Teaching Computer Programming Through Hands-on Labs on Cognitive Computing

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

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

Lidia Stanganelli
Software Engineer
Italy
ldistn@gmail.com

***Abstract*—In this work we report the experience of a long-lasting educational project that we have been carrying since a couple of years. In particular, we summarize the results achieved by students in the last year, when they were put to work on the collaborative development of small, yet full featured, software projects. At the same time, based on more recent findings, we seek to lay the foundations to build a pragmatic model to teach cognitive computing programming. The experience was carried on in a Programming course at the Universities of Naples "Federico II" and Genoa, in Italy, and fostered the use of a PaaS (Platform as a Service) environment for a cooperative learning activity, used to disseminate theoretical concepts acquired within the course, also by means of cognitive computing tools. The project, from its inception, has involved a relevant number of students. Initially, the experiment had to be concluded in one year but, instead, has continued evolving with new projects, as new tools and services were made available, carrying new opportunities. The evolution has led, in the most recent release, to using the IBM Bluemix platform with its wide range of components, including Watson. This work goes in the direction of developing the smart university model, by using innovative and intelligent services to help develop a new generation of applications, but also to promote and disseminate a new way for designing and building them.**

Keywords: cloud; cognitive computing; collaborative systems; smart university; technology enhanced learning; computer supported collaborative learning.

I. INTRODUCTION

As already stated in previous research, collaboration is a paramount aspect in education, especially in computer-assisted instruction, resulting in the implementation of the Computer-Supported Collaborative Learning (CSCL) paradigm [1]. This also reflects in the architecture of widely adopted e-learning platforms, that include functionalities specifically developed for allowing users to collaborate and/or to cooperate, to the aim of reaching common objectives. Generally speaking, collaboration is a learning strategy that can be applied in educational activities specifically designed to bring learners to a given learning outcome throughout collaborative tasks. Following a collaborative methodology empowers the acquisition of skills, competence and knowledge, and reinforces the whole learning process, also enhancing mental capabilities of individuals, driving them towards acquiring a cooperative attitude and sharing resources with peers. However, in specific disciplines, such as, e.g., in software engineering classes, collaboration is one of the learning objectives. In fact, software is developed by heterogeneous teams including designers, analysts, programmers, testers, user experience designers, graphic designers, user interface designers, and many others, also depending on the dimensions of the projects. All these people must share common objectives, must have a common and clear vision of the project, and must be able to communicate each other seamlessly. Thus, a suited working environment is required, which includes many non-standard functionalities that are not present in common Learning Management System (LMS) since they usually serve for the delivery and of content and the accounting of users. Moreover, we notice that making concrete experimentations in laboratories equipped with suited facilities, is one of the core components of education along the fan of the scientific disciplines [2]. This is due to the fact that experimenting enables students to learn by doing through simulations and hands-on experiences, which are at the basis of many practical educational activities. Specifically, considering the case of teaching software engineering, the laboratory equipment is not made of machineries, yet it is all about computers, networks, operating systems, IDEs (Integrated Development Environments) and software programs, which can even be conveniently virtualized [3].

Within this context, we have been developing, appositely for computer programming classes, a dedicated working environment, which enables students, alone or in small groups, working on small and simple tasks within more complex software engineering projects, covering the whole lifecycle of software development across analysis, design, and development. Furthermore, this architecture was also designed to allow for subsequent upgrades by other students in future years. Such a working environment was designed with reusability in mind and is continuously evolving, as new solutions emerge, which can be profitably exploited by both students and teachers. This reflects positively on learning methodology, fostering an innovative model for computer programming classes, but also results in dramatic changes in universities from an infrastructure and organizational point of view [4]. In fact, one of the main issues in making available a full featured flexible, sharable and reusable working environment is the fact that it clashes with the static structure of existing software engineering laboratories. Traditionally, universities have invested much money, and human resources as well, in setting up and maintaining large and costly computer rooms with fast Internet connections for their students' satisfaction. However, it is evident that this model has many drawbacks such as, e.g., *(i)* the need to be reconfigured according to the specific requirements that may come from different courses; *(ii)* the need of keeping updated, consistent and protected, the software installed as security patches or new versions are released; *(iii)* many security and safety threats; *(iv)* the need of specific and costly hardware supporting huge loads, even if concentrated in short periods, i.e., the few "lab-hours" scheduled during a semester; *(v)* the rapid obsolescence of hardware. All of these issues can be summarized in just one point: very high costs. Moreover, in recent years, other inefficacy reasons emerged, due to technology progresses and changes in the society and in people's lifestyle, habits and attitudes. In fact, modern construction techniques guarantee ever-growing computing power and progressive miniaturization rates, which result in affordable and capable devices. In addition, the fast and continuous innovation in network infrastructures and networking solutions [5] results in the straight connection between people and objects and in the rapid development of Internet of Things (IoT) applications and solutions, also bringing innovation to technology-enhanced learning [6]. Furthermore, the wide availability of applications for sensor-rich modern devices pushes to using methods and techniques strongly based on the interaction with users, which happens mainly through natural interfaces.

Accordingly, in this paper we focus our attention on a specific issue, that is, the exploitation of the Platform-as-a-Service (PaaS) paradigm in the software production [7]. 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 [8]. Moreover, we also focus on cognitive computing, which is a very challenging item in the software development scenario since it allows developing applications that manage big data, enables performing analytics on data in order to make complex decisions, and fosters interaction between machines and persons through natural language, both written and spoken. Cognitive computing will be increasingly linked both to IoT and analytics because of the large amount of data collected and the subsequent decision that can be taken once these data will be analyzed [9]. Especially, if we combine the pattern matching with the complex verbal communication, training students in our labs on these issues is necessary and important for the future applications and the formation of software engineers.

The remainder of the paper is organized as follows. First, in Section II, we present a summary of past experiences, which have led to the actual situation. Then, in Section III we introduce general issues about cognitive computing. In Section IV we report experiment with cognitive computing and the use of the IBM Watson cognitive computing facilities into several projects. Finally, in Section V, a glance on future work concludes the paper.

II. A Summary of Past Experiences

In this section, we report on the previous experiences that improved collaboration between students and drove the project to the current state. This evolution path must not only be seen as a recasting of the instruments used, which also served to enhance collaboration, but should also be regarded as the maturation of the way in which we obtained a strong evolution in the control management and implementation processes of collaboration between people from diverse groups and countries.

A. Enforcing Team Cooperation

In March 2010, a first project started from the collaboration between the Federico II University of Naples, IBM Italy, the Eclipse Italian community, and other Italian universities, later joined by universities from abroad too. The aim of the project was the collaborative creation of good-quality software products within university courses, to improve students' learning and to enhance the collaboration capabilities between members of the development teams. This resulted in a new way of teaching and bridged the gap between companies and universities, with the objective of preparing students to face at the best the labor market. The project was called ETC (Enforcing Team Cooperation) and it was attended by the following universities: Naples Federico II, Bergamo, Milano Bicocca, Genoa, Bari, Bologna. The project involved about one hundred students and was centered around the IBM Rational's Jazz vision (see references [10] and [11] for a more detailed description).

B. On the Road to Eclipse

The ETC methodology was successful and it was applied one year later, 2011, in a larger project called OTRE (On the Road Eclipse Platform) [12] for the realization of a collaborative project centered around the Eclipse community and tools; at the end, the results were collected, published and shared, in a workshop in which the teams that had collaborated remotely finally had the opportunity of meeting themselves as well as comparing their work and results with other teams involved in the same project. Something like a peer-to-peer conference, that wanted to be much more than a classic student contest.

C. Trans-Continental Collaboration

The next year, 2012, we faced a trial centered on cooperation of student groups belonging to both Italian and non-Italian universities [13]. The project was very ambitious and involved two classes: a programming class and a computer science II class. The former from the Federico II University of Naples (Italy) and the latter from the Ohio State University at Stark (USA). The goal was to form the heterogeneous classes on a platform (Rational-Jazz) and set up a 24 hours workshop that students could use at any time. As a consequence, a common training on common educational objectives could be achieved by standardizing the programs and using the same language.

D. Geographically Distributed Teams

The outcome of this trial was very satisfactory for both teachers and students, hence it was extended and repeated the next year 2013 [14]. New projects were set up with mixed teams that represent portions of both Italian and American students. This made it possible to tackle problems in training places such as, e.g., difficulty in speaking, different study habits, organizational habits, etc. We are proud to mention that this project has received a special mention by IBM, as the best practice in collaboration of the year[1].

The above-cited experiences, contributed to the definition of the model of smarter university given in [4], which depicts the progresses that universities are doing/should do to provide their students with smart services. These are included in remote laboratories and can be used everywhere, for all time and without the need of reservation. This concept finds its practical realization in the newly acquired possibility of using a set of cloud applications in a PaaS environment, i.e., the IBM Bluemix platform [7]. With this further step, students are enabled to cooperate, develop, maintain and manage any project with any team at any time without the need to move to any physical laboratory. The philosophy of virtual all without any installation in any laboratory or PC and the ability to use hardware resources and software for free (owing to the IBM Academic Initiative) make this a very powerful service. Finally, we notice highlight that all of these premises are changing the programming paradigms and are driving applications to demand more cognitive services, as it will be discussed in the next section.

III. COGNITIVE COMPUTING AND WATSON

Cognitive computing can be regarded as the computer simulation of the human reasoning processes. To give a more detailed definition, we must make some considerations on the concept of programming. More precisely, we recall that the era of programmable computing starts with the dawn of the digital computer around 1940. The big change introduced is that now one has general purpose computing systems that are programmable and that can be reprogrammed to perform different tasks, according to the needs of the moment. But ultimately, they have to be programmed and are still somewhat constrained in the way they interact with humans. This is the era that we're currently in. However, we see emerging over the last few years what we call cognitive computing, as the result of summing a number of factors. In fact, there are at least four issues that affect the cognitive computing.

A. Big Data

The first factor is the emergence of big data, originated by ever-growing number of services and applications that we daily use over the World Wide Web and the Internet, exchanging information with both humans and computers, and creating new knowledge. Apparently, we are not prepared to handle this huge amount of data and this results in using them rarely, wasting big opportunities and valuable resources.

B. Amplifying Human Cognitive Boundaries

The second factor is due to the need to amplify the human cognitive boundaries. It is unquestionable that our ability to reason and think deeply, as well as to solve complex problems, is really quite impressive. However, our ability to read, analyze and process huge volumes of data is really quite poor.

[1] IBM Academic Award http://ibm.co/1Og2YNt

C. Easing Communication

The third factor is due to the need to easily communicate to both machines and persons through natural language. So, instead of writing and editing programs that continuously interact with machines we can just talk and educate them about things to do. At the same way, machines can talk to other machines or humans in a dynamic IoT always-connected scenery.

D. Analytics

Finally, the fourth factor, which can be seen as a substrate, is analytics. This factor is more properly the key element of cognitive computing applications and is represented by the intense use of machine learning [15]. These four enablers push the cognitive computing. Probably there will be a transitional or long period in which traditional programming and cognitive programming will coexist and will overlap.

In this scenario one can understand why the IBM Bluemix platform and Watson co-exist. In fact, they are the way to share the two above-described programming paradigms. Figure 1 shows the Watson services available within the Bluemix catalog.

Figure 1. Watson boilerplates

In more detail, Watson is a full-featured cognitive computing system for the research and development of cognitive systems and services [16], [17], which can easily interoperate with other applications and legacy systems as well. Owing to the IBM Academic Initiative, universities can freely experiment Watson through the ever-growing variety of services and APIs (Application Programming Interfaces), made available within the IBM Bluemix platform [18], which represent an important and promising field of application for the exploitation of cognitive computing services is education [19].

Some of the many available Watson boilerplates will be used within the programming course for the creation of *cognos* for smartphones and personal computers. For example, the most used are: *Dialog*, *Text to Speech*, *Speech to Text*, *Visual Recognition*, *Language Translation*, and *Personality Insights*.

IV. THE PROJECTS

During the course of Programming I, in the academic year 2015/2016, the Federico II University of Naples launched the second trial with IBM Bluemix. Specifically, this year, the students carried on 14 projects. Among these, some are already finished and others are still running but nearing completion. Students working in about 50 groups (for a total of about 150 people involved) have crafted applications for both smartphones/tablets and PCs. As in previous years, the students were divided into groups and each of these was assigned a part of the project.

To better clarify what we mean with “students’ project”, we report the ones completed, which are currently in testing phase. The projects are, namely: *(i)* Bag of ideas, *(ii)* Sbulloniamoci, *(iii)* Cancer registry, *(iv)* Restart Campania, *(v)* Arduino car.

A. Bag of Ideas

The application deals with the creation of a Social Network Site (SNS) for sharing and searching opportunities for the development of ideas with some interesting features. In other words, you want to build an original social network that is able to provide potential customers with the possibility of requesting assistance and expertise. It works as follows: those who have an innovative idea but not the ability to achieve it, can post it on the site and ask for help (see Fig. 2). The administrators of the site make available skills and means to realize the idea. Valuable is the analytics service that collects data analyzes them and provides a feasibility hypothesis idea considering the technological drivers, territorial requirements, the existence of such ideas etc., in short applies analytics algorithms. Equally important is an application that allows to manage the reputation of the person or company through the judgment that others provide within the community. Reputation is managed through a scale of 10 values: five positives and five negatives.

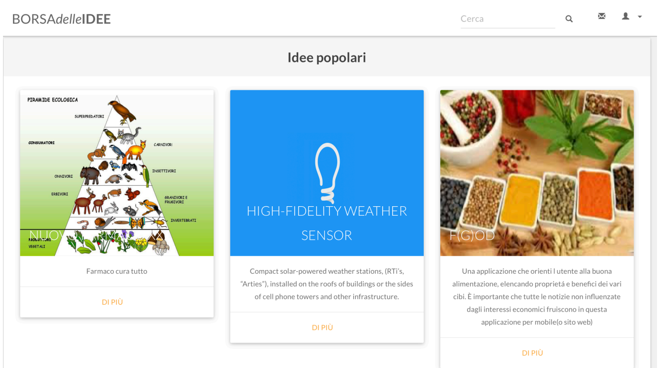

Figure 2. Bag of ideas

B. Sbulloniamoci

The main purpose of this web application is searching over the most popular SNS (e.g., Facebook and Twitter, also exploiting known provacy flaws [20], [21]) for possible threatening messages, in order to recognize the profile of a bully and report it to the community. An existing application that poses the same goal is ReThink: it analyzes the messages that the person who installed it is sending. In the event that they contain threats, the application displays the message «Are you sure you want to do?».

Sbulloniamoci (Fig. 3) wants to be more decisive, exposing the subject to the community, putting plenty of data on it (for example, name, address, etc). The execution of the application consists essentially of two phases. The first is the interaction with the user, achieved through use of *Dialog* from Watson. The second key stage of the application is the elaboration of the text written by the “bully” potential. This processing is carried out by *Personality* from Watson. It allows analyzing a text, to extrapolate the psychological profile of the person who writes, showing its main characteristics, as well as what actions he may or may not accomplish. Note that *Personality*, in order to obtain a meaningful result, needs about 3500-6000 words. Obviously, it is able to provide an even profile with fewer words, but the results will be less reliable. Before using *Personality*, it can be very useful text through the facility *Concept Expansion* and *Concept Insights*.

Figure 3. Sbulloniamoci

Concept Expansion helps "translate" any youth language or slang of the subject in more clear terms (for example, you can connect the words "Big Apple" to "New York"). A pre-processing via Concept Expansion could also be useful for user input. Moreover, *Concept Insights* allows you to link certain words to concepts; thus, it can detect any threats, insults, or aggressive language that may not be detected by scanning the text "traditional."

C. Cancer Registry.

The project's main aim is to build a website and an app that can show users the data for the more developed tumors in a given geographical area. The reason is to relate pollution (air, water and waste management in general) with the increase of cancer in certain areas of the country. The work has been divided into 4 groups as follows: the first dealt with the creation of the database; the second with the analysis of the data; the third took in charge the realization of the application (currently available only for Android OS devices, it will be built for iOS soon); the last group has realized the website. This is the project in which more than all have worked analytics on data from disparate sources, In fact, we used data from the National Cancer Registry, air pollution data, water, data from Google maps, etc. It must be said that, in this case, it was hard identifying a boiler plate tailored for the project. We are considering the possibility to develop one.

Analytics allow you to analyze data from different sources. In the chart of Figure 4, patient age hospitalized for tumors in the 2003-2013 decade is depicted. This chart allows knowing the age with more incidence of tumors in the municipality selected.

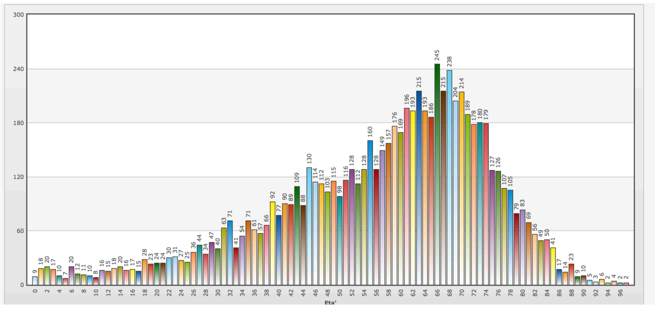

Figure 4. In this graph patient age hospitalized for tumors is analyzed in the 2003-2013 decade

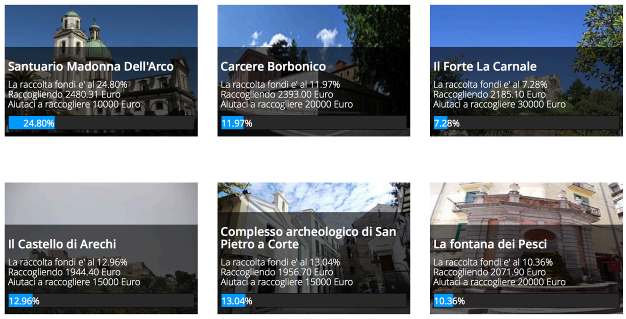

Figure 6. Photos of monuments of the region and relevant collected cash donations

D. *Restart Campania.*

The project consists of an application that allows you to stimulate and promote tourism in the Italian region Campania. One of the flagship applications is called visual recognition. For it is being operated as visual recognition of Google. The user sees a monument, photographs it, recognizes it and can adopt it by making a donation or by creating or participating in a community that, promoting it, develops the area and attracts visitors. It was performed on an app and a website. The work is to achieve an image recognition service, and creating a map in which the locations of the monuments are marked. For recognizing images was used the Visual Recognition tool, the image is sent to a server and subjected to matching with a database of images. The Visual Recognition allows to recognize monuments or artistic works from any user image taken; accordingly, to provide the user with more information on the selected image, it is automatically performed a Google search result. The monument is then identified on a Google map (see Fig. 5).

As for the map, inside there are markers for the locations of the monuments that have already been reported by users. Clicking on a marker, the user can display a link to the video (see Fig. 6).

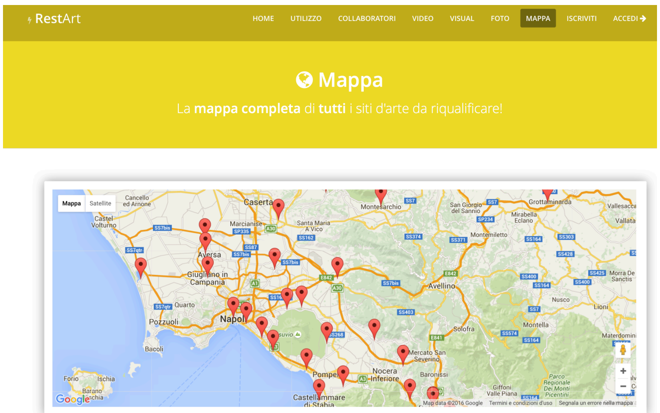

Figure 5. Map of adoptable monuments

E. *Arduino Car*

The Arduino Car project received the special mention by IBM during the last Eclipse-IT workshop, “Head in the Clouds”, which took place in Rome on last October. Arduino Car is the design and development of a real, working, toy-machine controlled by means of a smartphone app, developed within the IBM Bluemix PaaS and exploiting Watson functionalities. Figure 7 shows the prototype at its current state of development.

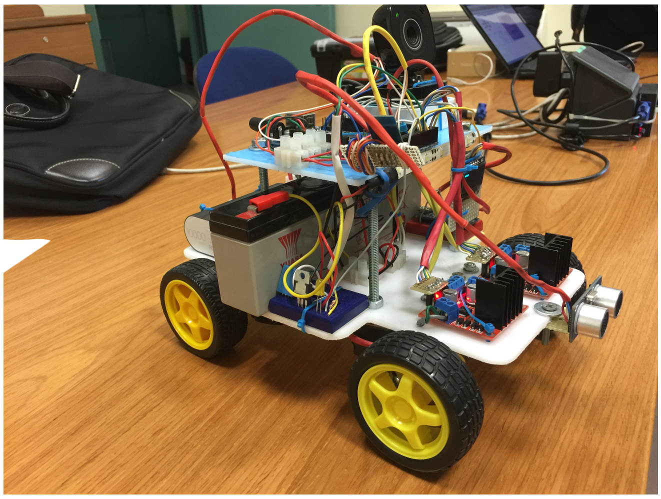

Figure 7. The Arduino Car

The project also involved students enrolled in the course of studies in mechanical engineering for engine design and on-board electrical systems to control the 2-wheels drive traction system. Aerodynamics is unsatisfactory yet. In more detail, it carries 4 different mini-cameras, heat sensors and distance sensors. The software developed gathers data, analyzes them and provides support to the decisions app that monitors the car’s performance. The working group that developed the project has had to deal with the hardware, the on-board software, the communication network, also developing a communication protocol between Watson and the control system. Future developments will be the voice command and a general improvement of body and aerodynamics of the vehicle.

V. Conclusions

In conclusion, we observe that teaching computer programming can be performed through a variety of strategies and based on different languages or IDEs according to the teachers' preferences and to the available facilities in terms of both software licenses and hardware. However, due the practical nature of such a discipline, we noticed that better results are achieved when students face to realistic problems and realize real working applications. This cannot be attained starting programs from the scratch but requires the ability of using collaborative methodologies and of reusing libraries already available, with an incremental approach to the construction of full-featured software prototypes. This is also the requirement of the labor market, which needs flexible professionals with problem solving capabilities and social attitudes. Using the ecosystem introduced within the paper, that is the IBM Bluemix PaaS and the relevant services, in hands-on labs of the Programming course has guaranteed a high quality of the final artifacts that were used for evaluation by the teachers. Moreover, through the PaaS, we can exploit advanced functionalities based on a very complex cognitive computing system, that is the Watson services. Such services, perform high-level operations and require high investments in terms of hardware computational capabilities and research and development as well, that could not be faced in a university class. Since the platform makes them available for free, this can empower all the applications with high level functionalities at no cost, also enhancing the satisfaction level of the students and their performances. Consequently, this reflects on a better performance from the teacher side too.

Acknowledgments

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

References

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

[2] 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.

[3] R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni, F.C.M. Martins, A.V. Anderson, S.M. Bennett, A.Kagi, F.H. Leung, L. Smith, "Intel virtualization technology," Computer, vol. 38, no. 5, pp. 48-56, 2005.

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

[5] L. Caviglione, M. Coccoli, V. Gianuzzi, "Opportunities, integration and issues of applying new technologies over e-learning platforms," in Proceedings of the 3rd International Conference on Next Generation Networks and Services, NGNS'2011; Hammamet; Tunisia; Dec. 18-20, 2011, pp. 12-17.

[6] 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, pp. 23–32, 2012.

[7] M. Coccoli, P. Maresca, L. Stanganelli, A. Guercio, "An experience of collaboration using a PaaS for the smarter university model," Journal of Visual Languages and Computing, vol. 31, no. 1, pp. 275-282, 2015.

[8] G. Lawton, "Developing software online with Platform-as-a-Service technology," Computer, vol. 41, no. 6, pp. 13–15, 2008.

[9] J.E. Kelly and S. Hamm, "Smart machines: IBM's Watson and the era of cognitive computing,". Columbia Business School Publishing, 2013.

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

[11] M. Coccoli, P. Maresca, L. Stanganelli, "Computer Supported Collaborative Learning in software engineering: (ETC - Enforcing Team cooperation)," Proceedings of the 2011 IEEE Global Engineering Education Conference, EDUCON 2011, Amman, Jordan; April 4-6 2011, pp. 990-995.

[12] P. Maresca, L. Stanganelli, M. Coccoli, "Managing a software project leveraging students' cooperation: on the road to Eclipse (OTRE) experience," in Proceedings of 12th International Conference on Product focused software development and process improvement: PKMT-Project and knowledge management trends, Torre Canne (Br), Italy, June 20-22, 2011, pp. 96-100.

[13] P. Maresca, A. Guercio, L. Stanganelli, "Building wider team cooperation projects from lessons learned in open communities of practice," in Proceedings of the 18th International Conference on Distributed Multimedia Systems (DMS2012) – Distance Education Workshop, Miami Beach, USA, August 9-11, 2012, pp. 144-149.

[14] P. Maresca, A. Guercio, L. Stanganelli, "Modeling multiple common learning goals in an ETCplus educational project", in Proceedings of the 19th International Conference on Distributed Multimedia Systems (DMS2013) – Distance Education Workshop, Brighton, United Kingdom, August 8-10, 2013, pp. 122-127.

[15] S. Earley, "Executive roundtable series: machine learning and cognitive computing," IT Professional, vol. 17, no. 4, pp. 56-60, 2015.

[16] G. Booch, "The soul of a new Watson," IEEE Software, vol. 28, no. 4, pp. 9-10, 2011.

[17] S. Sudarsan, "Evolving to a new computing era: cognitive computing with Watson," Journal of computing sciences in colleges archive, vol. 29, no. 4, pp. 4-4, 2014.

[18] K. Kobylinski, J. Bennett, N. Seto,G. Lo, F. Tucci, "Enterprise application development in the cloud with IBM Bluemix," in Proceedings of the 24th Annual International Conference on Computer Science and Software Engineering (CASCON '14), Markham, Canada, 2014, pp. 276-279.

[19] M. Coccoli, P. Maresca, L. Stanganelli, "Cognitive computing in Education," Journal of e-Learning and Knowledge Society, vol. 12, no. 2, pp. 55-69, 2016.

[20] L. Caviglione and M. Coccoli, "Privacy problems with Web 2.0," Computer Fraud and Security, vol. 2011, no. 10, pp. 16-19, 2011.

[21] L. Caviglione, M. Coccoli, A. Merlo, "A taxonomy-based model of security and privacy in online social networks," International Journal of Computational Science and Engineering, vol. 9, no. 4, 2014, pp. 325-338, 2014.