source GWT libraries: GWT Bootstrap 3[4], for responsiveness and GwtQuery[5], for drag-and-drop.

The interface of Enki is composed of widgets in different panels which intercommunicate with each others – *gadgets*. These gadgets are independent components provided by Mooshak 2.0, although some of them were, in fact, developed having Enki in mind. The most important gadgets are the following:

Resources Tree – to browse and select the available course contents;

Leaderboard – a table with the players' name and score, sorted by decreasing score;

Problem Stats – a chart built using the Google Chart Tools API for GWT, that summarizes the submission statistics of a problem;

Achievements – contains the unlocked achievements;

Profile – summarizes information about the logged-in student;

Code Editor – based on the Ace Editor, allows students to code in most programming languages with syntax highlight and code completion, starting from a skeleton provided by the exercise author;

Test Cases – allows students to verify their programs with both public or their own tests;

PDF Viewer – shows both problem statements and static educational resources;

Video Viewer – allows the student to see and share video resources;

Ask Question – lists questions already answered and allows students to submit their own questions;

Resource Rating – enables students to give feedback on educational resources.

The resources tree mentioned above is particular important as it drives student interaction by presenting both the course structure and content. Every level may hold educational resources (the leaves of the tree), which can be of different types: text (HTML or PDF), multimedia and activities (exercises). Each resource presented in the cell tree has an icon reflecting its type and a color depending on its state relative to the student: available, solved/seen, unavailable or recommended.

The global system of Enki also contains two additional interfaces for teachers, to view the results of the submissions of the students and to author exercises.

4. VALIDATION

This section presents an acceptability evaluation of Enki. To carry out this evaluation an experiment was conducted with undergraduate students at *Escola Superior de Estudos Industriais e de Gestão (ESEIG)* - a school of the Polytechnic Institute of Porto - from the 4th to the 15th of January of 2016.

The experiment took the form of an Open Online Course entitled "Introduction to C# Programming", free of charge and without participation limits. It had an enrollment of 70 students, of which 28 were females. The course contains resources of two types: expository and evaluative. The expository resources are typically videos with working examples of exercises solving and a few theoretical resources. The evaluative resources are programming exercises that allow students to consolidate their knowledge. The videos were created with Camtasia[6], a software that records screen activity and voice, and deployed on YouTube. The design requirements of these videos where the following: (1) cover all the curricula (coverage); (2) have several difficulty levels (diversity); (3) have at most 5 minutes (fragmentation); and

[4]https://gwtbootstrap3.github.io/gwtbootstrap3-demo/
[5]http://code.google.com/p/gwtquery/
[6]https://www.techsmith.com/camtasia.html

(4) be composed by pictures, sound and subtitles (completeness).

The exercises comply with the Mooshak programming problem package specification. This package is an archive containing a problem description (typically an HTML file), a file with the solution, an XML file with the structure of the package, a folder with tests and their output, and optionally a folder with images and a folder with skeletons of the solution.

In the last day of this experiment, students were invited to take a final test with 5 questions covering all the syllabus. After the experiment they were also invited to fill-in an online questionnaire based on the Nielsen's model [19], using Google Forms. It includes questions on the usefulness of Enki, i.e. on its utility and usability. Utility is the capacity of the system to achieve a desired goal. Usability is defined by Nielsen as a qualitative attribute that estimates how easy is to use an user interface. The survey was completed by 25 students, of which 9 were females.

Figure 3 shows the results grouped by Nielsen's heuristics. The collected data is shown in a bar chart, with heuristics sorted in descending order of user satisfaction.

On the positive side the results showed that the consistence, recognition and aesthetic were the heuristics with higher satisfaction. The respondents selected the minimalist design as one of the strongest points of Enki. On the negative side the results highlighted deficiencies in three areas: speed and reliability, error prevention and users help and documentation. Students complained about the delay when they validate or submit their programs. This was due to a machine overload that was already solved. Other students stated that the messages of the system were scarce and difficult to interpret. The improvement of feedback is one of the major requirements for future versions of Enki. Finally, students complained of the lack of documentation while using Enki.

The questionnaire finalizes with an overall classification of Enki in a 5 values Likert-type scale (very good, good, adequate, bad, very bad). The majority of students (56%) classified Enki as an adequate tool and many of them (40%) stated Enki as a good or a very good tool. Very few students (4%) found it either bad or very bad.

5. CONCLUSIONS

This paper presents Enki, a web-based IDE for learning programming languages in an adaptive and engaging way. This IDE resorts to gamification services to support the creation of leaderboards, reward students for their achievements, among others, in order to engage the students in the learning process. It also mimics game levels by integrating with a service for sequencing educational resources in different rhythms to heterogeneous students.

The Open Online Course to introduce students to the C# programming language is a proof of the acceptability of Enki as a pedagogical tool to learn programming languages. Also, it has proven its ability to integrate in an e-learning environment based on a LMS.

Enki and its network will be subject to improvements. The students and teachers involved in the Open Online Course reported some minor issues, related to the user interface, which are already being solved for the next version.

Regarding new features, the evaluation feedback will be the major focus point in the next version. Currently, the feedback provided is only based on tests, which is not adequate to introduce students to programming. The next version will benefit of a new improvement to Mooshak 2.0, concerning the static analysis of code, to provide richer feedback to students. Also, the ERSS system will include a long-term recommendation component. This component will recommend resources to students with the final goal set to pass the final evaluation with the best score possible within the available time left to the end of the course and the personal characteristics of the student.

6. ACKNOWLEDGMENTS

This work is partially funded by the ERDF – European Regional Development Fund, through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme, with National Funds of FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology), within project POCI-01-0145-FEDER-006961; by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF), within Project "NORTE-01-0145-FEDER-000020"; and by the Polytechnic Institute of Porto, by its Integration Research and Development Grants (BInt-ICD/IPP-BST/KMILT/01/2015).

7. REFERENCES

[1] J. Ahn, B. S. Butler, A. Alam, and S. A. Webster. Learner participation and engagement in open online courses: Insights from the peer 2 peer university. *MERLOT Journal of Online Learning and Teaching*, 9(2):160–171, 2013.

[2] M. Al-Smadi and C. Gütl. Soa-based architecture for a generic and flexible e-assessment system. In *Education Engineering (EDUCON), 2010 IEEE*, pages 493–500. IEEE, 2010.

[3] K. M. Ala-Mutka. A survey of automated assessment approaches for programming assignments. *Computer science education*, 15(2):83–102, 2005.

[4] N. H. Ali, Z. Shukur, and S. Idris. Assessment system for uml class diagram using notations extraction. *International Journal on Computer Science Network Security*, 7:181–187, 2007.

[5] M. Blumenstein, S. Green, A. Nguyen, and V. Muthukkumarasamy. An experimental analysis of game: a generic automated marking environment. In *ACM SIGCSE Bulletin*, volume 36, pages 67–71. ACM, 2004.

[6] O. Casquero, J. Portillo, R. Ovelar, M. Benito, and J. Romo. iple network: an integrated elearning 2.0 architecture from a university's perspective. *Interactive Learning Environments*, 18(3):293–308, 2010.

[7] M. Á. Conde, F. J. García, M. J. Casany, and M. Alier. Applying web services to define open learning environments. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, pages 79–83. IEEE, 2010.

[8] D. Dagger, A. O'Connor, S. Lawless, E. Walsh, and V. P. Wade. Service-oriented e-learning platforms: From monolithic systems to flexible services. *Internet Computing, IEEE*, 11(3):28–35, 2007.

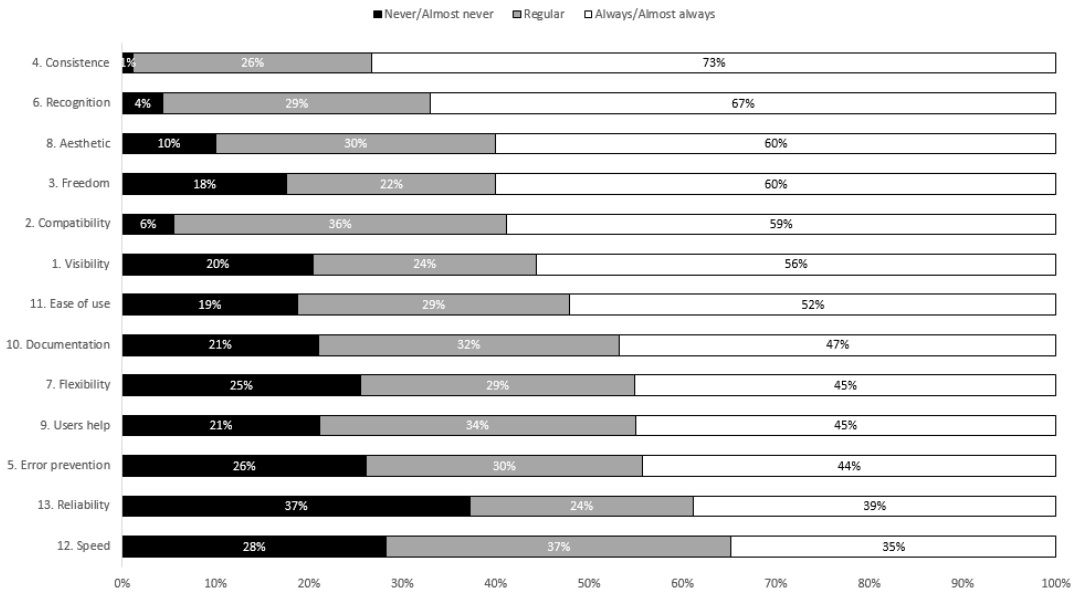

Figure 3: Enki acceptability evaluation

[9] A. El Saddik, S. Fischer, and R. Steinmetz. Reusable multimedia content in web-based learning systems. *IEEE Multimedia*, 8(3):30–38, 2001.

[10] R. K. Ellis. Field guide to learning management systems. 2009.

[11] A. Gogoulou, E. Gouli, M. Grigoriadou, M. Samarakou, and D. Chinou. A web-based educational setting supporting individualized learning, collaborative learning and assessment. *Educational Technology & Society*, 10(4):242–256, 2007.

[12] M. Harjula et al. Mathematics exercise system with automatic assessment. 2008.

[13] D. W. Juedes. Experiences in web-based grading. In *Frontiers in Education, 2003. FIE 2003 33rd Annual*, volume 3, pages S3F–27. IEEE, 2003.

[14] J. P. Leal, J. C. Paiva, and R. Queirós. Odin: A service for gamification of learning activities. In *to appear*, page to appear. Springer, 2015.

[15] J. P. Leal and R. Queirós. *A comparative study on LMS interoperability*. IGI-Global, 2011.

[16] J. P. Leal and R. Queirós. Using the learning tools interoperability framework for lms integration in service oriented architectures. 2011.

[17] J. P. Leal and F. Silva. Mooshak: a web-based multi-site programming contest system. *Software: Practice and Experience*, 33(6):567–581, 2003.

[18] M. Li and Z. Liu. The role of online social networks in students' e-learning experiences. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1–4. IEEE, 2009.

[19] J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213. ACM, 1993.

[20] Y. Pisan, D. Richards, A. Sloane, H. Koncek, and S. Mitchell. Submit! a web-based system for automatic program critiquing. In *Proceedings of the fifth Australasian conference on Computing education-Volume 20*, pages 59–68. Australian Computer Society, Inc., 2003.

[21] F. Prados, I. Boada, J. Soler, and J. Poch. Automatic generation and correction of technical exercises. In *International Conference on Engineering and Computer Education: ICECE*, volume 5, 2005.

[22] R. Queirós, J. P. Leal, and J. Campos. Sequencing educational resources with seqins. 2014.

[23] R. Queirós, J. P. Leal, and J. C. Paiva. Integrating rich learning applications in lms. In *State-of-the-Art and Future Directions of Smart Learning*, pages 381–386. Springer, 2016.

[24] J. J. Rodrigues, F. M. Sabino, and L. Zhou. Enhancing e-learning experience with online social networks. *IET communications*, 5(8):1147–1154, 2011.

[25] C. Severance, J. Hardin, and A. Whyte. The coming functionality mash-up in personal learning environments. *Interactive Learning Environments*, 16(1):47–62, 2008.

[26] C. Stewart, A. I. Cristea, T. Brailsford, and H. Ashman. 'authoring once, delivering many': creating reusable adaptive courseware. 2005.

[27] M. Vansteenkiste and E. L. Deci. Competitively contingent rewards and intrinsic motivation: Can losers remain motivated? *Motivation and Emotion*, 27:273–299, 2003. 10.1023/A:1026259005264.