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
Naples, Italy
lditin@gmail.com

Abstract—In this work we report more recent results of a long-lasting educational project that we have been carrying on for several years and is evolving continuously. The objective of the mentioned project is making students work on the production of small, yet full featured, software prototypes with a collaborative approach, in a dedicated development environment with suited tools and facilities. At the same time, we seek to lay the foundations to build a pragmatic model to teach cognitive computing. We conducted such experience in a programming course at the University of Naples “Federico II” exploiting a software development environment based on the Platform as a Service (PaaS). This made possible to perform a set of cooperative learning activities that we used to demonstrate practically some theoretical concepts, also stressing the use of cognitive computing tools, which introduce a new way of thinking the software design process. From its inception, this educational project has involved a relevant number of students, which has been growing continuously, year after year. At our first attempts, students were assigned a specific activity, which was expected to be concluded within one year. Then, as new classes arrived, we have seen that it was possible to start new activities starting from the precedent achievements, thus requiring further evolutions of the available prototypes. This allowed creating more complex and complete projects, as new tools and services were made available, carrying new opportunities. In the present release, such evolutionary path has led to using the IBM Bluemix platform with its wide range of components, including Watson that is devoted to cognitive computing. This work goes in the direction of developing the smart university model, by using innovative and intelligent services to help raising a new generation of software engineers but also to promote and disseminate a new way for designing and building innovative applications.

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 learning strategy that can be applied in educational activities specifically designed to bring learners to a given learning outcome throughout collaborative tasks. By adopting collaborative methodologies, teachers can empower the acquisition of skills, competence and knowledge, thus reinforce the whole learning process, also enhancing mental capabilities of individuals, driving them towards acquiring a cooperative attitude and sharing resources with peers. Especially, collaboration is a paramount aspect 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.

Furthermore, in specific disciplines, such as, e.g., in software engineering classes, collaboration is also one of the learning objectives. In fact, software development is done by heterogeneous people arranged in 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, have a common and clear vision of the project, and be able to communicate each other seamlessly. Thus, a suited working environment is required, which includes some non-standard functionalities that may be not present in common Learning Management Systems (LMS) since they usually serve for content delivery and 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

DOI reference number: 10.18293/DMS2016-023
The remainder of the paper is organized as follows. First, in Section II, we present a summary of past experiences, which have allowed reaching the actual situation. Then, in Section III we introduce general issues about cognitive computing. In Section IV we detail on the experiments carried on with the IBM Watson cognitive computing facilities into several projects. Finally, in Section V, summary considerations conclude 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 University of Naples Federico II, IBM Italy, the Eclipse Italian community, and other Italian universities. 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 one hundred students and used the IBM Rational’s Jazz vision (see references [12] and [13] for more detailed descriptions).

B. On the Road to Eclipse

Since the ETC methodology was successful, it was applied again one year later, 2011, in a larger project called OTRE (On the Road to Eclipse) [14] for developing a set of collaborative tools, to be used within the Eclipse community. The final results were collected, published and shared, in a workshop in which the teams that had collaborated remotely finally met and were able to compare their work with other teams involved in the same project. Something like a peer-to-peer conference, that wanted to be much more than a student contest.
C. Trans-Continental Collaboration

Encouraged by the good results achieved, the next year, 2012, we faced an ambitious trial centered on cooperation of student groups belonging to both Italian and non-Italian universities [15]. The project involved two classes: a programming class and a computer science 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. 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. [16]. We are proud to mention that this project has received a special mention by IBM, as the best practice in collaboration of the year.

III. COGNITIVE COMPUTING AND WATSON

The above-cited experiences contributed to the definition of the model of smarter university given in [4], which depicts the progresses that universities are expected to do (or are already doing) for providing their students with smart services. These are implemented in remote laboratories so that they will be available 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 step ahead, 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 highlight that all of these premises are changing the programming paradigms and are driving applications to demand more cognitive services.

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 are 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:

(i) big data,
(ii) expanding human cognitive boundaries,
(iii) improving communication, and
(iv) analytics.

Each of these is briefly described in the following.

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. Expanding 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.

C. Improving 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 always-connected IoT scenario.

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 [17].

![Figure 1. Some of the Watson boilerplates](image)

These four enablers push the development of cognitive computing and relevant applications. Probably, there will be a transitional, maybe long, period in which traditional programming and cognitive programming will coexist and, sometimes, will overlap. According to this outlook, 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.

In more detail, Watson is a full-featured cognitive computing system for the research and development of cognitive systems and services [18], [19], 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 [20], which represent an important and promising field of application for the exploitation of cognitive computing services is education [21]. Some of the many available Watson boilerplates will be used within the programming course for the creation of cognos for smartphones and PCs as well. For example, the most used are: Dialog, Text to Speech, Speech to Text, Visual Recognition, Language Translation, and Personality Insights.

IV. OUTCOMES OF HANDS-ON LAB ACTIVITY

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 different projects. Among these, only a small number are finished running while others are still under development, but are nearing completion. The students involved are 150, working in 50 groups for crafting applications for both smartphones/tablets and PCs. As in previous years, the students were divided into teams and each of these was assigned a part of a bigger 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) Shulloniamoci, (iii) Cancer registry, (iv) Restart Campania, (v) Arduino car.

A. Bag of Ideas

This application resulted in the creation of a Social Network Site (SNS) for searching opportunities and sharing ideas among creative people who want to develop innovative services with some interesting features. In other words, the SNS aims to provide potential customers with the possibility of requesting assistance and expertise. More specifically, it works as follows: if you have an innovative idea but not the ability to achieve it, then you can post it on the site and ask for help (see Fig. 2). The SNS administrators make available skills and means to realize the idea. From a logical point of view, the core of the system is the analytics service that collects data, performs analyses and, finally, provides an answer on the feasibility of the idea, keeping into account the technological drivers, the territorial requirements, the existence of similar ideas, etc., applying suited analytics algorithms. Moreover, it also provides an application allowing people to manage the reputation of the participants through a peer rating system operated within the community of registered users. The users’ reputation is measured ranging from 0 to 10.

B. Shulloniamoci

The main purpose of this web application is mining contents of the most popular SNSs (e.g., Facebook and Twitter, also exploiting their known privacy flaws [22], [23]) investigating possible threatening messages, in order to recognize the profile of a bully and report it to the community. This concept is already implemented in the popular application ReThink, which keeps trace of outgoing messages. Whenever a message contains threats, the application displays the message «Are you sure you want to do that?» to discourage unfair behaviors.

![Figure 2. Screenshot taken from Bag of ideas](image1)

However, Shulloniamoci (Fig. 3) wants to be more incisive and the result of writing malicious messages is not just a warning but the exposition of the evil person to the scorn.

From a technical point of view, the application performs two steps. The former is the interaction with the user, achieved by means of the Dialog component from Watson. The latter, which is the application’s core functionality, is the elaboration of the text that may result in the unmasking of the “bully”. This phase is performed by Personality from Watson. It allows analyzing a text to the aim of extrapolating the writer’s psychological profile, showing its main characteristics, as well as what actions he may or may not accomplish. Note that, in order to achieve significant results, Personality needs sets of 3500 to 6000 words. Using few words will reflect in getting less reliable profiles. Before using Personality, it can be very useful text through the facility Concept Expansion and Concept Insights also available within the Watson libraries.

![Figure 3. Screenshot taken from Resize the bully – “Shulloniamoci”](image2)
In fact, Concept Expansion helps clarifying youth languages or slang in more clear terms (for example, it makes a connection between the words Big Apple and New York that will be considered synonymous) and such a pre-processing can be very useful to better understand the users’ intentions. Moreover, Concept Insights allows linking given words to concepts; thus, it can detect any threats, insults, or aggressive language that may not be detected by merely scanning the text.

C. Cancer Registry.

The main aim of the Cancer Registry project is to build a website and an app to show the data for the more occurring tumors in a given geographical area. The motivation behind this project is finding a relationship between such serious diseases and pollution (air, water and waste management in general), which is supposed to increase cancer presence in specific areas of the country. This huge work uses data from disparate sources, i.e., the National Cancer Registry of Italy, environment sensors monitoring air and water pollution, geographical data from the Google maps service. The development was assigned to 4 separate groups as follows:

(i) database;
(ii) data analysis;
(iii) application development (currently available only for devices running the Android OS, it will be built for iOS soon);
(iv) website development.

It is worthwhile noticing that, in this case, it was hard identifying a specific boiler plate tailored to our needs. We are considering the possibility to develop an ad-hoc component based on Watson.

Owing to the Analytics functionalities provided, it is possible analyzing data from different sources. In the chart (Fig. 4), the age of patients hospitalized for tumors in the 2003-2013 decade is depicted. From this chart we can find the age with more incidence of tumors in the selected geographical area.

D. Restart Campania.

The Restart Campania project is the development of an application that allows promoting tourism in the Italian region Campania. One of its core functionalities is visual recognition, which enables users, i.e., tourists and citizens, to adopt the places and monuments they take picture of, by making a donation or by taking part to the activities of a community devoted to promote a specific point of interest, to attract more and more visitors, thus enhancing culture, tourism and commerce. The application implementing this service includes an app for smartphone and/or tablet and a website. As users take photos of specific monuments, the system creates a map with the relevant markers. Specifically, images are recognized through the Visual Recognition tool, which looks for a match in the huge database of images it relies on. Once the place or monument in the picture is identified, the app performs a Google search to provide the user with more information. Moreover, the monument is highlighted on a Google map (see Fig. 5) where corresponding markers are linked to videos uploaded by users. (see Fig. 6).

E. Arduino Car

The Arduino Car project received the special mention by IBM during the last Eclipse-IT workshop, “Head in the Clouds”, held in Rome on October, 2015. In more detail, Arduino Car is the design and development of a real, working, toy-machine and its remote control system, running in a smartphone app. It was entirely developed within the IBM Bluemix PaaS and exploiting Watson functionalities, including specific IoT facilities. Figure 7 shows the prototype at its current state of development.
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 to the practical nature of such a discipline, we noticed that better results are achieved when students have to deal with 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, i.e., 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. 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


