

Design and implementation of an IDE for learning programming languages using a gamification service

José Carlos Paiva, José Paulo Leal and Ricardo Queirós

December 18, 2017

Abstract

This chapter presents the architecture and design of Enki, an Integrated Development Environment (IDE) for learning programming languages on Massive Open Online Courses (MOOCs). This environment can be used as a tool by a Learning Management System (LMS) and a typical LMS such as Moodle can launch it using the Learning Tool Interoperability (LTI) API. Student authentication tokens are passed via LTI, thus integrating Enki in the single sign-on domain of the academic institution. The proposed tool has a web user interface similar to those of reference IDEs, where the learner has access to different integrated tools, from viewing tutorial videos, to solving programming exercises that are automatically evaluated. Enki uses several gamification strategies to engage learners, including generic gamifications provided by Odin and the sequencing of educational resources. The course content (videos, PDFs, programming exercises) is progressively disclosed to the learner as he successfully completes exercises. This is similar to what happens in a game, where new levels are unlocked as the previous are completed, thus contributing to the sense of achievement.

1 Introduction

Environments for learning programming languages have very precise requirements. Students need editors and compilers of the languages they are learning, in specific versions. If students attend practical classes in a lab then it is fairly easy to ensure they all have access to all the tools they need. However, if the students are enrolled in a MOOC, they must install themselves these tools on their own computers or devices. Installing software such as compilers is too complex for many students of introductory programming courses, and some may even use devices where these installation is not feasible, such as in tablets.

Enki is a programming learning environment designed for students enrolled in MOOCs. It integrates several features surveyed in Section 2, which are commonly found in e-Learning environments. It provides a user interface deployed on the web with automated assessment running on a remote server and was designed to interact with several e-Learning systems. The most common of these systems is the Learning Management Systems (LMS) that is interlinked

with Enki using the LTI specification. Enki communicates also with more specialized e-learning systems, namely with a content sequencing engine named Seqins. However, the most distinctive interoperability feature of Enki is the use of a gamification service.

The use of game concepts and mechanics in eLearning is an effective way to engage learners. Techniques and tools that have proven their effectiveness in computer games are routinely exploited in other areas of software development. The use of this approach led to the concept of gamification as a service, provided by major players such as Google and Microsoft. These services leverage on their large user base to provide support for points, leaderboards and badges, requiring a specific authentication from the client application.

However, e-learning systems are typically deployed in environments with a single sign-on managed by an academic institution that should not require students to have an account with a third party. Odin is a gamification service designed to circumvent this obstacle. It retains the salient features of the gamification services of reference without requiring registration of the end users. Its API is inspired in the Google Play Game Service (GPGS) with minor adjustments regarding user identification.

Section 3 presents Enki. It starts with a description of the IDE inspired user interface. Then, it describes the architecture of which Enki is the visible face. It encompasses Mooshak, the automated evaluation system hosting Enki, as well as gamification services, sequencing services and learning management systems. It also details an use case that shows how these components are interconnected. Section 4 concludes the chapter with an highlight of the main features of the described system.

2 Background

Teaching computer programming is considered to be difficult and often ineffective (Robins, Rountree, & Rountree, 2003). Our tool, Enki, is a web-based IDE designed to be integrated on MOOCs that allows users to learn computer languages efficiently. To the best of the authors' knowledge, there is no tool in the literature that provides all the Enki features such as gamification features, sequencing of educational resources, social collaboration, automatic evaluation and standard LMS integration. So, this section surveys systems with some of these features.

2.1 Gamification

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 (Vansteenkiste & Deci, 2003).

Codeacademy (Ryzac, 2012) is an online interactive platform that offers free coding classes in several programming languages. The programming courses are organized into lessons, consisting of sequences of exercises. The exercises are composed by a statement, a set of instructions and a skeleton exercise to be completed. Each exercise completed gives students one point and the completion of a lesson is registered as an achievement. Students are also rewarded with badges for attaining specific number of points or completing certain lessons or courses, which they can share on social networks such as Facebook.

PeerSpace (C. Li & Chasteen, 2013) is an online social network based learning environment to engage students in collaborative learning. The environment has a strong peer support, that leave students more comfortable willing to share knowledge and experiences, exchange ideas, and seek help. PeerSpace has a set of learning activities such as peer code review, group wiki-based exercises, and self-paced quizzes that encourage students to help and learn from each other. Study results show that students became more active in social activities.

Peer 2 Peer University (P2PU) (Ahn, Butler, Alam, & Webster, 2013) 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 framework1. P2PU also promotes social collaboration by allowing any stakeholder to create a course.

In short, nowadays gamification is applied in computer programming learning online environments by including game mechanics such as badges, achievements and leaderboards in order to maintain students' motivation.

2.2 Sequencing of Educational Resources

In the e-learning realm, the learner satisfaction when using interactive learning environments is influenced by the type of learning experience that the environment provides. Existing adaptive educational hypermedia systems have been using learning resources sequencing approaches in order to address the issues previously focused. These systems typically implement a rule-based sequencing approach based on a two steps procedure: they generate a sequence of concepts that matches the learning goal and then select learning resources for each concept of that sequence.

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 (Stewart, Cristea, Brailsford, & Ashman, 2005).

Karampiperis (Karampiperis & Sampson, 2005) presented an alternative approach since it first generates all possible learning paths that match a specific learning goal. After that, it adaptively selects the desired one, based on the use of a decision model that infers the suitability of learning resources for a specific targeted learner.

Brusilovsky (Brusilovsky & Millán, 2007) introduced the user model facet as an important piece to any adaptive system since the prior knowledge of information related with the user that uses the system can help the system in its guidance.

Fischer (Fischer, 2001) explored the use of metadata standards such as IEEE LOM and Dublin Core with regard to course and exercises sequencing. The following subsections focus on other important types of e-learning specifications such as sequencing and integration specifications that should be taken in consideration when developing learning resources sequencing approaches.

Multibook (El Saddik, Fischer, & Steinmetz, 2001) 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 (Rodrigues, Sabino, & Zhou, 2011). The natural characteristics of online social networks, such as content sharing and comments, promote an active and stimulating learning environment (Li & Liu, 2009).

The Web assumes nowadays a mandatory place to learn how to program. In fact, there are several interactive programming courses such as Codecademy, Coursera, EDX, OdinProject, Udacity, Udemy, CodeSchool, and many others, where we can get free or paid lessons in the form of HTML documents, PDF or videos. These lessons formalize a predefined and validated learning path so that the student does not feel lost in the panoply of educational resources gravitating on the web. In order to increase retention rates in these courses, these programming environments include features regarding social collaboration allowing students of the same educational level to interact each others or with tutors who have the responsibility to guide them. These features are essential to motivate learners and enrich interactive programming courses as scaffolding tools for computer programming learning.

SCALE (Supporting Collaboration and Adaptation in a Learning Environment) (Gogoulou, Gouli, Grigoriadou, Samarakou, & Chinou, 2007) 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 (Gogoulou et al., 2007) 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 (Ala-Mutka, 2005; Prados, Boada, Soler, & Poch, 2005). Several of these systems are only prepared for the assessment of programming assignments (Juedes, 2003; Pisan, Richards, Sloane, Koncek, & Mitchell, 2003; Leal & Silva, 2003; Blumenstein, Green, Nguyen, & Muthukkumarasamy, 2004) 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 (Prados et al., 2005; Ali, Shukur, & Idris, 2007; Harjula et al., 2008).

Mooshak (Leal & Silva, 2003) 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 (Dagger, O'Connor, Lawless, Walsh, & Wade, 2007). 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 (Al-Smadi & Gütl, 2010; Casquero, Portillo, Ovelar, Benito, & Romo, 2010), including web services layers within the LMS infrastructure (Severance, Hardin, & Whyte, 2008; Conde, García, Casany, & Alier, 2010) or providing support for interoperability specifications (Leal & Queirós, 2011b).

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 ¹ is a recent proposal to simplify the implementation of LTI tools.

3 Implementation

This section describes the implementation of Enki, a web-based IDE for learning programming languages which integrates several services to ease programming learning. Enki blends assessment and learning, presenting content, from hyper-text to video, as well as exercises, in an adaptive and engaging way.

¹<http://csev.github.io/tsugi/>

This 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 contains 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 graphic user interface and architecture of Enki and its main components, and describe an use case.

3.1 Graphic User Interface

Enki can also be described as an entry level IDE, a scaffolding tool to support the progress of students towards more complex environments and tools. It borrows from IDEs the characteristic graphic user interface, organized in regions which contain several overlapping windows organized using tabs. The layout scheme for these regions locates them roughly in the main directions of the compass: north, south, east and west. Regions are resizable and the windows they contain can be moved among regions, according to the needs and preferences of students.

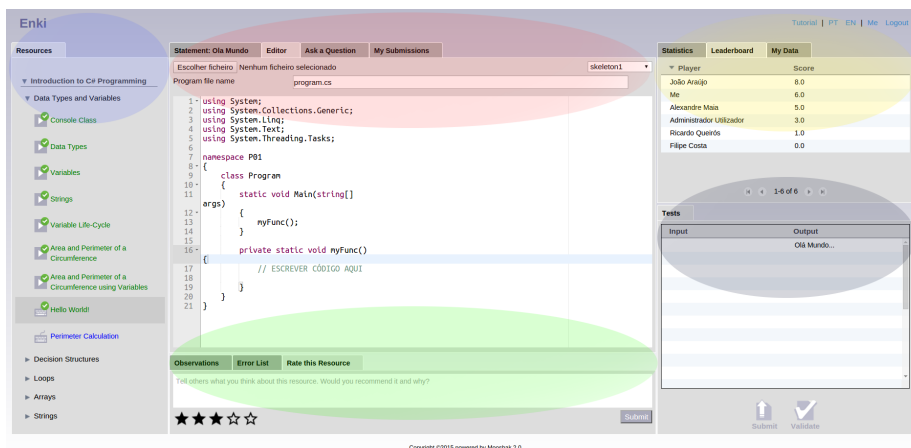


Figure 1: Interface of Enki for students with different regions highlighted in different colors

Enki's GUI is composed of 5 regions located in the west, center, south, northeast and southeast regions of the screen. This regions are easily mapped to those existent in an IDE.

The west region, with a purple highlight in Figure 1, contains, by default, a *resource browser*. In known IDEs, such as Eclipse or NetBeans, this region contains browsers that provide different views of the workspace, either organized in projects, packages and classes, or organized in directories and files. In Enki, this region is used for browsing pedagogical resources using a tree widget that mimics those used by IDE browsers.

The resource browser is a particularly important part of Enki's GUI as it drives student interaction by presenting both the course structure and content. The view of course unfolds as the student progresses throughout the course. This view is mediated by Seqins (described in subsection 3.2). Tree leaves may hold educational resources of different types: text (HTML or PDF), multimedia and activities (exercises). They are presented in the tree with an icon reflecting its type and a color depending on its state relative to the student: available, solved/seen, unavailable or recommended.

The center region, highlighted in magenta in Figure 1, is the main region of Enki's GUI. The expository and evaluative resources selected in the browser will be open, by default, in this region, in one or more tabs. Expository resources are presented in specialized viewers (eg: a sequence of steps can be illustrated by an embed video hosted in YouTube, and reference material can be presented in PDF or HTML formatted pages). On the other hand, evaluative resources open several windows: to show the exercise statement, to edit a solution and to evaluate it. Currently Enki supports two editors for different kinds of computer science languages: a code editor for programming languages such as Java or C#, and a diagram editor for diagrammatic languages such as EER or UML.

The northeast region, highlighted in yellow in Figure 1, aggregates gamification related windows, such as the leaderboard and the achievements, and windows that global or personal progress, such as problem statistics and profile data. These windows are not always present, it depends on the selected resource type. The majority of the information presented in this region is provided by Odin and Seqins (both are described in subsection 3.2).

The southeast region, by default, is just available in exercises and activities. In this case, it contains the window to write input test cases to execute with the exercise, where corresponding outputs are later on presented, and buttons to validate or submit the solution.

The windows presented in the south region (highlighted in green in Figure 1) depend on the selected resource type. In exercises and activities, three windows are shown in different tabs: the observations window which contains automated feedback on an exercise, the error list window that summarizes the compilation/syntax errors and warnings, and a window that allows the student to rate and/or comment a resource. This last window is also shown on educational resources, as well as one with links to related resources.

3.2 Architecture

Enki is a part of the Mooshak 2.0, the new version of Mooshak (Leal & Silva, 2003), a web environment for automated assessment in computer science, both in competitive and pedagogical learning. The new version is a complete re-implementation 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 2.

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

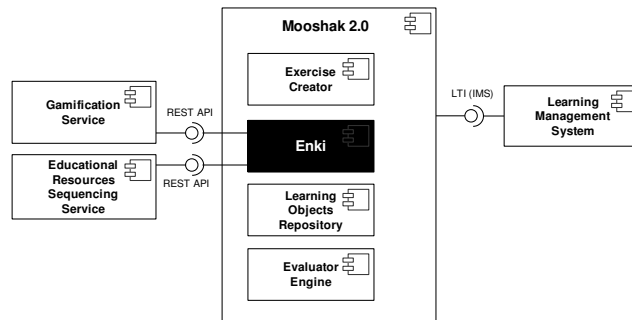


Figure 2: Components diagram of the network of Enki where Mooshak 2.0 acts as a tool provider for an LMS

of systems that compose the network presented in Figure 2, and introduce the selected system(s) for each of the components.

3.2.1 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 (Ellis, 2009). 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 (Leal & Queirós, 2011a).

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 (Queirós, Leal, & Paiva, 2016), 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.

3.2.2 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 – (Paiva, Leal, & Queirós, 2015) 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 API².

3.2.3 Educational Resources Sequencing Service (ERSS)

The ERSS selected was Seqins (Queirós, Leal, & Campos, 2014). Seqins is a sequencing tool of digital educational resources that includes a flexible sequencing model that fosters students to learn at different rhythms. Enki feeds 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.

3.2.4 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.

3.2.5 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.

3.2.6 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.

²<https://developers.google.com/games/services/web/api>

3.3 Use Case

A typical use of Enki by a student starts with its launch from the LMS and involves a submission of a solution to a problem. The main interactions that these actions trigger are summarized in Figure 3.

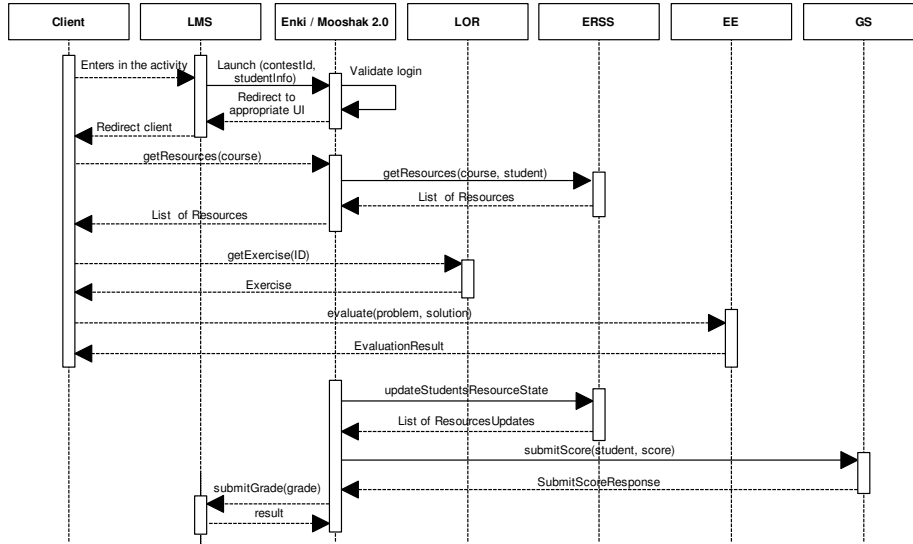


Figure 3: Sequence diagram of an use case of Enki

Firstly, after the student enters in the activity presented in the LMS, the LMS will send an LTI request, with the available information of the student and the id of the course, to the LTI authentication servlet of Mooshak 2.0. This servlet parses the LTI request using a Java package which implements both sides of the LTI communication – LTI Wrapper (Queirós et al., 2016) –, authenticates the student and invokes the appropriate interface, if the authentication succeeds.

While the interface is built, a Remote Procedure Call (RPC) is made to the server side of Enki requesting the resources to present to the student. Enki answers this request by performing a similar request to the ERSS and giving back the response, an XML formatted tree of resources with the state of each resource to the student annotated.

Once the student selects an evaluative resource, an RPC call is made to the LOR requesting the problem. The LOR responds with the statement, skeletons and public test cases of the requested problem. After solving the exercise, the student submits his solution to the EE – Mooshak 2.0 – through an RPC call and gets the feedback. After the evaluation, Enki performs three actions: (1) the result and an identifier of the student is sent through HTTP POST to the ERSS, which answers with a list of updated resources, (2) a score submission request is sent to the GS and (3) the grade is submitted to the LMS.

The communication with REST APIs of the ERSS and the GS is made through *Jersey Core Client*, which is a part of Jersey³ - an open-source framework that is the reference implementation of the Java API for RESTful Web Services, extending it with additional features and utilities to further simplify

³<https://jersey.java.net/>

RESTful service - to easily create a client that can communicate with REST services.

4 Conclusion

Existent environments for learning programming languages have a lot of specific requirements, which makes its installation and usability too complex for many students enrolled in introductory programming courses. The installation of these environments may not even be feasible in some devices, such as tablets.

This chapter presents the architecture and design of 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.

Enki provides a user interface deployed on the web, with programming tools, running on a remote server. Thus, it does not have specific requirements and does not require installation. Its user interface is also similar to those of traditional IDEs, such as Eclipse or NetBeans, so, it can be seen as a scaffolding tool to support the progress of the students to more complex environments.

Enki and its network will be subject to improvements. 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.

References

- Ahn, J., Butler, B. S., Alam, A., & Webster, S. A. (2013). 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.
- Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer science education*, 15(2), 83–102.
- Ali, N. H., Shukur, Z., & Idris, S. (2007). Assessment system for uml class diagram using notations extraction. *International Journal on Computer Science Network Security*, 7, 181–187.
- Al-Smadi, M., & Gütl, C. (2010). Soa-based architecture for a generic and flexible e-assessment system. In *Education engineering (educon), 2010 iee* (pp. 493–500).
- Blumenstein, M., Green, S., Nguyen, A., & Muthukkumarasamy, V. (2004). An experimental analysis of game: a generic automated marking environment. In *Acm sigcse bulletin* (Vol. 36, pp. 67–71).

- Brusilovsky, P., & Millán, E. (2007). The adaptive web: Methods and strategies of web personalization. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), (pp. 3–53). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Casquero, O., Portillo, J., Ovelar, R., Benito, M., & Romo, J. (2010). iple network: an integrated elearning 2.0 architecture from a university’s perspective. *Interactive Learning Environments*, *18*(3), 293–308.
- C. Li, R. H. U., Z. Dong, & Chasteen, M. (2013). Engaging computer science students through gamification in an online social network based collaborative learning environment. *Int. J. Inf. Edu. Technol.*, *3*(1), 72–77.
- Conde, M. Á., García, F. J., Casany, M. J., & Alier, M. (2010). Applying web services to define open learning environments. In *Database and expert systems applications (dexa), 2010 workshop on* (pp. 79–83).
- Dagger, D., O’Connor, A., Lawless, S., Walsh, E., & Wade, V. P. (2007). Service-oriented e-learning platforms: From monolithic systems to flexible services. *Internet Computing, IEEE*, *11*(3), 28–35.
- Ellis, R. K. (2009). Field guide to learning management systems.
- El Saddik, A., Fischer, S., & Steinmetz, R. (2001). Reusable multimedia content in web-based learning systems. *IEEE Multimedia*, *8*(3), 30–38.
- Fischer, S. (2001, March). Course and exercise sequencing using metadata in adaptive hypermedia learning systems. *J. Educ. Resour. Comput.*, *1*(1es). Retrieved from <http://doi.acm.org/10.1145/376697.376700> doi: 10.1145/376697.376700
- Gogoulou, A., Gouli, E., Grigoriadou, M., Samarakou, M., & Chinou, D. (2007). A web-based educational setting supporting individualized learning, collaborative learning and assessment. *Educational Technology & Society*, *10*(4), 242–256.
- Harjula, M., et al. (2008). Mathematics exercise system with automatic assessment.
- Juedes, D. W. (2003). Experiences in web-based grading. In *Frontiers in education, 2003. fie 2003 33rd annual* (Vol. 3, pp. S3F–27).
- Karampiperis, P., & Sampson, D. (2005). Adaptive learning resources sequencing in educational hypermedia systems. *Educational Technology & Society*, *8*, 128–147.
- Leal, J. P., & Queirós, R. (2011a). A comparative study on lms interoperability. In R. Babo & A. Azevedo (Eds.), *Higher education institutions and learning management systems: Adoption and standardization*. IGI-Global. Retrieved from <http://www.igi-global.com/bookstore/titledetails.aspx?titleid=50529&detailstype=description> doi: 10.4018/978-1-60960-884-2
- Leal, J. P., & Queirós, R. (2011b). Using the learning tools interoperability framework for lms integration in service oriented architectures.
- Leal, J. P., & Silva, F. (2003). Mooshak: a web-based multi-site programming contest system. *Software: Practice and Experience*, *33*(6), 567–581. Retrieved from <http://dx.doi.org/10.1002/spe.522> doi: 10.1002/spe.522
- Li, M., & Liu, Z. (2009). The role of online social networks in students’ e-learning experiences. In *Computational intelligence and software engineering, 2009. cise 2009. international conference on* (pp. 1–4).
- Paiva, J. C., Leal, J. P., & Queirós, R. (2015). Languages, applications and technologies: 4th international symposium, slate 2015, madrid, spain,

- june 18-19, 2015, revised selected papers. In J.-L. Sierra-Rodríguez, J.-P. Leal, & A. Simões (Eds.), (pp. 194–204). Cham: Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-27653-3_19 doi: 10.1007/978-3-319-27653-3_19
- Pisan, Y., Richards, D., Sloane, A., Koncek, H., & Mitchell, S. (2003). Submit! a web-based system for automatic program critiquing. In *Proceedings of the fifth australasian conference on computing education-volume 20* (pp. 59–68).
- Prados, F., Boada, I., Soler, J., & Poch, J. (2005). Automatic generation and correction of technical exercises. In *International conference on engineering and computer education: Icece* (Vol. 5).
- Queirós, R., Leal, J. P., & Campos, J. (2014). Sequencing educational resources with seqins.
- Queirós, R., Leal, J. P., & Paiva, J. C. (2016). Integrating rich learning applications in lms. In *State-of-the-art and future directions of smart learning* (pp. 381–386). Springer.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172. Retrieved from <http://www.tandfonline.com/doi/abs/10.1076/csed.13.2.137.14200> doi: 10.1076/csed.13.2.137.14200
- Rodrigues, J. J., Sabino, F. M., & Zhou, L. (2011). Enhancing e-learning experience with online social networks. *IET communications*, 5(8), 1147–1154.
- Ryzac. (2012). Codecademy. Retrieved from <http://www.codecademy.com/>
- Severance, C., Hardin, J., & Whyte, A. (2008). The coming functionality mash-up in personal learning environments. *Interactive Learning Environments*, 16(1), 47–62.
- Stewart, C., Cristea, A. I., Brailsford, T., & Ashman, H. (2005). 'authoring once, delivering many': creating reusable adaptive courseware.
- Vansteenkiste, M., & Deci, E. L. (2003). Competitively contingent rewards and intrinsic motivation: Can losers remain motivated? *Motivation and Emotion*, 27, 273-299. Retrieved from <http://dx.doi.org/10.1023/A:1026259005264> (10.1023/A:1026259005264)

5 Additional Reading

- Paiva, J. C., Leal, J. P., & Queirós, R. (2015). Languages, applications and technologies: 4th international symposium, slate 2015, madrid, spain, june 18-19, 2015, revised selected papers. In J.-L. Sierra-Rodríguez, J.-P. Leal, & A. Simões (Eds.), (pp. 194–204). Cham: Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-27653-3_19 doi: 10.1007/978-3-319-27653-3_19
- Queirós, R., Leal, J. P., & Paiva, J. C. (2016). Integrating rich learning applications in lms. In *State-of-the-art and future directions of smart learning* (pp. 381–386). Springer.
- Queirós, R., Leal, J. P., & Campos, J. (2014). Sequencing educational resources with seqins.

Paiva, J. C., Leal, J. P., & Queirós, R. (2016). Enki: A pedagogical services aggregator for learning programming languages. (in press)

6 Key Terms and Definitions

Gamification: the concept of applying game mechanics to engage and motivate people to achieve their goals.

Interoperability: the ability of a system to work with or use parts of another system.

Integrated Development Environment (IDE): a software application that provides facilities for software development. An IDE typically has a source code editor and build automation tools.

Sequencing of Educational Resources: presenting educational resources one after another.

Social Collaboration: the process of help and share information with other people to achieve a common goal.

Massive Open Online Course (MOOC): a course available over the Internet, without charge, to an unlimited number of students.

Scaffolding Tool: a temporary tool used to support the progress of a big change.