A PLATFORM FOR LEARNING INTERNET OF THINGS

Zorica Bogdanović, Konstantin Simić, Miloš Milutinović, Božidar Radenković
and Marijana Despotović-Zrakić
Department for e-Business, Faculty of Organizational Sciences, University of Belgrade
Jove Ilića 154, Belgrade, Serbia

ABSTRACT

This paper presents a model for conducting Internet of Things (IoT) classes based on a web-service oriented cloud platform. The goal of the designed model is to provide university students with knowledge about IoT concepts, possibilities, and business models, and allow them to develop basic system prototypes using general-purpose micro-devices and a cloud and service infrastructure. The model was based on a cloud infrastructure deployed at the E-Business Department at the Belgrade University, and some implementation details are given. The model was tested and evaluated in a pilot course.

KEYWORDS

Internet of Things, Raspberry Pi, Arduino, Cloud Computing

1. INTRODUCTION

The expression “Internet of Things” describes the existence of a number of various things or objects like tags, sensors, actuators, mobile devices, capable of cooperating in order to achieve a common goal (Atzori et al. 2010). Such intelligent devices can take a number of forms and roles, and the composition of such systems can be adjusted dynamically, according to the needs of the users. This gives IoT an almost unlimited area of application within both business and industry (process management, intelligent transport, automation), as well as in homes and public environments (smart homes, e-health, assisted learning).

The term IoT encompasses an unbounded, growing set of devices and technologies, and as the IoT technologies gain traction globally, the need for experts that combine knowledge from various technical fields increases. IoT projects are likely to need designers, system integrators, developers and technicians in order to take an idea from inception to execution. Such diverse requirements can create an understanding gap between business-oriented individuals and their ideas, and the actual implementers that deal with realistic constraints. Ideally, an individual with an IoT business idea would be able to understand the possibilities and work in a small team, developing a prototype using off-the-shelf parts.

Introducing IoT into an environment is accomplished by introducing and interconnecting intelligent devices and, essentially, making an environment intelligent and supportive of any human activity. Applications of IoT are therefore as diverse as human activities and environments are. It is impossible to foresee the specifics of future IoT development, but some currently relevant, broad domains of application include transportation, logistics, healthcare, smart homes/offices/plants, and personal and social domains (Atzori et al. 2010).

Internet of Things represents an advanced paradigm, one that requires technology, knowledge and infrastructure, available in rich, developed countries. However, IoT solutions and especially IoT education can immensely benefit developing countries, offering a way of catching up faster, as well as a profitable industry for outsourcing, with predictions of IoT market being worth 22 to 50 billion dollars in 2020 (Schlautmann et al. 2011). Thanks to the cheap micro-devices like Raspberry Pi and Arduino, it is possible to develop various systems with less investment into infrastructure. The Raspberry Pi is of special interest, since it represents an entire computer the size of a credit card, and some systems based around it can be found in (Raihan 2013)(Kaloxylos et al. 2014). Raspberry Pi and similar devices are generally complemented by software APIs that abstract low-level operations, allowing effective utilization from a higher perspective, which is well-suited for individuals with a background in business informatics.

Traditional teaching in practical engineering areas usually has a twofold structure, where the first part presents theoretical foundations, and the second introduces real-world issues and applications. A different approach explored by some institutions gives more freedom to students, allowing them to choose the direction, breadth, and depth of their education, as well as combine their pre-knowledge from other areas of study (Director et al. 1995). These stipulations can still potentially be applied even in the context of a single IoT course, especially if it is not strictly hardware-oriented. Small classes comprised of business informatics students could produce a motivating and individual experience for every student by taking advantage of their diverse background.

Inclusion of technology into higher education is well suited for models based on constructivism and socialization, and can transform the educational process by making it more effective and attractive to students (Bustos Andreu & Nussbaum 2009). IoT classes can capitalize on this effect since they inherently deal with technological gadgets and information communication technologies. Several types of environments also attempt to produce an experimental setting combining people and technology in order to motivate innovation, development and research. Some examples of such environments and approaches are given in (Chin & Callaghan 2013), where “living labs”, “iCampus”, “smart box”, and Pervasive-interactive-programming are combined to produce a highly motivating and effective educational environment.

Depending on the educational context, several approaches to teaching IoT can be adopted. At the lowest level are individual IoT devices, and understanding them requires the knowledge of electronics and low-level microcontroller programming. The middle level is informatics-oriented, encompassing communication protocols, system integration, web services, human interfaces, etc. At the highest level are the design and business aspects of developing IoT applications. Teaching IoT comes with a set of problems for both the students and the educators, especially when concentrating on the higher levels of IoT. The main problems are (Callaghan 2012): the lack of electronic design expertise among students; the need for complex hardware and software tools; the time-consuming nature that limits complexity; and student-built hardware usually has fixed functionality and too simple to give realistic product development experience.

One approach in teaching IoT is the use of simulation tools to simulate the devices or the environment in which they are deployed. An example can be seen in (Yilmaz 2011) where the authors utilized a test card capable of processing digital and analog inputs and simulating home appliances, a model of a home, and a simple control interface with 3D models of house interior. In this approach, the teaching can be performed even without some or all of the hardware, and it allows the course to concentrate more on the software aspect of IoT. However, the simulation cannot replace the benefits of actually working with the IoT devices and potentially constrains the students’ imagination, making them think within the limits of the simulated scenario.

Working in the wide field of IoT technologies can require skills such as problem solving, team work, and leadership, as well as practical experience with actual “things” used. The active learning approach is shown to be very effective in such conditions, and a mixture of collaboration, competition and peer learning in a hands-on environment reinforces the students’ transversal skills (teamwork, communication, critical analysis) (Panadero et al. 2010).

The main aim of this paper is to improve the process of learning IoT using modern technologies and a hands-on approach. A model for conducting hands-on IoT classes with business informatics students, supported by a cloud infrastructure and web services is presented. The model relies on the existence of cheap, general-purpose programmable devices like Raspberry Pi microcomputer and the Arduino microcontroller boards. Using these devices, it is possible to easily produce slightly less efficient IoT solutions that do not require specialized components or expert knowledge to build. A pilot class was performed with a small group of interested students as a test of feasibility, and this paper, accordingly, represents a starting point for further research that will be performed on larger groups of students.

2. MODEL FOR TEACHING IOT

The generic model of platform for teaching IoT is shown in figure 1 and describes the required equipment and supporting infrastructure. The model is split into three layers – the device layer, the service layer, and the app layer. Some components used in the actual implementation at the Department for E-Business of the Faculty of Organizational Sciences at the Belgrade University are also mentioned below.

Ideally, the equipment for an IoT class should be as rich as possible, allowing the students to design a multitude of solutions. In reality, financial limitations can be severe, especially in poorer countries, and the equipment should be selected carefully. The main components at the device layer should be multifunctional, cheap microcomputers and microcontrollers, capable of taking many roles according to user programming. Such devices are readily available, the examples being the Raspberry Pi and Arduino which were used in the pilot implementation. Temperature, light, noise, and other sensors are usually simple and cheap, while expensive actuators and controllable appliances can be simulated by using diodes and simple circuits (Yilmaz 2011).

The operating logic can be distributed between devices, but an easier solution is to centralize it behind a well defined web service API. Devices that perform measurements (i.e. sensor nodes) can occasionally report to web services which take on the role of the middleware in the system. The devices can receive their instructions either by polling predetermined services, or by running their own web services for input if they are powerful enough. The web service approach is especially beneficial in an educational, cooperative environment. Students can develop their own services and share them with other students; more complex services can be built by integrating students' services with cloud-provided and external web APIs. Client applications, whether web, desktop, or mobile-based, interact only with the web services and need not concern themselves with device-specific knowledge.

The core component for IoT class implementation was the web hosting service freely provided to the students, which was implemented using the ISPConfig software allowing the deployment of student web services and web applications. Reporting of sensor readings was recognized as a common scenario in IoT applications, so a simple API was provided to the students for this purpose, allowing the storing, use, and sharing of sensor readings, as well as importing or generating external data for use in simulations of some specific conditions. SMS sending/receiving service was also provided to the students, and other services could be provided depending on the existing infrastructure of the educational institution. All services that need user accounts for their operation can, for improved interoperability, rely on the centralized store of student credentials. In the actual implementation, the OpenLDAP directory used by the learning management system of the Department was used for this purpose.

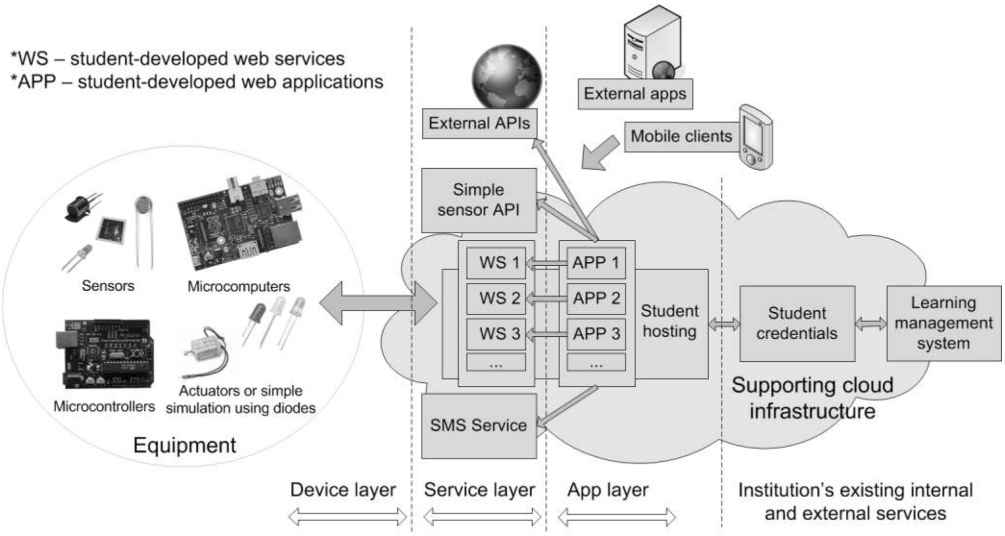

Figure 1. Model of infrastructure layers

The implementation of the presented model can potentially be very complex, with a large number of interconnected services, some of them possibly integrated with existing learning systems in order to allow easier tracking and grading of student activity. Although not necessary, an optimal solution in such conditions is to utilize a cloud platform, providing high scalability and redundancy, with better utilization of available processing resources (Beaty 2013; Sultan 2010). The suitability of utilizing cloud infrastructure in higher education is presented in a number of research papers (Despotović-Zrakić et al. 2013; Ercan 2010). The pilot class relied on the existing, OpenStack based cloud infrastructure used by the E-Business Department.

3. COURSE STRUCTURE

The goal of the IoT pilot course is to introduce and educate students with a background in business informatics in using the hardware, operating systems, software, and tools for automation of smart environments. The course consists out of four units (figure 2):

- Introduction to technologies used,
- Defining scenarios,
- Developing web services,
- Developing web and mobile applications

The students are first introduced to all of the elements of systems used for smart environment automation. This introductory part includes a review of hardware components, scenarios of their use, business models for their application, and successful existing systems.

The course is realized mainly through laboratory exercises. The students should be divided into teams, and the exercises done in the form of workshops. During the semester, students should receive a specific task for every exercise. These tasks describe the context of application and the method for exercise completion. Every team should get a number of user requests for development of an intelligent environment. Based on these requests, the students need to envision a scenario and design two schematic representations showing hardware components that would satisfy these requirements. The first scheme should give a technical presentation of sensors and actuators and their connections to microcomputers and microcontrollers. The students can, for instance, use a free, open-source application called Fritzing for this purpose. The other scheme is to present the entire intelligent environment with the locations and interactions of larger units comprising the solution, and the students can create it using any standard software for drawing diagrams.

The implementation of the hardware environment should be based on the previously designed scheme for interconnecting sensors and other devices. For this task, the students are to be provided with devices like Arduino microcontrollers, Raspberry Pi microcomputers, and various sensors, actuators, and other components. In order to make it easier for the students to review their knowledge at home, without actual equipment, a set of video clips showing different aspects of handling the hardware and other tools can be provided to them.

After the implementation of the hardware infrastructure, the students should design and implement a software system to complement their previous solution. Arduino microcontroller uses a modified version of C++ programming language, while the Raspberry Pi platform best supports the Python programming language, and the students should utilize any previous knowledge available and work on those parts they are most familiar with. The students can utilize Xively in their projects, the public cloud service for collecting and accessing data from various sensors, or develop similar solutions smaller in scope.

As previously mentioned, the exercises are performed in the form of workshops, where every team receives a specific context and a task to implement a scenario for use of smart devices in said context. The examples of some of the main contexts/tasks include the following:

- Smart home context - Arduino or Raspberry Pi are to be used in combination with temperature sensor. If the temperature exceeds a certain value, an SMS message should automatically be sent to the owner of the smart home. Using a web or mobile application, the owner has a continuous insight into the measured temperature.
- Smart classroom context - Arduino or Raspberry Pi are to be used in combination with an NFC tag reader and a digital display. During the entry into the classroom, every student should tap his NFC card on the reader, which will register his presence in the current lecture. Since the capacity of the

classroom is limited, if all seats are taken, the display outside of the classroom should show an appropriate notice. The students can use a web or a mobile application to check the number of free seats in the classroom.

- Smart library context - for smart library automation, the students should utilize noise sensors and a speaker in combination with the standard Arduino or Raspberry Pi devices. If the noise sensors record an amount of noise above a certain limit, a voice message should be played using the speakers in the library. The administrators should be able to check noise levels using a web or mobile application.

The final outcome of this course should be the integration of all laboratory exercises into a single project. The final grade is composed out of grades for tasks completed in exercises, the grade for an electronic test, and a grade for completion of the project, with the tasks and the project making up to 80% of the grade. The project should define user requirements, scenarios, project documentation, web services, and a web or mobile application for automation of the smart environment.

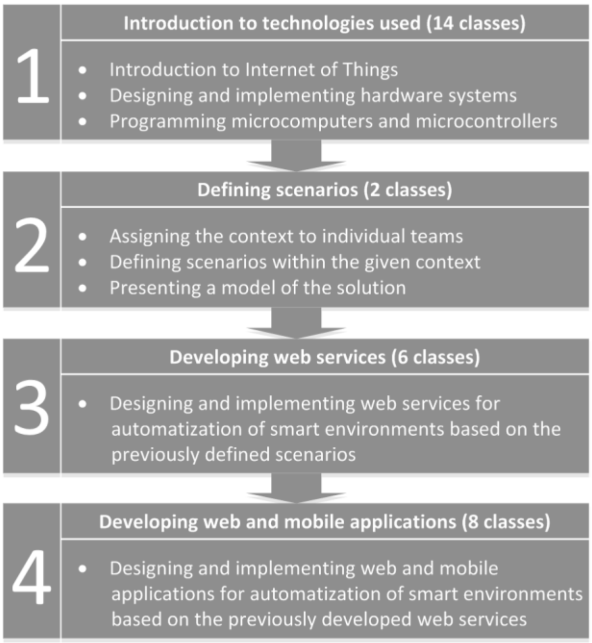

Figure 2. IoT course structure

4. PILOT CLASS IMPLEMENTATION AND RESULTS

The pilot course was implemented at the Laboratory for E-Business according to the course structure outlined in the previous chapter, with 8 students of master studies taking it as an elective class. All of the students were between 20 and 25 years of age, all had similar backgrounds in business informatics, with only slight variations in their area of specialization and interest, and that half of the students were employed in some form (part-time or full). The pilot course was shorter in length compared to the previously described structure and consisted out of 12 classes in total, four covering the introduction and remaining eight the other, practical sections.

After the completion of the entire course, the students were presented with a questionnaire with questions grouped into five sections: demographic data, opinions about course structure and execution, opinions about workshop content, student's motivation, and student's perception of own knowledge. Students were also graded according to their completed tasks, project, and test, and out of eight students participating, three had the final grade 8, four had the grade 9, and one had the maximum grade of 10, for an average of 8.75.

The questions from the first group concerning the course structure and execution were to ascertain whether the course structure was well balanced, if the tasks were too difficult and the allotted time sufficient, and what was the students' opinion of the interaction model between the students and teachers, as well as the interaction between students themselves. Here, 62.5% students thought that the optimal length of a single class within the course was three hours, while the remaining students approved the actual length used (1.5 hours). All students, however, agreed they had sufficient time to complete the tasks they were assigned. The tasks were mostly marked as being of medium difficulty (87.5%), but most of the students (75%) thought that the assistant's help was needed, and 25% thought it was necessary. All of the students agreed that the competitiveness between the groups was a motivating factor.

The third part of the questionnaire was concerned with the actual content of workshops and was to ascertain as to which parts of the workshop were best liked by the students, which were considered the most difficult, and which were seen as the most important. These questions were to provide insight into the balance of the course. Accordingly, the activity of thinking up IoT scenarios was mostly ranked as average (62%), with all other answers going above that. The students also thought this activity was of average interestingness (50%), and that the proper understanding of the scenarios was very important for completing the given tasks. Most of the students didn't have any preknowledge about IoT, but didn't think that dealing with the hardware was too difficult. This part of the tasks was rated mainly as very interesting (50%), and a similar opinion was received concerning the importance of these technologies in the wider task. The students had, on average, best preknowledge in the domain of web services, since this was a part of their earlier education (50% marking it as average), but most also thought that dealing with services was of medium and high difficulty. This can be explained by the fact that it was expected for the students to have a certain preknowledge in this area, and the requirements were accordingly stricter. The students also thought that web services were a crucial component of their solutions (25% - important, 75% - very important). The main sentiment was that the Android development was not too difficult, and that this activity was very interesting (50%). Some of these results are given in Table 1.

Table 1. Questions concerning the perceived interestingness and difficulty of dealing with IoT hardware and web services

Question	Answer	# responses	Question	Answer	# responses
How interesting was dealing with software and hardware components?	Very boring	0	How difficult was dealing with software and hardware components?	Very easy	2
	Boring	1		Easy	2
	Average	2		Average	2
	Interesting	4		Difficult	2
	Very interesting	1		Very Difficult	0
How interesting was dealing with web services?	Very boring	1	How difficult was dealing with web services?	Very easy	1
	Boring	1		Easy	0
	Average	1		Average	2
	Interesting	2		Difficult	3
	Very interesting	3		Very Difficult	2

The fourth part of the questionnaire was short, and contained questions about how the students were motivated to continue improving their knowledge of the technologies covered during the class. 75% of students were motivated to do so, with 12.5% motivated partially, and remaining 12.5% not motivated.

The final part was concerned with the students preknowledge and about how they perceived their improvement in certain areas. Students did not have much knowledge about IoT beforehand, but had an average level of knowledge with programming and working with web services. The students mainly had only an elementary level of knowledge about Android OS, with only some of them having completed an elective course in Android programming. The students thought that the provided educational materials were sufficient (75%), and that they have gained a satisfactory amount of knowledge about IoT and other applied technologies.

Additionally, the students were asked to give their opinions about the class and about possible improvements. All of the opinions were very positive, and some of them are given below:

- "Intriguing, interactive, and fun. Motivating."
- "It's very interesting to see the practical results of programming"
- "It allowed me to review important things"
- "There should be more classes like this. Working on a concrete example, whose state can be modified through PHP or Android was a top-class experience"

- “No objections. The workshop was interesting and motivating. Also, to see the results improves the dry programming experience a lot.

5. CONCLUSION

The presented model has several advantages - it is cheap, effective, scalable, and well-suited for students not coming from a hardware background. The model requires some supporting infrastructure and establishing a number of software services, but all well within the capabilities of any business informatics or software oriented curriculum. The students taking part in the pilot class were well-motivated, but the short class length and small number of participants did not allow us to make any general conclusions, and further research will be performed in future classes.

The main deficiency of the pilot class was the fact that only eight students participated. Still, there is currently not many research papers concerning the course structure and execution of IoT classes, and some general remarks, possibly useful for future research and larger IoT classes can be given based on the obtained results and experience. The students rated the help from the assistants as very important for the completion of their tasks; this especially needs to be considered if future classes are performed with a larger number of students. IoT exercises that attempt to encompass both the hardware and software aspects of IoT should likely be of longer length, as the students were mostly in favor of 3 hour classes over those twice as short. The SMS service that was provided to the students for sending and receiving SMS messages was very well liked, although it was not a core part of the IoT classes. This is probably the case since the SMS messages are a very familiar concept, and it allowed the students to envision IoT integration in a more realistic manner. The educational institution should therefore attempt to provide the students with as many other services as possible in order to widen the possibilities and increase student motivation. Students also felt that seeing the results of programming immediately was motivating, and IoT exercises should be designed in such a way to provide directly verifiable results, and preferably a direct impact on the physical world in some way.

ACKNOWLEDGEMENT

The authors are thankful to Ministry of Education, Science and Technological Development of Republic of Serbia for financial support grant number 174031.

REFERENCES

Atzori, L., Iera, A. & Morabito, G., 2010. The Internet of Things: A survey. *Computer Networks*, Vol. 54, No. 15, pp.2787–2805.

Beaty, D.L., 2013. Cloud Computing 101. *ASHRAE JOURNAL*, Vol. 55, No. 10, pp.88–93.

Bustos Andreu, H. & Nussbaum, M., 2009. An experimental study of the inclusion of technology in higher education. *Computer Applications in Engineering Education*, Vol. 17, No. 1, pp. 100–107.

Callaghan, V., 2012. Buzz-Boarding; Practical Support for Teaching Computing Based on the Internet-of-Things. In *1st Annual Conference on the Aiming for Excellence in STEM Learning and Teaching*. London, United Kingdom, pp. 1–5.

Chin, J. & Callaghan, V., 2013. Educational Living Labs: A Novel Internet-of-Things Based Approach to Teaching and Research. In *9th International Conference on Intelligent Environments*. IEEE, pp. 92–99.

Despotović-Zrakić, M. et al., 2013. Scaffolding Environment for e -Learning through Cloud Computing. *Educational Technology & Society*, Vol. 16, No. 3, pp. 301–314.

Director, S.W. et al., 1995. Reengineering the curriculum: design and analysis of a new undergraduate Electrical and Computer Engineering degree at Carnegie Mellon University. *Proceedings of the IEEE*, Vol. 83, No. 9, pp. 1246–1269.

Ercan, T., 2010. Effective use of cloud computing in educational institutions. *Procedia - Social and Behavioral Sciences*, Vol. 2, No. 2, pp.938–942.

Kaloxylos, A. et al., 2014. A cloud-based Farm Management System: Architecture and implementation. *Computers and Electronics in Agriculture*, 100, pp. 168–179.

Panadero, C.F., Roman, J.V. & Kloos, C.D., 2010. Impact of learning experiences using LEGO Mindstorms® in engineering courses. In *IEEE EDUCON 2010 Conference*. Madrid: IEEE, pp. 503–512.

Raihan, K., 2013. Raspberry Pi Image Processing based Economical Automated Toll System. *Global Journal of Researches in Engineering Electrical and Electronics Engineering*, Vol. 13. No. 13.

Schlautmann, A. et al., 2011. Wanted: Smart market-makers for the "Internet of Things." Available at: http://www.adlittle.se/prism_se.html?&view=383 [Accessed April 28, 2014].

Sultan, N., 2010. Cloud computing for education: A new dawn? *International Journal of Information Management*, Vol. 30, No. 2, pp.109–116.

Yilmaz, E.N., 2011. Education set design for smart home applications. *Computer Applications in Engineering Education*, Vol. 19, No. 4, pp. 631–638.