IT in Industry, vol. 2, no. 3, 2014

Design of an IT Capstone Subject - Cloud Robotics

Kieren Lim, Jayantha Katupitiya
School of Mechanical & Manufacturing Engineering
The University of New South Wales
Sydney, Australia
j.katupitiya@unsw.edu.au

Ka Ching Chan, Mary Martin
Department of Computer Science & Computer Engineering
La Trobe University
Bendigo, Australia
ka.chan@latrobe.edu.au, m.martin@latrobe.edu.au

***Abstract*—This paper describes the curriculum of the three year IT undergraduate program at La Trobe University, and the faculty requirements in designing a capstone subject, followed by the ACM's recommended IT curriculum covering the five pillars of the IT discipline. Cloud robotics, a broad multidisciplinary research area, requiring expertise in all five pillars with mechatronics, is an ideal candidate to offer capstone experiences to IT students. Therefore, in this paper, we propose a long term master project in developing a cloud robotics testbed, with many capstone sub-projects spanning across the five IT pillars, to meet the objectives of capstone experience. This paper also describes the design and implementation of the testbed, and proposes potential capstone projects for students with different interests.**

Keywords—cloud robotics; networked robotics; Internet of Things; capstone projects; IT education

I. INTRODUCTION

A. Capstone Subject

As defined by the FSTE (Faculty of Science, Technology, and Engineering) at La Trobe University, a capstone is a culminating subject within the final year of a discipline course. A capstone subject provides opportunities for students to synthesis and demonstrate discipline knowledge acquisition of graduate capabilities, employment skills and areas of learning acquired during their undergraduate degree [1]. At La Trobe University, Information Technology is a three year undergraduate degree with a strong focus on work integrated learning [2]. Students are required to complete at least one capstone subject in the final year. The design of a capstone subject is the responsibility of the capstone subject coordinator; guidelines for this are provided by the faculty. Most of the design guidelines are assessment related. The design requirements unrelated to assessments are given as follows.

- a capstone subject must be designed to be authentic within the discipline, often through the use of real-world problem solving; and
- integrate relevant subject elements especially where these are delivered in different subjects.

In the next section we will discuss the IT curriculum and explain how cloud robotics as final year projects meets the above design requirements well.

B. IT Curriculum

The IT curriculum at La Trobe University is ACS (Australian Computer Society) accredited, and follows the latest 2008 curriculum guidelines for undergraduate degree programs in IT, recommended by the ACM (Association for Computing Machinery) [3]. The ACM report was an extensive undertaking, with direct contribution by over thirty people and widely reviewed by academics and practitioners. According to this report, the academic discipline of IT can well be characterized as the most integrative of the computing disciplines. The depth of IT lies in its breadth: an IT graduate needs to be broad enough to recognize any computing need and know something about possible solutions. The IT graduate would be the one to select, create or assist to create, apply, integrate, and administer the solution within the application context [3]. This report also suggested the five pillars of the IT discipline:

1. programming
2. networking
3. human-computer interaction
4. databases
5. web systems

The five pillars should be built on a foundation of knowledge of the fundamentals of IT. Overarching the entire foundation and pillars are information assurance and security, and professionalism.

Fig. 1 depicts the IT curriculum at La Trobe University. It consists of four fundamental IT subjects, two or more subjects in each of the five pillars, one subject in Information and IT security, one subject in Professional Development, electives, industry based learning, minor and major projects. The final year one semester project has been chosen as the capstone subject for the degree.

Cloud robotics, coined by James Kuffner of Google in 2010 [4], aims to remove conventional limitations and introduce greater capabilities through applying the benefits of cloud technology to robotics and automation. Any application of cloud robotics requires in-depth understanding and integration of all five pillars of the IT discipline with mechatronics. IT's broad nature makes cloud robotics an ideal candidate to become an IT capstone. This paper proposes to develop a cloud robotic testbed as a long term master project, with many sub-

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

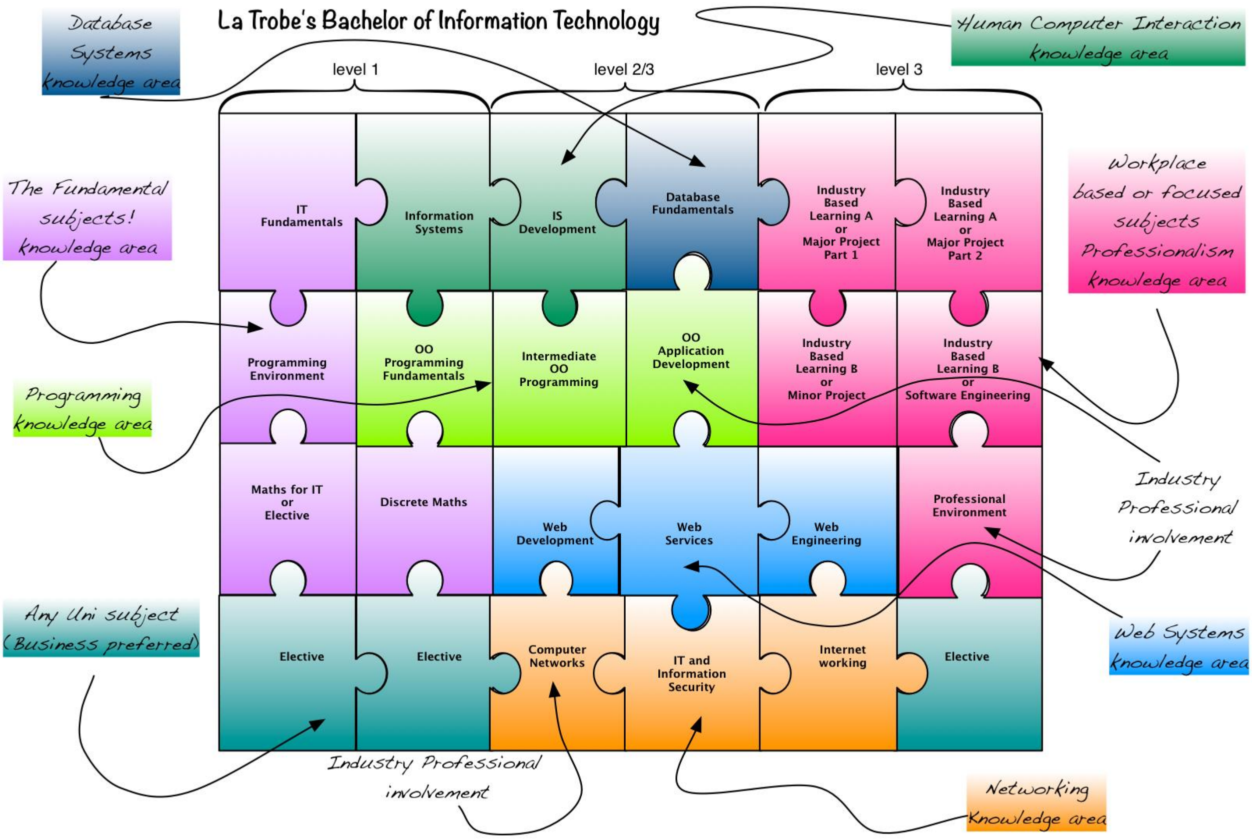

Fig. 1. Curriculum of La Trobe University's Bachelor of Information Technology.

projects offered as capstone projects to final year IT students. Through the capstone experiences, students will develop expertise in their areas of interests. At the same time, working on a project as part of a larger project with a common theme will help students understand the bigger pictures, as well as the connections of the pillars, and current and emerging IT technologies.

The cloud testbed has been constructed for both teaching and research purposes. Its requirements:

- Low cost: Components should be open-source, off-the-shelf, and inexpensive.
- Flexible: Various types of hardware and software should be easily integrated into the testbed.
- Adaptable: The testbed should be able to accommodate various networking methods and protocols.
- Accessible: The testbed should be comprised of relatively common components so as not to complicate any results produced.

Open source software, off-the-shelf electronic and mechanical components, DIY robotic kits, open source system-on-a-chip (SoC), low cost wireless communications components such as Wi-Fi and ZigBee, and low cost mobile devices, have gradually become commodities in the marketplace and readily available through online shops around the world. This continuous downward trend on pricing has made the development of such a cloud robotics testbed affordable and attractive from an educator's perspective.

II. CLOUD ROBOTICS

Traditional robotics was characterised by computerised systems of sensors and actuators performing various functions, facilitated by onboard computation, programming and data storage. These robots usually have limited capabilities, operate as standalone machines, and lack the ability to network with other devices. As such, a solution was proposed in the form of networked robots. The IEEE Robotics and Automation Society defined "networked robot" as a robotic device connected to a communications network such as the Internet or LAN [5]. With this extended connectivity, the functionality of robots extended to tasks no single robot could accomplish.

Networked robots have significant advantages as any single robot has access to the computational power and data storage of the entire network of robots resulting in improved task efficiency. A major advantage is that sensor data of every device in the network can be shared [6]. Networked robots are likened to animals working in groups, enabling behaviours far more complex and intelligent than is possible by any individual.

The modern trend of cloud computing adds another dimension to networked robotics. Cloud computing is a term used to refer to the use of modern networking resources to develop dynamic computing solutions by reducing limitations in areas such as data storage and computational power. This is accomplished through interaction with a significant number of computers and servers, often represented by virtual hardware, which introduces benefits such as reduced costs in electricity and network bandwidth [7]. Enabling robots access to these cloud computing resources led to the development of cloud robotics. With cloud computing providing computing resources on demand, robots would have the option of offloading computationally intensive tasks to these resources.

Robots with access to the cloud no longer require expensive processors for heavy computations as they can be offloaded. Operations like image processing, voice recognition, 3D mapping or other uncertainty-plagued tasks can use parallel processing in the cloud and the results returned to the robot. In addition, tasks such as motion planning [8] and control [9] can also be performed in the cloud. All these strategies result in lower power consumption and lighter hardware requirements.

Goldberg and Kehoe [10] identify further potentials of cloud robotics:

- Access to global libraries of images, maps and object data
- Robot sharing of outcomes, trajectories and dynamic control policies
- On-demand human guidance

Going beyond relational databases, through the cloud a robot has access to ever-growing datasets of images, maps and object data, also known as “Big data” [11]. With access to such extensive data, a variety of applications are possible. One example is the use of cloud datasets to facilitate robot grasping [12], a task which requires extensive data on objects, their features and how to manipulate them. Since a robot could be asked to grasp any object, access to the cloud gives the robot access to constantly updated object and grasp data. This operation is further enabled by the parallel processing capability of cloud computing to evaluate algorithms and determine optimal grasps. In particular, parallel processing is advantageous in situations with shape uncertainty.

As robots or other devices and systems perform operations, their results become accessible to other devices through the cloud. This opens the possibility of learning new functions and capabilities based on successfully performed operations by other robots. Extending the grasping example, a successful grasp on a particular object performed by another robot can be downloaded and performed in the same context removing uncertainty and the need to employ intensive computations. Path planning is another suitable candidate as robots can use paths previously determined as successful. This concept of shared capabilities and robot learning provides a significant advancement in robot development.

Goldberg and Kehoe [10] suggest a crowdsourcing situation where, in particular situations, robots seek the assistance of humans. This enables human intervention on tasks that may require human intuition or complex decision-making. This is particularly relevant to dynamic environments in which robots often encounter difficulties. Questions arise about which events require human guidance; however, it is evident that this ability is a desirable consequence of cloud robotics.

Furthermore, crowdsourcing enhances human-robot interaction, an area of increasing relevance as interest in service robots and the Internet of Things (IoT) [13] continues to grow. Rusu et al. [14] developed a service robot system, based on ubiquitous computing, designed to operate in a kitchen and perform common human tasks. This system is characterised by distributed and embedded computing devices throughout the environment, such as Radio Frequency Identification (RFID) tags, which allow the robot to identify objects.

The desirability of affordable robotic devices with consistently supported software, in addition to the current trend of enabling most electronic consumer products with network connectivity, will result in a portion of the IoT consisting of robots [15]. Coordinated interaction between IoT and cloud robots could include utilization of data from environmental sensors to achieve the localization of a robot with no onboard sensors [16]. IoT architecture and Machine to Machine (M2M) communication will enable cloud robotics to perform many tasks with minimal human interaction [17]. Consider a scenario in which a cloud robot identifies a faulty component, which is then automatically printed by a nearby 3D printer and finally installed by the human user in a similar fashion to Swedish furniture.

It has become apparent that a number of current and emerging technologies, such as big data, IoT, M2M, wireless sensor networks, mesh networking, mobile computing, and cloud computing, etc are converging at a fast pace, enabling. innovative applications in multiple industries, such as health, manufacturing, farming and agriculture. It is also not difficult to imagine one day a human-robot social network making new things and doing fun things together.

Two state-of-the-art cloud robotics developments worth mentioning are Willow Garage and Google cooperating on porting Willow Garage’s Robot Operating System (ROS) to Android devices enabling the development of robots that could interact with the cloud [18] and RoboEarth.org creating a database of reusable skills thus facilitating one of the key concepts of cloud robotics [19, 20].

III. CLOUD ROBOTIC TESTBED DESIGN

Cloud robotic systems are complex entities requiring multiple elements and layers for operation. The client-server system architecture, illustrated in Fig. 2, indicates the cloud is the central component through which communication and access requests are made. Furthermore, it must be capable of extensive computations and hold data to be shared amongst robots and users. Design and implementation of a highly-available cloud platform, using open source software such as OpenStack [21] or Proxmox [22], to support our cloud robotics research are ideal capstone project topics.

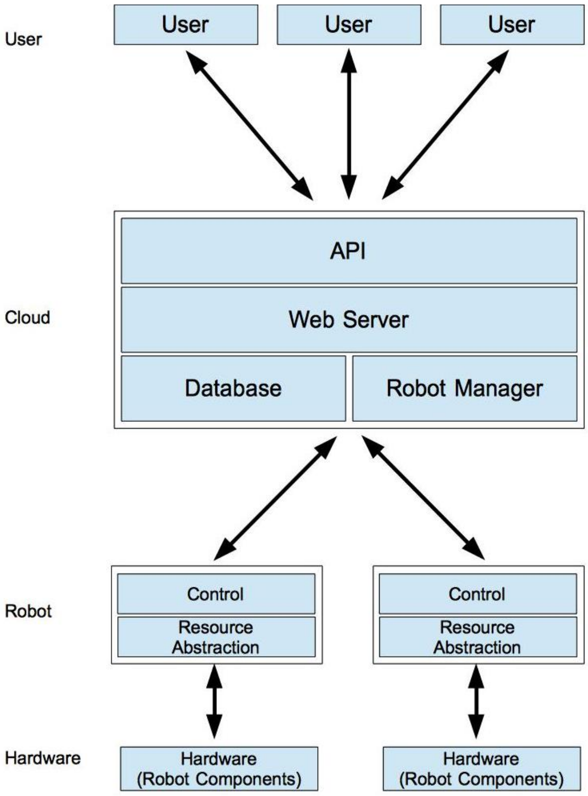

Fig. 2. Client-server cloud robot architecture.

An open source Apache web server satisfies the requirement of accessibility, providing avenues for communication such as an API or web service. Furthermore, it is accessible by any device with Internet capabilities, extending access beyond PCs or robots to other embedded devices such as smartphones. Design and building additional web services and applications for increased cloud functionality are also ideal capstone projects. For the testbed to be operated on common platforms and simplified for users, MySQL [23] and Structured Query Language (SQL) were selected.

To allow multiple robotic platforms to perform within the testbed, Chen and Bai [24], Johnson et al. [25], and Kato et al. [26] demonstrated intermediate platforms within the cloud to track and manage all the robots in operation. The robot manager performs this function. It provides task management capabilities if necessary, such as coordinating multiple robots for a single service, however, in simple operation it acts as the interface between the cloud and robotic devices.

In this model, the cloud is a server to which robot clients must connect. A resource-based architecture was ideal to abide by Service-Oriented Architecture (SOA) design. This meant robot clients must be abstracted into resources accessible by the server. However, the question arose as to the level of abstraction necessary for this project. For example, a robot could simply be presented as a device with the capabilities of proximity detection and motion, or it could be presented as multiple sensors and motors, each with its own functions. Alternatively, if grouping of sensors is desired then it may occur at different levels such as grouping all infrared sensors together and all ultrasonic sensors together or grouping sensors according to their location on a device. It was decided that, in order to enable this architecture to operate with differing hardware platforms, resource abstraction should occur on the robot before notifying the server. Presenting each robot as a single resource, while significantly reducing complexity in the system, would limit functionality and flexibility as users could not take advantage of individual sensors or robot capabilities. As a result, in this project, each individual robotic component is considered a resource.

In a similar manner to resource abstraction, control operations may differ on different devices due to various hardware configurations. It was decided that control would be performed on the robot itself.

The front-end of the system enables users to access the testbed. For experimentation, user interaction is necessary as opposed to simply viewing the system. A user-facing API was desirable as provides an interface for researchers to build applications upon the resources within the testbed. This API was to be built upon a web service framework.

A. *Robot Setup*

Initially there are two robots, of varying configuration and different levels of capabilities, in the testbed, but these will grow. The first robot built consists of multiple sensors to identify its surrounding environment. These components are placed in a configuration upon the chassis, consisting of motors for robot motion. The array of sensors selected was to demonstrate both operation of a multi-sensor robot and to show integration of multiple types of sensors into this system. As indicated in Fig. 3, three types of sensors were chosen:

1. Inertial Measurement Unit (IMU): IMUs are available with various components such as an accelerometer, gyroscope and magnetometer. This is useful for determining the location and orientation of a robotic platform.
2. Infrared (IR): These proximity sensors emit a beam of light which reflects off any object in its line of sight to a receiver. IR sensors were placed at the front-left and front-right of the robot.
3. Ultrasonic: Another type of proximity sensors that emit an ultrasonic pulse which bounces off objects and returns to a receiver. An ultrasonic sensor was placed centrally at the front of the robot.

The second robot has no proximity sensors but, other than motors for motion, only has an IMU to identify its location and orientation. The inability to identify its environment through interpretation of sensor data causes it to be completely reliant on information received from the cloud. In this system, robots of this kind may function, provided there is another source relaying data of the surroundings, such as another robot with proximity sensors or an online map.

Both robots followed the same basic design. The first step was building the chassis followed by installing the Raspbian operating system [27] on the Raspberry Pi [28]. Subsequently, the GertDuino expansion board [29] was initialised and

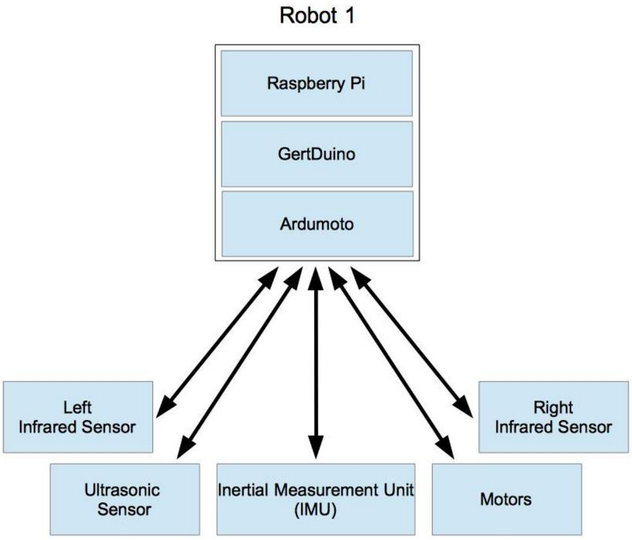

Fig. 3. Robot configuration

interfaced with the Raspberry Pi through serial communication. Then the Ardumoto motor shield [30] was connected and control of the motors was established. Finally, the sensors were configured and calibrated to provide accurate measurements.

As IT capstone projects, mechanical design of robots is not the main focus. Project students are given the flexibility to design and buid their own robots, or to hack and modify those remote control toy cars, trucks, or tanks. The testbed will consist of an army of different types of robots, which may have similar or different designs and functionalities, working together to achieve certain purposes. In addition to the two basic robots, as shown in Fig. 4, one student has built a two-wheeled self-balancing robot; and another has hacked a low cost remote control toy racing car. These robots are very different in design and require different control algorithms. However, similar to the two basic robots, they are also controlled by a Raspberry Pi and Arduino. We have found that hacking low cost remote control cars is an attractive option for students. There are many toy cars with creative and interesting designs available in toy shops or supermarkets. Students enjoy the work involved in hacking and controlling them with their own systems.

B. *I/O Expansion*

The GertDuino board is an expansion board to interface between the Raspberry Pi and Arduino Uno compatible shields. It contains two microcontrollers: the Atmega328, the same microcontroller as the Arduino Uno, and an Atmega48.

This expansion board increases the current capabilities of the Raspberry Pi as it provides access to an Analog-to-Digital Converter (ADC), for 6 analog inputs, and 14 digital outputs, of which six can be used for pulse width modulation (PWM) output. Additionally, the GertDuino enables interfacing with established Arduino hardware and software and the extensive Arduino community.

Using a custom version of AVRDUDE provided by Gordon Henderson, the Atmega328 microcontroller was configured to run at 16MHz to enable compatibility with the Arduino Uno [30]. With initial setup complete, it was possible to program the Atmega328 through the Arduino IDE on the Raspberry Pi. However, the program would run independently of the Raspberry Pi.

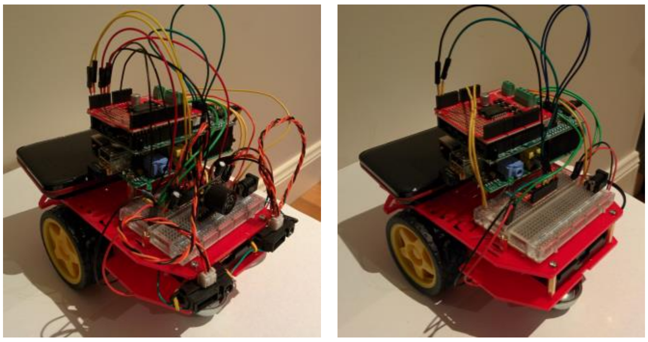

Fig. 4. Robot 1 and robot 2.

To take advantage of the capabilities of the Raspberry Pi and any connected Arduino shields, communication is required between the devices. Through the GertDuino, a serial connection between the Raspberry Pi and Atmega328 was established [28, 32, 33]. The Atmega328 was programmed using the Arduino Serial library and the pySerial package for Python 2.7. With the motors and sensors configured, it was possible for the Raspberry Pi to control the motors and receive sensor data.

C. *Networking and Communications*

To initiate wireless communication, Universal Serial Bus (USB) Wi-Fi dongles were used for both robots and individual static IP addresses configured for each robot. This provided LAN and Internet access to the robots, and allowed for a virtual network computing (VNC) server to be set up on each Raspberry Pi. The VNC server allows a PC wireless access to the robots and removes the need for peripherals, such as a monitor or keyboard. Changes in software and control can be performed remotely, significantly easing testing and operations. TightVNC software was used to set up both the VNC servers on the robots and the client on the PC [34].

With Wi-Fi setup on each robot, the testbed allows students interested in networking to experiment with MANET (Mobile Ad Hoc Network) and VANET (Vehicular Ad Hoc Network), and wireless mesh network routing protocols such as OLSR, Batman, and Babel. Other wireless communications technologies, such as IPv6 over Low power Wireless Personal Area Networks (6LowPan) and the IEEE 802.15 ZigBee networks, also have roles to play in Cloud Robotics. Other potential capstone projects include design, implement, and evaluate Voice over IP (VoIP) and video streaming technologies over different network environments. An example of such applications can be found in [35]. Students will develop their own communication systems using the same core VoIP technology, but with their own creative design of graphical user interfaces for remote control using a web browser and/or an Android tablet or phone.

D. Cloud Infrastructure

To support the long term development of the testbed, the open source cloud platform OpenStack, was specifically designed as part of the whole project. The whole platform has been setup in La Trobe University's Internetworking lab [36]. The cloud computing platform leverages the power of virtualization and redundancy to provide highly available computing infrastructure to multiple robots, and host all the required backend systems and services. The platform also provides varieties of robots an ideal, centralized data storage platform for communication and cooperative control.

The infrastructure includes a number of Cisco 2801 routers, Cisco 2960 switches, Cisco Aironet access points, Cisco Catalyst 3750 series wireless LAN controllers, purpose built Intel i-7 servers with a minimum of 24GB of RAMs, and QNAP TS-410 1U Networked Storage Servers (NAS). Networking Students are involved in aspects of the design, setup, and evaluation of high available network and cloud infrastructures to support the testbed.

E. Programming and Control

Within the proposed architecture, there are two key interfaces where connectivity and communication are required: the cloud-user interface and robot-cloud interface. A client-server model was selected for interaction throughout the architecture with the cloud acting as centralised servers and the robots being clients. At the other interface, the users would act as clients to the cloud server. This forms the basis of communication as, for this testbed operation, a connection with the cloud is essential for each device and is the most suitable with a limited number of devices. For further development, with a greater number of robots, a peer-to-peer topology could be established with modest effort. In a similar scenario, a publish-subscribe communication model could be presented. The HTTP protocol is used for data and message transfer in API and web application development, largely due to its relative lack of restrictions [37]. In API development, there are two widely used frameworks for using HTTP, REST and SOAP [38]. REST is able to communicate through multiple data formats, while SOAP uses a standard format in XML. In a comparison of JSON and XML, JSON was found to be ideal for this type of project as it is populated with resource limited devices [39]. Hence JSON was used for data transfer throughout this project; and consequently the REST framework was implemented for an API between the cloud and users. In addition, it allowed a web service to be stateless meaning client requests had to be explicit and URIs had to provide a comprehensive method of resource identification. Using similar technology, the Robot-Cloud interface was designed as a web service on the server side to provide connections to the robot clients. For programming students abundant capstone opportunities are available in areas such as API development, mobile app development, web services and cloud based control algorithms.

IV. CONCLUSION

This paper described the curriculum of the three year IT undergraduate program at La Trobe University, and the faculty's requirements in designing a capstone subject. As the broadest and most integrative of the computing disciplines, ACM recommended the five pillars of the IT discipline. We have proposed to develop a cloud robotics testbed, as a master project, with many capstone sub-projects spanning across the five pillars. We have presented the initial design and implementation of the testbed, which will continue to offer abundant opportunities to students to gain valuable capstone experiences.

ACKNOWLEDGEMENT

The authors would like to thank La Trobe University for the financial support through a special research grant scheme to aid regional research. This paper is an extended version of a paper previously presented at the 9th International Conference on Information Technology and Applications (ICITA), in Sydney, 1-4 July 2014.

REFERENCES

[1] E. Johnson, "Capstone subjects in FSTE," unpublished, Faculty of Science, Technology and Engineering, La Trobe University, 2013.

[2] L. Staehr, M. Martin, and K. Chan, "A multi-pronged approach to work integrated learning for IT students," *J. Information Technology Education: Innovations in Practice*, vol. 13, pp. 1-11, 2014.

[3] B. M. Lunt, J. J. Ekstrom, S. Gorka, G. Hislop, R. Kamali, E. Lawson, R. LeBlanc, J. Miller, and H. Reichgelt, "Information Technology, 2008, Curriculum guidelines for undergraduate degree programs in Information Technology," Association for Computing Machinery (ACM) and IEEE Computer Society, 2008. [Online]. Available: http://www.acm.org/education/curricula/IT2008%20Curriculum.pdf

[4] J. Kuffner, "Cloud-Enabled Robots," in *IEEE-RAS Int. Conf. Humanoid Robots*, Nashville, TN, 2010.

[5] IEEE RAS Technical Committee. (2012, September). *IEEE Society of Robotics and Automation's Technical Committee on: Networked Robots* [Online]. Available: http://www-users.cs.umn.edu/~isler/tc/

[6] V. Kumar, G. Bekey, and A. Sanderson, "Networked Robots," in *Assessment of International Research and Development in Robotics*, NASA, 2006, pp. 73-80.

[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, *et al.*, "A view of cloud computing," *Communications of the ACM,* vol. 53, pp. 50-58, 2010.

[8] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA'04.* 2004, pp. 4344-4349.

[9] A. Barili, M. Ceresa, and C. Parisi, "Energy-saving motion control for an autonomous mobile robot," in *Proc. IEEE Int. Symp. Industrial Electronics, ISIE'95.* 1995, pp. 674-676.

[10] K. Goldberg and B. Kehoe, "Cloud robotics and automation: A survey of related work," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-5,* 2013.

[11] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, Angela Hung Byers. (2011, May). *Big data: The next frontier for innovation, competition, and productivity* [Online]. Available: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation

[12] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *IEEE Int. Conf. on Robotics and Automation*, 2013.

[13] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks,* vol. 54, pp. 2787-2805, 2010.

[14] R. B. Rusu, B. Gerkey, and M. Beetz, "Robots in the kitchen: Exploiting ubiquitous sensing and actuation," *Robotics and Autonomous Systems,* vol. 56, pp. 844-856, 2008.

[15] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *Network, IEEE,* vol. 26, pp. 21-28, 2012.

[16] L. Wang, M. Liu, M. Q. H. Meng, and R. Siegwart, “Towards real-time multi-sensor information retrieval in Cloud Robotic System,” in *2012 IEEE Conf. Multisensor Fusion and Integration for Intelligent System, MFI.* 2012, pp. 21-26.

[17] L. Tan and N. Wang, “Future internet: The internet of things” in *2010 3rd Int. Conf. Advanced Computer Theory and Engineering, ICACTE.* 2010, pp. V5-376.

[18] Willow Garage. (2013, December) *PR2*. [Online]. Available: https://willowgarage.com/pages/robots/pr2-overview

[19] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, *et al.*, “Roboearth,” *Robotics & Automation Magazine, IEEE,* vol. 18, pp. 69-82, 2011.

[20] The RoboEarth Consortium. (2013, September). *What is RoboEarth?* [Online]. Available: http://roboearth.org

[21] OpenStack. (2014, June). *OpenStack Cloud Software.* [Online]. Available: http://www.openstack.org

[22] Proxmox Server Solutions. (2014, June). *Proxmox.* [Online]. Available: http://www.proxmox.com

[23] Oracle Corporation. (2014, June). *Mysql The world’s most popular open source database.* [Online]. Available: http://www.mysql.com

[24] Y. Chen and X. Bai, “On robotics applications in service-oriented architecture,” in *28th Int. Conf. Distributed Computing Systems Workshops, ICDCS'08*. 2008, pp. 551-556.

[25] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, “Mobile emulab: A robotic wireless and sensor network testbed,” in *IEEE INFOCOM,* 2006.

[26] Y. Kato, T. Izui, Y. Tsuchiya, M. Narita, M. Ueki, Y. Murakawa, *et al.*, “RSi-cloud for integrating Robot Services with internet services,” in *IECON 2011-37th Annu. Conf. IEEE Industrial Electronics Society*, 2011, pp. 2158-2163.

[27] Raspbian. (2014, April). *Welcome to Raspbian*. [Online]. Available: http://www.raspbian.org/

[28] Raspberry Pi Foundation. (2014, March). *About Us*. [Online]. Available: http://www.raspberrypi.org/about

[29] G. van Loo. (2014, March). *GertDuino User Manual*. [Online]. Available: http://www.element14.com/community/servlet/JiveServlet/downloadBody/64534-102-2-287165/User%20manual%20Gerduino%205.6.pdf

[30] SparkFun Electronics. (2014, March). *Ardumoto – Motor Driver Shield*. [Online]. Available: https://www.sparkfun.com/products/9815

[31] G. Henderson. (2014, April). *ATmega Setup*. [Online]. Available: https://projects.drogon.net/raspberry-pi/gertduino/atmega-setup/

[32] Friends of the Unicorn. (2014, April). *Raspberry Pi-GertDuino Serial*. [Online]. Available: http://friendsoftheunicorn.net/content/raspberry-pi-gertduino-serial

[33] CE Linux Forum. (2014, April). *RPi Serial Connection*. [Online]. Available: http://elinux.org/RPi_Serial_Connection

[34] TightVNC. (April 2014). *TightVNC Software*. [Online]. Available: http://www.tightvnc.com/

[35] K. C. Chan, J. Katupitiya, J. W. Hanrahan, C. J. Jackson, D. Rose, “An emergency communication system for an agricultural autonomous vehicle,” in *Proc. Int. Conf. Intelligent Agriculture, IPCBEE*, vol. 63, 2014, pp. 76-82.

[36] K. C. Chan and M. Martin, “An integrated virtual and physical network infrastructure for a networking laboratory,” in *Proc. 7th Int. Conf. on Computer Science and Education, ICCSE.* Melbourne, Australia, 2012, pp. 1433-1436.

[37] K. W. Ross and J. F. Kurose, *Computer networking: a top-down approach featuring the Internet*, 6th ed. England: Pearson Education Limited, 2013.

[38] C. Pautasso, O. Zimmermann, and F. Leymann, “Restful web services vs. big'web services: making the right architectural decision,” in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 805-814.

[39] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, “Comparison of JSON and XML Data Interchange Formats: A Case Study,” *Caine,* vol. 9, pp. 157-162, 2009.