Teaching Computer Programming in a Platform as a Service Environment

Mauro Coccoli
DIBRIS
University of Genoa
Genoa, Italy
mauro.coccoli@unige.it

Paolo Maresca
DIETI
Federico II University of Naples, Italy
paolo.maresca@unina.it

Lidia Stanganelli
Università Telematica eCampus
Novedrate (CO), Italy
lidia.stanganelli@ecampus.it

Angela Guercio
Dept. of Computer Science
Kent State University
Kent OH, USA
aguercio@kent.edu

***Abstract*—In this paper we review 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 based our research 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 be the most prominent assets of modern university environments to improve the effectiveness of higher education. We called such a university a "smarter university" in which knowledge is a common heritage of teachers and students. In this paper, we report 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 it is referred to a technological application scenario.**

Keywords- cloud computing; smart applications; collaborative systems; technology enhanced learning.

I. INTRODUCTION

Owing to modern technologies, ever-growing computing power, miniaturization, and innovation in network infrastructures and networking solutions, 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]. Moreover, information and data are created and spread at high speed in settings with fewer and fewer boundaries. 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 may be not enough and they should become "smarter" to improve their effectiveness, to enhance their performances, to be more flexible, and, last but not least, to be able to cope with novel and emerging needs of both society and the labor market. In fact, currently university teachers perform cutting-edge research activities, which make them the exclusive holders of knowledge, and they act consequently. If regarded from a theoretical point of view, it looks like a good model but, in present conditions, the fact of having very specific and deep knowledge in a very narrow research sector does not match with labor market needs that are focused on flexibility and require more and more interdisciplinary competencies. At present, it is not so easy to find university courses that provide people with such skills. In fact, people with such skills should come from both technical faculties, such as, e.g., engineering, and social sciences ones; but at the same time they should hold knowledge and 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 it?" 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 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 should cooperate to develop an ecosystem together in which they could learn from each other. In this way, 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 the previous work, we focus our attention on a specific technological issue. In particular, we consider a novel research trend that is gaining consensus in the scientific community and that we expect to have a prominent role in the very next future, that is, the exploitation of the Platform-as-a-Service (PaaS) paradigm in software production

[4]. This will allow the use of remote virtual machines in place of local hardware and software, thus avoiding time-consuming and expensive installation procedures as well as annoying maintenance tasks. More specifically, in this paper we are going to showcase the outcome of an educational activity that we 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 ETC (Enforcing Team Collaboration) 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 throughout the Jazz ecosystem in conjunction with the renowned open source Eclipse IDE (Integrated Development Environment) and, in the last year, we started using also the IBM Bluemix platform and its relevant facilities.

The remainder of the paper is organized as follows. Section II explores related works, then in Section III we present the ETC Project and the framework in use. Section IV describes the Eclipse framework as a learning environment and Section V outlines the ETC-BLUE project. A summary of the results is reported in Section VI and a glance on future work concludes the paper.

II. 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 IBM Bluemix platform (the one we used in this work) 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 says: *"by hosting the entire development environment, PaaS increases productivity, lets organizations release products faster, and reduces software's cost."* Considering these results, this work proposes a smart education model which creates a CSCL and exploits the tools available in the IBM Bluemix platform to create applications in Java, which run on the Android operating system, and to achieve the following advantages: shorter learning time, better quality of prototypes/products, and implementation of the T-shaped model [11]. In [12] 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 *"we also project that courses will involve less instruction and lecturing and more dialogue 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 [13] 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."*

III. Enforcing Team Collaboration

The Enforcing Team Collaboration (ETC) project was created to meet the need to develop cooperation skills between university students 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. In particular, 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 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 behavioural paradigms in acquiring and disseminating knowledge and sharing experiences, thus they are needed in the learning process as well [15]. In support of 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]. In particular, we can envisage new learning models which ease the implementation of hands-on activities and can fully exploit users' interaction, due to the absence of local machines, physical devices and structures as well as working environments such as computer rooms with limited numbers of seats and access time constraints. We want to demonstrate that this can improve learning outcomes and accelerate the education process, while making the design of courses, lectures and practical activities to be assigned to students for the assessment and evaluation of their competencies more flexible. Moreover, sharing resources and collaboratively constructing reusable learning assets, can significantly reduce costs in terms of both time and money [17].

The project was born by noticing that in many cases the software production process can become hard due to lack of a full integration among the tools and meta-tools of different teams such as database, interface, processes, knowledge sharing, and so on. This has nothing to do with being able to design and write good code and can cause significant loss of time and demotivation during the learning process. Hence, the proposed solution is teaching both teamwork and computer programming in parallel. Based on novel programming paradigms and tools specifically created for supporting teamwork, a suitable software platform, enabling effective team work, was set up in order to coordinate cooperation in developing code among students enrolled in different universities, having different working time-frames, and possibly from different countries.

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. For such environments to be effective, we need something like an "orchestra director" over the development process. 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 complex platform, released by IBM, is usually adopted worldwide by IBM researchers for the cooperative development of software. Before the ETC project was launched, such a complex platform had never been used in an academic setting for a geographically distributed project. Consequently, in 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 the preferred programming language, students have to cope with other tasks such as debugging, compiling, and, finally, the deployment of activities on specific hardware platforms and operating systems. This requires the inclusion of resources necessary to run experimental distributed software architectures.

Based on such considerations, we can summarize that the ETC project consists of experimenting with the realisation of collaborative activities based on the Eclipse community and tools 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. The project was sponsored by IBM Italia and the University Federico II of Naples received an IBM Faculty award in 2011 for the project. A variety of Italian universities participated in the project: the University of Napoli Federico II, 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. Each university formed teams of students from different courses. Specifically, the software engineering course from the University of Napoli Federico II, the University of Bologna and the University of Milano Bicocca; the web design course for the University of Genova and the University of Bari; the advanced programming and testing course from the University of Bergamo. 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* (usually a computer engineering or computer science Ph.D. student) is put in charge of corresponding with a teacher and acts as a manager for each local group and support. One computer engineering Ph.D. student is put in charge of the technical direction of the entire ETC platform (both software and hardware).

Based on the encouraging results deriving from the ETC experience, we aimed at building wider team cooperation projects from lessons learned in open communities of practice [21] and we have extended the original project by designing new activities for groups of students that included the Kent State University, thus creating a more complex and broad working environment. The project was called ETC-plus [22].

IV. The Eclipse Framework as a Learning Environment

In this Section we discuss how 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 different purposes. In fact, Eclipse is an open universal platform for tool integration, is an open and extensible IDE, and an open source community as well. The aim of Eclipse creators, and hence of the Eclipse-based tools, is to give to developers the freedom of choice in a multi-language, multi-platform, multi-vendor environment supported by multiple vendors. In addition, Eclipse provides a unique environment for members of the academic community to build new tools for teaching, doing research, and fostering further growth of the Eclipse community [18]. 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 added the Jazz project, 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 [19]. In brief, 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);

(iii) Rational Requirement Composer (RRC).

These three tools assist teams in developing in cooperation software specifications while maintaining quality constraints.

An overview of the results obtained with the use of Eclipse on the Jazz platform is presented in [20].

After having successfully used and developed systems on the Eclipse-and-Jazz integrated platform in the ETC-plus project [21, 22], since October 2014, we have joined the IBM Bluemix program. The use of the IBM Bluemix 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.

V. 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 due to the fact that activities are bounded learning tasks. Also in this case we found a solution on the IBM shelf 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 [23]. 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 created the most recent release of the ETC project, 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 participants are the University of Naples Federico II and IBM Italia. The participating students are the ones enrolled in the “Computer programming - I” course of the Software Engineering degree at the Univ. of Naples. IBM has 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 available services of the platform, and thus transparent to the developer. Moreover, the IBM Bluemix platform is accessed via a web browser and no software has to be installed on local machines, allowing students to bring their own computers without having to rely on university facilities. In addition, they can continue studying and experimenting at home or anywhere else and at any time. This releases universities from maintaining huge number of machines with specific configurations in students’ laboratories, thus it is more in line with the model of virtual laboratories. The result is the saving of money and resources which could be used to improve other services and increase the students’ satisfaction.

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 Univ. 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 require taking into account significant geographic distances, different time, and even different languages. The collaborative environment, which was already intensively tested in the previous years with the ETC-plus project, has proved once more to be very effective and the first results achieved within this cooperation are really encouraging.

VI. 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, respectively, with the responsibility of the entire group project management and of the external communications (i.e., with the teacher). In the following we list some of the running projects developed in the experiment to give a flavor of the type of activities carried out, and we provide a short description.

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. For example, there are students who while working on a master thesis have acquired a deeper knowledge about that topic. With the help of the Knowledge Hound app, information can circulate easily and students can teach each other by exchanging and merging their own individual competencies. In this respect, the project develops a proximity-based app in which every student lists his/her personal abilities and skills and searches for his/her missing ones in other students’ profiles. 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, laptops and desktop computers. In Figure 1 we show some screenshots.

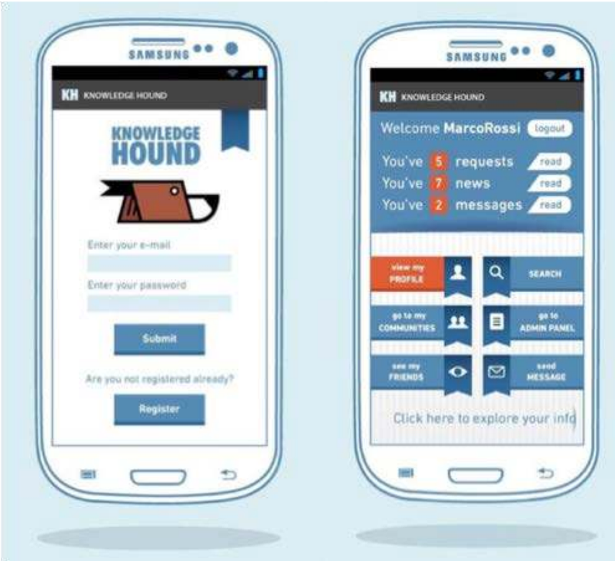

Figure 1: The login/registration page and the home page of the Knowledge Hound app

2) **K12**. This is a project launched in collaboration with Kent State University at Stark which fully exploits the features offered by the IBM Bluemix platform. The objective of this activity is defining innovative educational materials supporting both teachers and students in their relevant learning activities. As per its name, the project addresses K12 students by providing students and teachers with open source and reusable learning resources. Learning activities include, among others, quizzes at different difficulty levels, which can be customized to individual students' profiles. In Figure 2 we show some screenshots from the student app, composed of a main screen, a quiz screen and a chat.

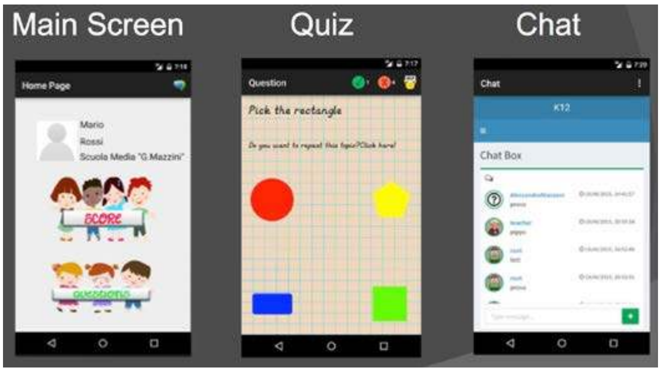

Figure 2: The K12 app for math. Particular attention was given to the graphical interface, to be attractive for kids.

3) **SmartApp**. This project has the objective of developing a real useful app, beyond a mere academic exercise, which will be delivered via the app stores to the public. Two different groups of 5 students each are working on the project. The target public should be composed of tourists or, generally speaking, traveling people. The main functionality of the app is the collection of apps based on location. "Where are you? And, hence, what specific apps could you need now?" Possible suggested apps could be, e.g., local transportation routes and timetables, local museums, local weather forecasts, and others. In addition the app can ask, "Who are you?" and provide the user customized replies by listing, for example, only apps related to music rather than sports events, finding contents in your language and suited to your age or other criteria.

4) **ElectionUp**. This project involves 5 groups of 5 students each, and consists of the design of an app made to follow real-time election results. This involves real-time communication with a shared database and the need of suitable tools and algorithms for data analysis and visualization. In addition data should be accessible through common interfaces and APIs to other systems. Consequently a user friendly interface is needed. Figure 3 and Figure 4 show the user interface and the data visualization screen. The app has been tested during the June 2015 elections in Campania, Italy.

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

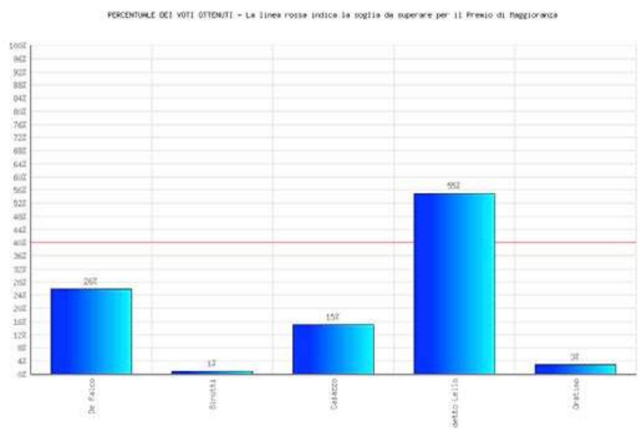

Figure 4: The testing of ElectionUp: the percentage of votes obtained.

Overall, 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 and duly finished their assigned tasks. At the completion of this stage, those students who have finished the activities they were assigned to assume the role of project managers for the newly entering students in the team. Consequently 50 additional students have been charged with new tasks to improve the previous apps, such as 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.

Based on this philosophy, the project's outcome can be incremental. Every team can start an assigned task by inheriting parts previously developed and add to or improve functionalities of an app that will become more and more complete, easy to use, and powerful. 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 ADT (Android Development Tools) and the foundations provided by the IBM Bluemix boilerplates.

From the educational perspective, we highlight that beside the development of the above-cited components, students involved in this first stage also specific training materials to enable future students to use the common workspace. In addition they also 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 Figure 5 we show a screenshot taken from the IBM Bluemix dashboard. The picture illustrates the modules used by the Knowledge Hound project. Each tile gives access to the configuration environment of each relevant service. They include services for:

(i) Mobile Application Security,
(ii) Mobile Quality Assurance,
(iii) Mobile Data,
(iv) Push,
(iv) SQL database server.

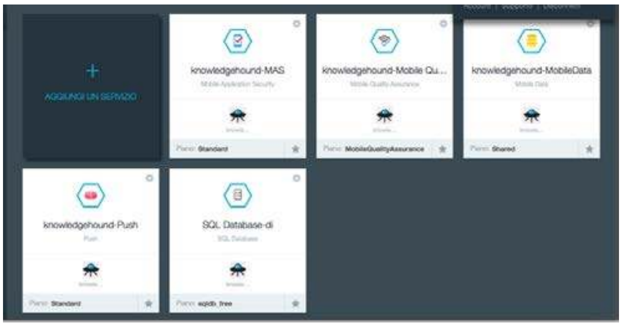

Figure 5: The services used in the Knowledge Hound app. The view from within the IBM Bluemix dashboard

Of course, 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. Moreover, we highlight that the adoption of cloud-based solutions for software services and storage could solve a huge number of problems for universities' computer laboratories that should not install and maintain numerous software packages for many different purposes. One sad note from the authors about this solution is that in Italy a large bandwidth connection is still a serious problem in many geographical areas and this can slow down the deployment of cloud services.

VII. Conclusions and Future Work

In conclusion, the most important lesson learned from the ETC-Blue project is that the joint efforts started between the university and industry can give outstanding results for a wide range of reasons. Ironically, it is again the collaboration between two entities (university and industry) that generate a stronger and supporting environment for the formation of T-shaped people. The main motivations beyond the ETC-Blue project are: students learn to cooperate in small teams; the collaboration implies a split of tasks and drives everyone to make the best of his effort and this implies that individuals' competencies emerge; among them, leadership is one of the abilities which clearly appears, and consequently, future e-leaders can be identified. In this newly developed framework the collaboration between university and industry has been give-and-take. The university took the training course 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 on to the complex eco system 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, we can say that ETC-Blue has reported several advantages and we observe that the most important part of the architecture is the collaboration (smart ETC-Blue). In fact, the collaboration has made possible the design and development of resources useful to all teams. Moreover, ETC-Blue drives standards and forces open innovation networks, requires mature organizations and produces high quality products. It is noted that if there are mature organizations that act as a driver of an experience like this, one can get high quality products (i.e., software) but also students trained in an excellent manner as reflected in the results achieved in our experience.

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 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. 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 dedicated to finalizing the projects that are still open. We want to reserve specific space for a student session within the forthcoming annual workshop of the Italian Eclipse community (*Eclipse-IT 2015* hosted in Rome, Italy, on October 14th, 2015), where they will have the opportunity to show to a wide audience of academics and professionals what they did. 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 by tracing the previous experiences made with the IBM Jazz platform through the various ETC project releases depicted in this paper. ETC-Blue has created a very powerful educational experience that students met with enthusiasm, attaining encouraging results. 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 criticism. To this aim, we have put significant effort into developing instructional materials in the form of video tutorials, user guides, 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.

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," Journal of E-Learning and Knowledge Society, vol.8, no.2, 2012, pp. 23-32.

[2] M. Coccoli, I. Torre, "Interacting with annotated objects in a semantic web of things application," Journal of Visual Languages and Computing, vol.25, no.6, 2014, pp. 1012-1020.

[3] M. Coccoli, A. Guercio, P. Maresca, L. Stanganelli, "Smarter universities. A vision for the fast changing digital era," Journal of Visual Languages & Computing vol.25, no.6, 2014, pp. 1003-1011.

[4] G. Lawton, "Developing Software Online With Platform-as-a-Service Technology," Computer , vol.41, no.6, 2008, pp.13-15.

[5] M. Coccoli, P. Maresca, L. Stanganelli, "Enforcing team cooperation: an example of computer supported collaborative learning in software engineering," Proc. of the 16th International Conference on Distributed Multimedia Systems, Workshop on Distance Education Technologies, pp. 189-192, 2010.

[6] B.G. Silverman, "Computer Supported Collaborative Learning," Computers Education, vol.25, no.3, 1995, pp. 81-91.

[7] B. Dong, Q. Zheng, J. Yang, H. Li, M. Qiao, "An e-learning ecosystem based on cloud computing infrastructure," Proc. 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," Proc. of the 2012 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," Proc. of the ACM International Conference on Information Technology Education, pp. 171-176, 2011.

[10] G. Lawton, "Developing software online with Platform-as-a-Service technology," Published by the IEEE Computer Society, June 2008.

[11] T in http://tsummit2014.org/t, last accessed July, 2015.

[12] 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, March, 2015.

[13] N. Sultan, "Cloud computing for education: a new dawn?" Int. J. of Information Management, vol.30, no.2, 2010, pp.109-116.

[14] M. Coccoli, P. Maresca, L. Stanganelli, "Computer supported collaborative learning in software engineering," Proc. of Global Engineering Education Conference, pp. 990-995, 2011.

[15] L. Caviglione, M. Coccoli, "Enhancement of e-learning systems and methodologies through advancements in distributed computing technologies," in 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," Proc. of International Conference on Next Generation Networks and Services, pp. 12-17, 2011.

[17] L. Caviglione, M. Coccoli, E. Punta, "Education and training in grid-enabled laboratories and complex systems," in Remote Instrumentation for eScience and Related Aspects, pp. 145-157, Springer, 2012.

[18] IBM, "Eclipse," http://www.research.ibm.com/eclipse, 2010, last accessed July, 2015.

[19] L. Cheng, S. Hupfer, S. Ross, and J. Patterson, "Jazzing up Eclipse with collaborative tools," in Proc. of the 2003 OOPSLA Workshop on Eclipse Technology Exchange, pp. 45-49, 2003.

[20] R. Frost, "Jazz and the Eclipse way of collaboration," IEEE Software, vol.24, no.6, 2007, pp. 114-117.

[21] P. Maresca, A. Guercio, L. Stanganelli, "Building wider team cooperation projects from lessons learned in open communities of practice," Proc. of the 18th International Conference on Distributed Multimedia Systems, Workshop on Distance Education Technologies, pp. 144-149, 2012.

[22] A. Guercio, P. Maresca, L. Stanganelli, "Modeling multiple common learning goals in an ETC-plus educational project," Proc. of the 19th International Conference on Distributed Multimedia Systems, Workshop on Distance Education Technologies, pp. 122-128, 2013.

[23] K. Kobylinski, J. Bennett, N. Seto, G. Lo, F. Tucci. "Enterprise application development in the cloud with IBM Bluemix," Proc. of the 24th International Conference on Computer Science and Software Engineering, pp. 276-279, 2014.