Journal of Visual Languages and Computing 31 (2015) 275–282

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

An experience of collaboration using a PaaS for the smarter university model ☆

Mauro Coccoli [a,*], Paolo Maresca [b], Lidia Stanganelli [c], Angela Guercio [d]

[a] DIBRIS, University of Genoa, Genoa, Italy
[b] DIETI, Federico II University of Naples, Italy
[c] Università Telematica, eCampus, Novedrate (CO), Italy
[d] Department of Computer Science, Kent State University, Kent, OH, USA

ARTICLE INFO

Available online 29 October 2015

Keywords:
Cloud computing
Smart applications
Collaborative systems
Technology enhanced learning
Smart university

ABSTRACT

In this paper we continue our previous research on the development of the current model of higher education, which pointed out that the labor market is looking for people with competencies and skills reflecting a T-shape model. As a consequence, universities should include a wider mix of disciplines in the curricula of their courses. Hence, to overcome existing criticisms and to provide some suggestions on how to enhance universities' performances, we thought of education as a process with inputs, outputs, and relevant dependencies. We called such a university a "smarter university" in which knowledge is a common heritage of teachers and students. In our research the smarter university model is based on a smart-city-like model, due to the fact that next generation networks and relevant services are going to be more and more integrated with existing infrastructure and information management systems. Thus, it is mandatory that smart solutions are the most prominent assets of modern university environments to improve the effectiveness of higher education. In this paper, we report the experimental results from a specific case study of collaboration between industry and university, which could be used as a reference for the definition of patterns to be applied in the redesign of the current education systems, even though the experiment refers to a technological application scenario.

1. Introduction

Owing to modern technologies, ever-growing computing power, miniaturization, and recent innovations in network infrastructures and networking solutions, nowadays people and things are connected to each other like never before. In this context, the most advanced solutions, such as the use of applications based on the Internet of Things (IoT) and the exploitation of semantic web capabilities, can bring innovation to e-learning [1] and push the interaction of people with both learning objects and learning environments in a real semantic web of things [2]. After analyzing this general scenario, in a previous work we observed the actual teaching model at universities. We found some weaknesses and strengths, and we identified the main building blocks for the definition of a reference model of what we called a "Smarter University" [3]. In our study, we refer to smarter university, instead of smart university, due to the fact that, generally speaking, today's universities widely adopt cutting-edge technologies and systems; thus we can consider they already have the desired smartness characteristics. However, that is not enough and they should become "smarter" to improve their effectiveness, enhance their performances, be more flexible, and, last but not least, be able to cope with the

☆ This paper has been recommended for acceptance by Henry Duh.

* Corresponding author.
E-mail addresses: mauro.coccoli@unige.it (M. Coccoli), paolo.maresca@unina.it (P. Maresca), ldistn@gmail.com (L. Stanganelli), aguercio@kent.edu (A. Guercio).

http://dx.doi.org/10.1016/j.jvlc.2015.10.014
1045-926X/

novel and emerging needs of both the modern society and the labor market. Currently, university teachers already perform cutting-edge research activities, which make them the exclusive holders of the knowledge, and they act consequently. From the theoretical point of view, this approach looks like a good model but, in the present conditions, the fact of having specific and deep knowledge in a very narrow research sector does not match with the labor market needs, which is focusing on flexibility and requires more and more interdisciplinary competencies. At present, it is not so easy to find university courses that provide such skills to their students. In fact, they should include a mix of technical skills and social sciences, and, at the same time, nurture abilities in management, social behavior and human interaction, communication, teamwork, problem posing and solving, creativity, lateral thinking, and resilience.

This is the hard work that universities are challenged to do. And "How should they achieve these objectives?" is the big question. Beside the disciplinary matters and the strategies leading to the choice of a suitable mix of classes in specific courses, we are convinced that one of the most powerful enabling factors to finding a solution is a tight collaboration between academia and industry, on common projects, with common objectives, to drive students to learn how to apply theoretical concepts for the solution of some real world problems. In such a vision, universities, organizations and companies cooperate to develop an ecosystem in which they could learn from each other. Hence, universities will be able to achieve the new "smarter" level and be ready to teach novel designs and methodologies and new reasoning paradigms, while industry and organizations could find new market shares to conquer. Finally, people could find new jobs.

According to this vision, and making reference to the pillars of smarter universities listed in [3], in this paper we focus our attention on a specific technological issue, that is, the exploitation of the Platform-as-a-Service (PaaS) paradigm in the production of software. The adoption of PaaS allows using remote virtual machines in place of local hardware and software, thus avoiding time-consuming and expensive installations as well as annoying maintenance tasks [4]. More specifically, in this paper we are going to showcase the outcome of an educational activity carried out in a distributed environment based on cloud computing and services. Specifically, we make reference to the most recent stage of a long-term project aimed to empowering collaboration skills in software engineering students, the Enforcing Team Collaboration (ETC) project [5]. In this project, students are grouped together to create small teams regardless of whether they might be from different universities, as well as from different countries. Then, within the tasks they are assigned, they must cooperate to achieve common objectives, which include, among others, the development of working prototypes of some web-based applications. All the students that enrolled in the laboratory could rely on a set of professional tools specifically crafted to support software development and for the management of the software lifecycle, which are made freely available at their universities through the IBM academic initiative. In this framework, previously we started experimenting with the Jazz ecosystem in conjunction with the renowned open source Eclipse Integrated Development Environment (IDE) and, in the last year, we designed and implemented the ETC-Blue project, which uses the IBM Bluemix platform and its facilities as PaaS.

The remainder of the paper is organized as follows. Section 2 explores related works. In Section 3 we present the ETC project, the framework in use, and we illustrate how the Eclipse framework becomes a learning environment. Section 4 outlines the ETC-Blue project. Finally, Section 5 shows the activities and the relevant results. Conclusions are reported in Section 6, where a glance on future work concludes the paper.

2. Related works

The Computer Supported Collaborative Learning (CSCL) strategy implemented through the PaaS paradigm can be found in the literature in other related works. According to Silverman [6], *"the adoption of a collaborative learning strategy can be useful in many situations (and it can be realized with or without the use of technology)."* Moreover, Dong et al. [7] assert *"the current models of e-learning ecosystems lack the support of underlying infrastructures, which dynamically allocate the required computation and storage capacities for an e-learning ecosystem. Cloud computing is a promising infrastructure which provides computation and storage resources as services."* In addition, in [8] the authors conclude that *"e-learning systems can use benefits from cloud computing using: Infrastructure (i.e. use an e-learning solution on the provider's infrastructure), Platform (i.e. use and develop an e-learning solution based on the provider's development interface), Service (i.e. use the e-learning solution given by the provider)"*. Given these general considerations, we observe that the use of the cloud is also reported in [9] where the authors describe how their students worked on a database in the cloud, in a virtual laboratory environment, and in [10] where the author states: *"hosting the entire development environment, PaaS increases productivity, lets organizations release products faster, and reduces software's cost."* In [11] the authors say that *"seamless and pervasive intelligence is already proving disruptive in education, with traditional campus-based education models changing as new teaching methods evolve, augmented by automated and interactive learning outside the classroom and distance participation"* and, even more, *"we also project that courses will involve less instruction and lecturing and more dialog with expert professors, resulting from the ability to use technology for interaction outside the classroom"*. The perspective is that education will be seamless and ubiquitous for those who can afford information technologies. To conclude this quick review, we mention [12] where the author says that *"the potential of cloud computing for improving efficiency, cost and convenience for the educational sector is being recognized by a number of US educational (and official) establishments."* and *"there is also an increasing number of educational establishments that are adopting cloud computing for economic reasons."*

Supported by these results, in this paper we propose an experiment in the line of the smarter education model to create applications in Java running on the Android operating system, to achieve the following advantages: (i) shorter learning time; (ii) better quality of products; and (iii) implementation of the T-shaped model [13].

3. Enforcing Team Collaboration

The ETC project was created to fulfill the need to develop cooperation skills between university students, skills that are a “must” when they are required to work in groups. Teamwork ability is an essential skill for students to acquire; learning and practicing this skill can give a preview of their future team-working experience. For our particular case study, we are talking about students involved in software engineering activities and, consequently, the ETC project creates an effective CSCL system for higher education that targets the area of software engineering, computer programming, and team cooperation for software analysis and software development [14]. In the rest of the paper we will refer to this particular case study. However, the authors believe that the same principles applied by ETC to the above areas, can be applied, with different tools, to other areas beyond computer programming.

We can face the problem of developing such skills from different points of view. On one hand, from the *educational* perspective, we observe that people must be duly trained to acquire competences in software engineering models and techniques, as well as in project management and human relations. On the other hand, from the *technological* perspective, we observe that a complex system is needed to enable and support collaboration as well as to ease interactions between the participants. Finally, we observe that distributed architectures and cloud computing can foster new behavioral paradigms in acquiring and disseminating knowledge and sharing experiences, thus they are needed in the learning process as well [15]. In support to this, we consider the vision on how recent advancements in grid and cloud computing and mobile communications have significantly changed many concepts at the basis of e-learning, as presented in [16]. We want to show that this approach can improve the learning outcomes and accelerate the education process, while making the design of courses, lectures and practical activities more flexible. Moreover, sharing resources and collaboratively constructing reusable learning assets can significantly reduce the costs in terms of both time and money [17].

The proposed solution is teaching both teamwork and computer programming in parallel. After looking among tools specifically created for supporting teamwork, a suitable software platform which enables effective teamwork was set up in order to coordinate the cooperation among students during code development. The students were enrolled in different universities, possibly in different countries, and had different working time frames.

3.1. The ETC development framework

Many different systems and tools are available for the coordination of the software development process activities. At the same time recent integrated environments are shifting the focus to remote cooperation, which is considered the best way to cut down time and money. However, for such environments to be effective, we need something like an “orchestra director”. Generally speaking, the orchestra director is the one who knows exactly when each instrument must be played, and how to leverage the quality of the overall execution. Specifically, we found all of these features within the Jazz development platform. This platform is usually adopted worldwide by IBM for the cooperative development of software. Before the ETC project was launched, such a complex platform had never been introduced in an academic setting. Within the ETC project, together with the Jazz environment we adopted the renowned open source Eclipse IDE as a development platform. It is worth noticing that, in addition to writing clean and working lines of code in their preferred programming language, students have to cope with other tasks such as debugging, compiling, and, finally, deploying activities on specific hardware platforms and operating systems. This requires the inclusion of specific resources to run distributed software architectures.

Based on such considerations, we can summarize that the ETC project consists of experimenting with the realization of collaborative activities in which different teams have to complete a group of tasks that have to be integrated with systems or subsystems developed by other teams. To reach the main project objective, the IBM Rational software tools were integrated into software engineering academic projects. IBM Italia sponsored the project and the University Federico II of Naples received an IBM Faculty award in 2011 for that. A number of Italian universities were involved: the University of Milano Bicocca, the University of Bologna Alma Mater, the University of Bergamo, the University of Genova and its regional campus in Savona, the University of Bari and its regional campus in Taranto. Heterogeneous teams were composed of students from different universities with one teacher as a tutor for each group. Moreover, for each University, a *champion student* (i.e., a Ph.D. student) was put in charge of corresponding with a teacher and acted as a manager for each local group and support. In addition, a Ph.D. student acted as a technical director of the entire ETC platform. Keeping into account the encouraging results achieved within the ETC experience, we aimed at building wider team cooperation projects from lessons learned in open communities of practice [18] and we designed new activities for groups of students that included the Kent State University, thus creating a more complex and broad working environment. The extended project was called ETC-plus [19].

3.2. The Eclipse framework as a learning environment

The Eclipse IDE can be considered as the inner center of a learning framework where students of computer programming write down their code and can easily interface with a number of external tools and services for a wide variety of purposes. In fact, Eclipse is an open universal platform for tool integration, an open and extensible IDE, and an open source community as well. We point out that integration is part of the software development process and it occurs through tools inside and outside the IDE. In order to maximize the collaboration results with minimal effort, we have incorporated the Jazz platform, which seeks to integrate collaborative capabilities into the Eclipse IDE, thus enabling small teams of software developers to work together in a more productive way [20]. Specifically, team cooperation in the context of ETC is enabled by the Jazz platform via the following tools: (i) Rational Team Concert (RTC); (ii) Rational Quality Manager (RQM); and (iii) Rational Requirement Composer (RRC).

An overview of the results obtained with the use of Eclipse on the Jazz platform in ETC is presented in [21]. After having successfully developed systems on the Eclipse-and-Jazz integrated platform in the ETC-plus project, since October 2014 we have considered experimenting with the IBM Bluemix tools in the ETC-Blue project. The use of this platform has made it easier to cope with issues related to the management of data, infrastructures, connectivity, and servers. In fact, its use allows a paradigm shift so that now we can exploit advanced solutions according to the Software as a Service (SaaS) model, on a Platform as a Service (PaaS) environment. Consequently, we do not have to worry about managing servers, databases, virtual machines, and multiple releases of instances. Furthermore, the extensive use of the cloud relieves us from data management issues, including security of both network and software.

4. The ETC-Blue project

The lessons learned from past experiences have driven us to the definition of a new scenario, which takes into account that requirements change very quickly. We found a suited solution in the recently released IBM Bluemix platform, which allows developers to use a combination of the most prominent open source computer technologies to power their apps, by handling the integration with apps and systems running elsewhere in a seamless way and managing data in the cloud [22]. We believe that the adoption of the IBM Bluemix platform will foster further developments and increase the educational results achievable through collaborative work activities, resulting in students that learn faster and acquire competencies and skills in different fields. To prove the validity and the effectiveness of the presented concepts, we have launched a new chapter of ETC, called the ETC-Blue project. The idea behind the experiment is that of "grafting" a university course onto a formalized company internal training process with the aim of getting T-shaped students. In this experiment we involved a pool of university students of a software engineering degree course with the main objective of (i) strengthening the vertical part of the T, which is made up of a deep and narrow knowledge of computer programming and operating systems; (ii) completing the horizontal layer of the T by developing skills in project management, collaboration, and leadership.

The project involved primarily the University of Naples Federico II and IBM Italia. The participating students were enrolled in the "Computer Programming – I" course of the Software Engineering degree. IBM supplied the students with a crash course on the Bluemix platform to make them aware of the features of the platform and to speed up the learning curve. In this way students could quickly focus on the design phase and start implementing sophisticated functionalities for their apps without having to worry about databases, server connections, security issues, etc., which are ready-to-use services of the platform, and thus transparent to the developer. The IBM Bluemix platform is accessed via a web browser; consequently no software has to be installed on local machines, allowing students to bring their own computers without having to rely on university facilities, and to continue studying and experimenting anywhere and at any time. This releases universities from maintaining a huge number of machines with fast-changing configurations in students' laboratories and provides mobility to students.

More specifically, the IBM crash course took two days, for a total of eight hours of lectures, including hands-on lab and it was run two times in November 2014. 120 Students attended the course but a total number of 150 people were involved in the project, including professors from the local University of Naples, nearby universities (i.e., the University of Cassino), a professor and a small group of students from the Kent State University at Stark, USA, Ph.D. students as well as professionals from different cities (i.e., from Milan).

Considering the variety of participants in the project, ETC-Blue is a concrete example of distributed team collaboration where working together may overcome geographic distances, time jet lag, and even different languages. The collaborative environment already designed, implemented, and intensively tested with the ETC-plus project, proved once more to be very effective and the first results achieved within this cooperation are encouraging.

5. Activities and relevant results

Immediately after the completion of the above-cited crash course, many students started individual projects in small groups. Specifically, 92 students were arranged in 26 groups of 4 or 5 people. Then, for each group 2 students were assigned the role of team leader and deputy, with the responsibility of the entire group project management and of external communications (i.e., with the teacher). To give a flavor of the activities carried out, in the following we list some of the running projects developed in the experiment and we provide a short description for each of them.

(1) **Knowledge Hound**. This work aims to help the community to build around specific activities carried on in the university both in education and research. In fact, if a student needs help in finding the solution to a specific problem, it is possible that other students are working on the same problem. With the help of the Knowledge Hound app, information can circulate easily and students can teach each other. In this respect, the project

participants develop a proximity-based app. The development of this project started on the IBM Bluemix platform and the outcome is an Android app, which runs on different devices such as smartphones, tablets. In Fig. 1 we show some screenshots of the application.

(2) **K12**. This project is the result of the collaboration with the Kent State University-Stark. Its objective is the creation of novel educational materials supporting both teachers and students in their relevant learning activities. As per its name, the project addresses K-12 students by providing open source and reusable learning resources, which can be customized to individual profiles. In Fig. 2 we show some screenshots from the students' app.

(3) **SmartApp**. This project has the objective of developing an app beyond a mere academic exercise, which will be delivered to the public via the app stores. Two different groups of 5 students each are currently working on the project. The app is addressed to traveling people and its main functionality is advising other apps based on the users' location. "Where are you? And, hence, what

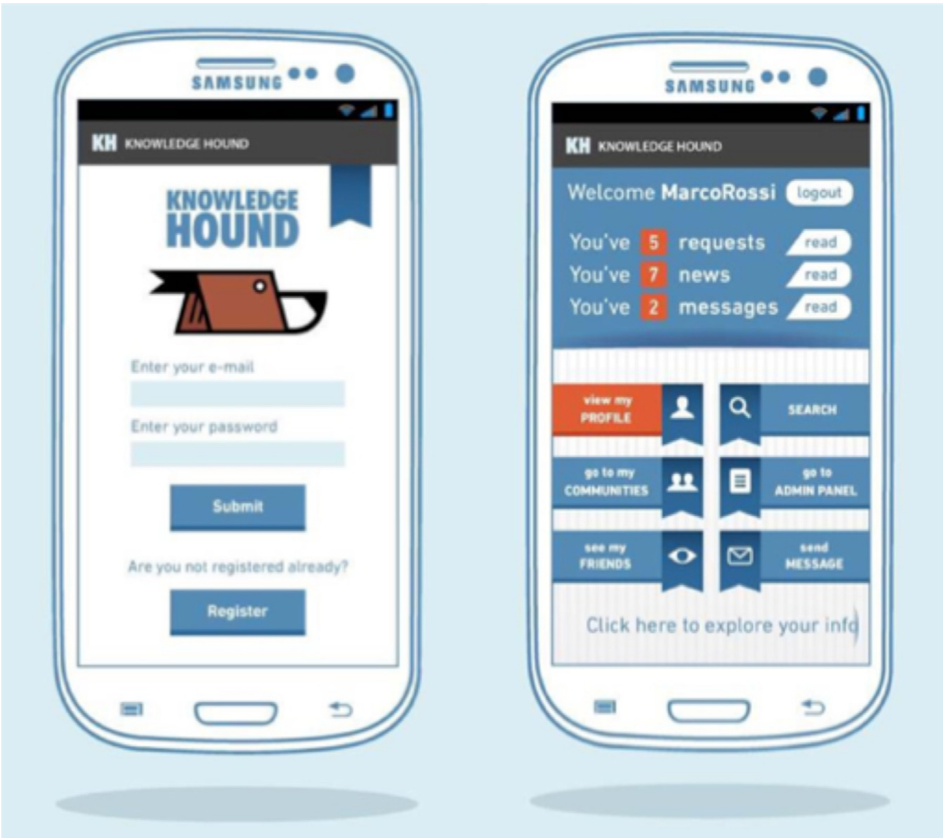

Fig. 1. The login/registration page and the home page of the Knowledge Hound app.

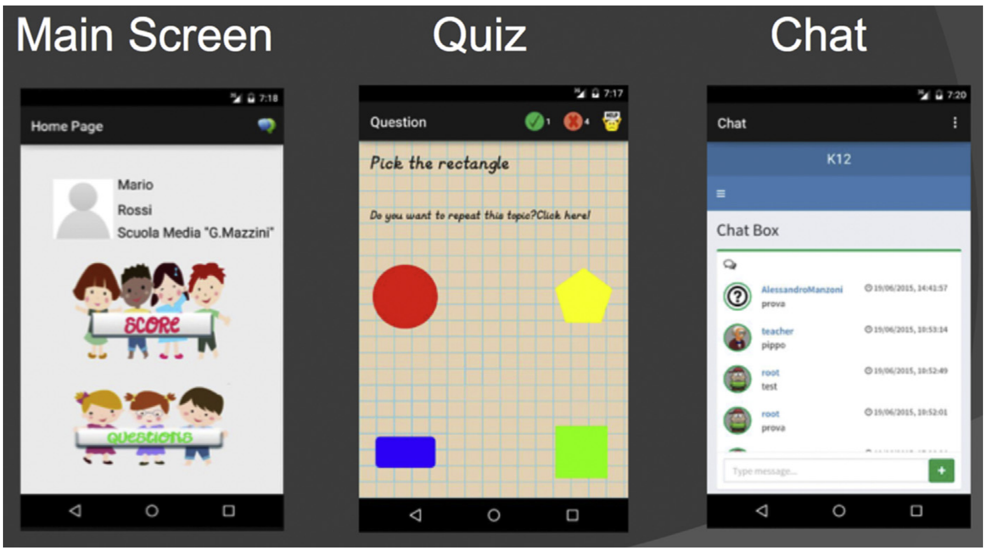

Fig. 2. The K12 app for math. Particular attention was given to a graphical interface appealing to kids.

specific apps could you need now?" Suggested apps could be, e.g., local transportation routes and timetables, local museums, local weather forecasts, and many others (even sponsored ones). In addition the app can ask, "Who are you?" and provide customized services.

(4) **ElectionUp**. This project involves 5 groups of 5 students each, and consists of designing a system to monitor election results. This involves real-time communication with a shared database and the need of suitable algorithms for quick data analysis and visualization. In addition, data should be accessible through common interfaces and APIs to other systems. Fig. 3 shows the user interface.

Summarizing, 12 groups are working on the "Knowledge Hound" app, 2 groups on the "K12" app, which includes a team of 4 American students, 2 groups on the "SmartApp" and 5 groups on the "ElectionUp" app for a total amount of 92 people working on the IBM Bluemix platform. At the completion of this stage, those students who have accomplished the mission they were given, assume the role of project managers for the newly entering students in the team, implementing turnover. Consequently 50 additional students have been charged with new tasks to improve the previously released apps, such as, e.g., fixing bugs, finishing uncompleted tasks, implementing new features, refining existing prototypes, and testing and evaluating the current work. It is worthwhile noticing that evaluation of the software performance and usability happens between peers, while assessment is the teacher's job.

At the present stage, two different teams have laid the foundations for a set of projects. The former developed some basic building blocks and a common knowledge base, which constitutes the substratum for forthcoming groups to operate on. The latter, developed a variety of interfaces for the Android operating system, by exploiting the Eclipse Android Development Tools (ADT) and the foundations provided by the IBM Bluemix boilerplates.

From the educational perspective, we highlight that students involved in this first stage have also developed specific training materials to enable future students to use the common workspace. In addition they also have set up suitable tools for the management and coordination of groups. But this educational activity has interesting points even from the pure software engineering perspective. In fact, the groups involved used the IBM Bluemix life cycle management tools illustrated during the initial crash course, passing through Jazz and DevOps. As an example, in Fig. 4 we show a screenshot taken from the IBM Bluemix dashboard. The picture illustrates the modules used within the Knowledge Hound app, which includes services for: (i) Mobile Application Security, (ii) Mobile Quality Assurance, (iii) Mobile Data, (iv) Push, (iv) SQL database server.

All of them are provided as a service by the Bluemix platform and this is a great advantage from both the maintenance point of view and the achievement of software engineering skills and design capabilities. In fact, students are freed from specific implementation issues and software availability at their universities, rather they can think about their computer programs in terms of their architecture and high level interfaces.

6. Conclusions and future work

In conclusion, the most important lesson learned from the ETC-Blue project is that joint efforts between the university and industry can give outstanding results and generate a stronger and supporting environment for the formation of T-shaped people. The main motivations beyond the ETC-Blue project are (i) students learn to cooperate in small teams; (ii) the collaboration implies a split of tasks and drives everyone to make the best of her/his effort and this implies that

Fig. 3. The ElectionUp app for the Italian election system. Particular attention was given to the graphical interface in order to be user friendly.

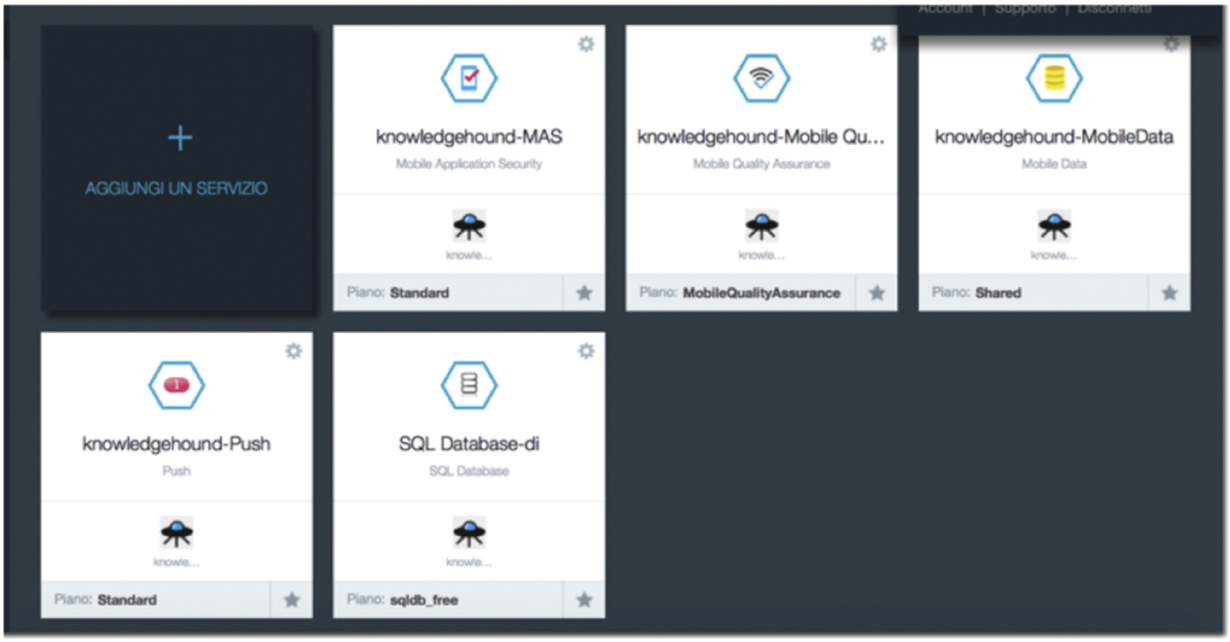

Fig. 4. The view from within the IBM Bluemix dashboard and the services used in the Knowledge Hound app.

individuals' competencies emerge. Among them, leadership is one of the abilities appearing more clearly and consequently, future e-leaders can be identified at an early stage, during university studies.

In this newly developed framework the collaboration between university and industry has been give-and-take. The university took the training courses from industry, contributing to the definition of their contents with the aim of readily exploiting them in specific projects for the dissemination at a student's level. The preliminary results demonstrate that the use of the IBM Bluemix platform, tacked onto the complex ecosystem based on the Eclipse IDE and seamlessly integrated into the Jazz products and solutions developed in past years, can greatly improve the performance of students, who gain core competencies faster and in a realistic working environment.

Summarizing, ETC-Blue fosters learning methods that are student-led versus instructor-led, with professors playing a mentor role in the learning process. This student-centric paradigm constitutes the basis for collaboration between people in teams and among groups. While observing the large number of students involved in this experiment, we noticed that some groups were *crawling*, other groups were *walking*, others were *running* and some were even *flying*. The teacher, acting as a coach, should identify those groups that fly and motivate them so that they can quickly achieve the best results and so that they can spread their knowledge to other groups, helping them to reach a superior level.

In a broader vision, we highlight that ETC-Blue can nurture the creation of smarter campuses, which are interconnected, enriched and fed by on-the-ground knowledge being developed over social networks. ETC-Blue favors the creation of smarter universities and forces teachers to have the most updated and relevant curricula, which then attract the best students who then will have the best formations, thus creating a virtuous circle of collaboration between universities and companies. In brief, ETC-Blue implements team-based projects across geographical, disciplinary and institutional boundaries and sustains a community that enables the formation of T-shaped people. Finally ETC-Blue fosters leadership and e-leadership.

Future work will be devoted to finalizing the projects that are still open and we want to encourage students to present their work to a wide audience of academics and professionals at a conference. Moreover, we want to carry on new experiments involving more companies, and even startups, to prove that (i) innovative working environments that push collaboration can enhance their productivity; (ii) that they can profit from university think-tanks through students' internships even before they graduate or by other means of collaboration, (iii) they can participate in this way in the education process, which can bring a great added value to consolidated realities. It is worth noticing that the same experience can be replicated in other settings regardless of the chosen PaaS platform. Another target is to launch an IBM Bluemix ecosystem. ETC-Blue has created a very powerful educational experience and, at the end of the activity, we observed a high degree of satisfaction in the students and a growth in personal appreciation as well. Despite this, some students complained about a steep learning curve. We have taken into account their feedback and have taken specific actions to overcome this problem in the future. To this aim, we have put significant effort into developing instructional materials in the form of video tutorials, user guides and handbooks on essentials, mind maps and more. Such learning assets will be readily available to future people involved in similar activities and they will be the starting point in their learning experience.

In addition we believe that the importance of a course offered by a company implements a different formative method. We like to explore further this innovative working method in which the university learns from companies and vice versa. In this way, we do believe that students are growing quickly and are able to organize themselves in most authoritative teams. The results so far are very good and will be investigated in a future research.

Acknowledgments

The authors wish to thank IBM and the Italian Eclipse Community for their friendship and their kind collaboration.

References

[1] G. Adorni, M. Coccoli, I. Torre, Semantic web and Internet of Things supporting enhanced learning, J. E-Learn. Knowl. Soc. 8 (2.) (2012) 23–32.
[2] M. Coccoli, I. Torre, Interacting with annotated objects in a semantic web of things application, J. Vis. Lang. Comput. 25 (6) (2014) 1012–1020.
[3] M. Coccoli, A. Guercio, P. Maresca, L. Stanganelli, Smarter universities. A vision for the fast changing digital era, J. Vis. Lang. Comput. 25 (6) (2014) 1003–1011.
[4] G. Lawton, Developing software online with Platform-as-a-Service technology, Computer 41 (6) (2008) 13–15.
[5] M. Coccoli, P. Maresca, L. Stanganelli, Enforcing team cooperation: an example of computer supported collaborative learning in software engineering, In: Proceedings of the 16th International Conference on Distributed Multimedia Systems, 2010, pp. 189–192.
[6] B.G. Silverman, Computer supported collaborative learning, Comput. Educ. 25 (3) (1995) 81–91.
[7] B. Dong, Q. Zheng, J. Yang, H. Li, M. Qiao, An e-learning ecosystem based on cloud computing infrastructure, In: Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies, 2009.
[8] N. Radhakrishanan, N. Poorna Chelvan, D. Ramkumar, Utilization of cloud computing in e-learning systems, In: Proceedings of the IEEE International Conference on Cloud Computing, Technologies, Applications & Management, 2012.
[9] E.P. Holden, J.W. Kang, G.R. Anderson, D.P. Bills, Databases in the cloud: a status report, In: Proceedings of the ACM International Conference on nformation Technology Education, 2011, pp. 171–176.
[10] G. Lawton, Developing Software Online with Platform-as-a-Service Technology, IEEE Computer Society, 2008.
[11] H. Alkhatib, P. Faraboschi, E. Frachtenberg, H. Kasahara, D. Lange, P. Laplante, A. Merchant, D. Milojicic, K. Schwan, What will 2022 look like? The IEEE CS 2022 report, IEEE Computer (2015).
[12] N. Sultan, Cloud computing for education: a new dawn? Int. J. Inf. Manag. 30 (2) (2010) 109–116.
[13] J. Spohrer, P.P. Maglio, J. Bailey, D. Gruhl, Steps towards a science of service systems, Computer 40 (1) (2007) 61–77.
[14] M. Coccoli, P. Maresca, L. Stanganelli, Computer supported collaborative learning in software engineering, In: Proceedings of the Global Engineering Education Conference, 2011, pp. 990–995.
[15] L. Caviglione, Enhancement of e-learning systems and methodologies through advancements in distributed computing technologies, in: J. Abawajy, M. Pathan, M. Rahman, A. Pathan, M. Deris (Eds.), Internet and Distributed Computing Advancements: Theoretical Frameworks and Practical Applications, 2012, pp. 45–69.
[16] L. Caviglione, M. Coccoli, V. Gianuzzi, Opportunities, integration and issues of applying new technologies over e-learning platforms, In: Proceedings of the International Conference on Next Generation Networks and Services, 2011, pp. 12–17.
[17] L. Caviglione, M. Coccoli, E. Punta, Education and training in grid-enabled laboratories and complex systems, Remote Instrumentation for eScience and Related Aspects, Springer, 2012, 145–157.
[18] P. Maresca, A. Guercio, L. Stanganelli, Building wider team cooperation projects from lessons learned in open communities of practice, In: Proceedings of 18th International Conference on Distributed Multimedia Systems, 2012, pp. 144–149.
[19] A. Guercio, P. Maresca, L. Stanganelli, Modeling multiple common learning goals in an ETC-plus educational project, In: Proceedings of 18th International Conference on Distributed Multimedia Systems, 2013, pp. 122–128.
[20] L. Cheng, S. Hupfer, S. Ross, J. Patterson, Jazzing up Eclipse with collaborative tools, In: Proceedings of the 2003 OOPSLA Workshop on Eclipse Technology Exchange, 2003, pp. 45–49.
[21] R. Frost, Jazz and the Eclipse way of collaboration, IEEE Softw. 4 (6) (2007) 114–117.
[22] K. Kobylinski, J. Bennett, N. Seto, G. Lo, F. Tucci, Enterprise application development in the cloud with IBM Bluemix, In: Proceedings of the 24th Inernational Conference on Computer Science and Software Engineering, 2014, pp. 276–279.