Presented at the 9th IEEE International Conference on Intelligent Environments, Athens, Greece, 16-19 July 2013

Educational Living Labs; A novel Internet-of-Things based Approach to Teaching and Research

Jeannette Chin
Computing and Technology
Anglia Ruskin University
Cambridge, United Kingdom
jeannette.chin@anglia.ac.uk

Vic Callaghan
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom
vic@essex.ac.uk

Abstract **-** This paper explores some a novel approaches to harnessing the Internet-*of-Things (IoT)* as a teaching and research vehicle in education. For teaching we argue that the *Internet-of-Things* provides a highly motivating topic to capture students' imaginations, and a perfect platform for teaching computer science. In addition, we explain the potential for entire campuses or buildings to be constructed from *Internet-of-Things* technologies and the potential for this infrastructure to act as a teaching platform. This proposition is perfectly captured by the axiom "*The college building (or campus) is the lab*". This philosophy is part of a wider movement that started in the EU, called Living Labs. In achieving these aims, our work seeks to combine a number of concepts; first we utilise the *Internet-of-Things*, second we incorporate *Living Labs* ideas, third we harness the *iCampus* vision, forth we use the '*Smart Box*' concept and finally we implement the Pervasive-interactive-Programming (PiP) paradigm. We contend this approach can be used in various mixes to produce highly motivating and effective educational environments. We illustrate this work by describing the application of these ideas to a real-world venture, the Harlow UTC (in the UK). The main focus of this paper concerns the use of PiP, together with the *Internet-of-Things,* to teach elementary programming skills. In in support of this we present results of an evaluation of PiP with 18 participants (students and staff) of varied age and gender. The main conclusions of these evaluations were that PiP enabled students and staff, with diverse backgrounds, to quickly master the programming skills involved. The paper concludes by describing our future plans for this work.

Keywords-component; Internet-of-Things; Cloud-of-Things; End-User Programming; Intelligent Campus; Smart Boxes, Living Labs, Intelligent Environments, Education

I. Introduction

Customisation of environments is a widely displayed human desire, ranging from people decorating their homes to personalising their phone screens. The Internet-of-Things (IoT) extends this capacity for customisation into the world of embedded-computer devices that are integrated into the very fabric of our physical realities; the buildings and cities we live in. The Internet-of-Things is an umbrella term used to describe the emerging trend of embedding small Internet-enabled computers into everyday objects (e.g. alarm clocks, TVs, environmental-sensors e.t.c.) thereby enabling the built environment to stretch beyond the more regular '*bricks and mortar*' into the intelligent environments world of '*sensors and effectors*", which opens up a whole new market and economy. There are no reliable estimates for the size of this market but a report by the "Arthur D. Little management consultancy" suggests that by 2020 the IoT market could be worth between 22 billion and 50 billion dollars [19] made up of some 16 billion connected devices [20]. This huge emerging market presents tantalizing opportunities to businesses while at the same time placing new demands on educational institutions to provide graduates with the required skills. This paper addresses this challenge in a novel way by exploring the possibility of using the box-like spaces we inhabit (e.g. buildings, rooms or there similes) as the laboratory infrastructure for teaching computer science. In this paper we explore some of the issues relating to teaching computer science to14-18 year old college students, many of whom are computer-programming novices. This work is part of a larger and on-going project in the Harlow UTC which is part of a larger UK government strategy to improve the UK's science and engineering base by providing better and more diverse pre-University educational options. In the following pages we will introduce our concepts and present some preliminary results of evaluations of PiP (Pervasive-interactive-Programming) gleaned from trials with a varied set of 18 academic students and staff.

II. Internet of Things

Since Sir Tim Berners-Lee first proposed the World-Wide-Web (WWW), back in late 1980s, the Internet-of-Things (IoT), it has gradually spread into everyday environments. The IoT has been described by United Nation [16] as "*physical devices that have network connectivity that provide various services*" (e.g. pollution sensing, door locks etc.) A combination of mass production and innovative technologies, such as tiny radio frequency identification tags (RFID), enable networked technology to proliferate widely, giving rise to a number of related research areas such as Intelligent Environments, Ambient Intelligence, Smart Homes, Pervasive and Ubiquitous computing. Today, the Internet-of-Things is not just regarded as a community of network enabled devices, but rather it is seen as "*a layer of digital connectivity on top of existing infrastructure comprising networked things*" [17]. Thus the technologies concerned are wide ranging in nature and encompass network infrastructure, services, tools and applications.

Given the open-ended nature of the IoT, the massive connectivity and the huge numbers of potential users and applications, the possibilities are virtually endless. Thus it is not surprising to find that the Internet-of-Things technology has been applied to creating a wide range of applications such as smart homes, factories, cities and even university campuses [1]. The UK government and the European Union have supported and funded a number of such projects [8] including a European Initiative on Smart Cities [10] which has investigated issues such as green and renewable low carbon energy, energy networks using smart grids, smart metering and future transport systems. There are a numerous cities around the world that are exploring the smart city concept for example Songdo in South Korea which has a population of 350,000 inhabitants [7] has spent $35bn (£23bn) on a project that has explored the use of the IoT [10] with a particular emphasis on green and sustainable issues [6].

In this paper we have focused on exploring the application of IoT technology to the Intelligent Campus, a holistic model for ICT in education. This concept is explained in more detail in the following section but at this point it is convenient to think of an iCampus and a smaller scale version of a smart city.

III. The iCampus

The origins of the iCampus concept can be traced back to a seven-year project by MIT which had the ambitious goal of revolutionising the practice of higher education [21]. Later, other researchers continued this work [1] [22] [23] [24]. Of particular relevance to this paper is work by University of Essex in 2010 which created an iCampus in the form of a test-bed for the exploration of digital service delivery based on different networking technologies [14]. Shortly after, British Telecom (BT) proposed a more broadly based ICT version of the iCampus which they characterized as a knowledge ecosystem that included numerous stakeholders ranging from students, through teachers to managers [18]. The BT iCampus model is especially interesting for the work described in this paper as it decomposes the ICT activities of an iCampus into six major issues, namely learning, management, governance, health, socializing and the environment which is illustrated in Figure 1, below.

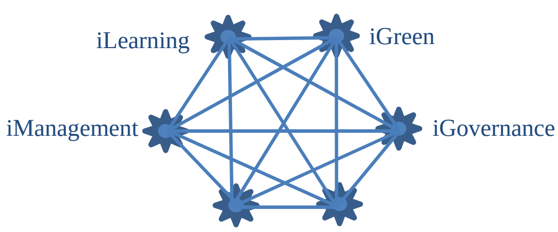

Figure 1: The 6 Pillars of the iCampus [18]

The diverse nature of the iCampus makes it an inherently interdisciplinary concept. This model can be viewed from different perspectives. From a student's perspective learning is the core issue with issues such as health and socializing being also important. Of course how the University is governed and managed is also important to students but is not so high up their perception hierarchy. Likewise for the University management they will have their own perception hierarchy with issues such as governance, management and green issues being high in their awareness. In this paper we will be looking at the iCampus model from a student learning perspective. Like the iCampus model we will be utilizing a network intensive framework of connected devices and service; in this case to produce a type of Living Lab, that forms the basis of student teaching and research, which is introduced in the follow section.

IV. Living Labs

The term 'living lab' was, allegedly, coined by William Mitchell, Kent Larson, and Alex Pentland at MIT but enthusiastically adopted by the European Union, most notably in their European Network of Living Labs (ENoLL)[1] which was founded in November 2006 and is an international federation of some 300 benchmarked Living Labs. A Living Lab is, as the name suggests, a living environment which houses both people and technology, in a semi experimental setting that promotes symbiotic innovation, development and research. An example of such a living lab is the Essex University iSpace which is a purpose built domestic apartment in which researchers live and innovate new technologies for the smart home [30]. The key aspect of a Living Lab is that it transforms the role of the users, from being observed subjects, into being members of a co-creative ecosystem. This is what we are trying to achieve in our vision for IoT buildings; an educational environment in which the underlying technology and structure forms both the teaching and research facilities, together with supporting everyday working and living activities, thereby creating a novel type of educational establishment which we characterise by the axiom "The college building (or campus) is the lab". We are approaching these lofty goals by adopting a 'smart-box' strategy, where boxes range in size from desktop facilities to whole rooms, buildings or even campuses. Indeed, we see the concept of the iCampus and a Living Lab as being complementary and mutually supportive models. Our particular approach is introduced in the following sections.

[1] http://www.openlivinglabs.eu/

V. The Internet-of-Things and iCampus as a Framework for Education Reform

From the above sections it can be seen that the iCampus and the IoT share pervasive networking as their key enabling technology. The iCampus harnesses network services to improve campus management and act as a vehicle for more efficient delivery of learning content whereas the IoT can provide both the essential gadgets for an iCampus (sensors, effectors, mobiles phones etc) and a multidisciplinary topic for learning. Thus, in this paper we argue that the iCampus and IoT are natural bedfellows to be combined to produce a novel and cutting edge campus which, when used for teaching or research, becomes a Living Lab. This approach fits with current trends such as the increasing interest in using mobile devices as the medium for teaching and learning [3]. This trend is further compounded by the rise of interesting new embedded Internet technologies such as mbed and Raspberry Pi [31] both of which are supported by the BuzzBoards[2] Internet-of-Things kit (Figure 2) which provides a versatile and cost-effective platform for learning computer science [15]. More significantly, there is a growing view by governments around the world that computer science is an important skill for modern knowledge based economies. For example, in the UK there is a government driven curriculum shift from an application centred studies of ICT to more science oriented studies of computing starting in pre-University education such as the new AQA-GCSE Specifications for Computer Science that will be implemented in UK schools from 2014 onwards [2]. It is not just governments that are pushing such reforms but so too are companies who depend on these skills to compete in the global economy. For example Google strongly advocates these policies [5]

Figure 2 BuzzBoards IoT Kit (some examples)

Likewise the software giant, Microsoft, has gone further and urged primary schools to consider teaching Computer Science, a view that has also found favour in the UK Department of Education [4]. Studies [26] [27] suggested that a customisable curriculum, which provides the flexibility to cater the needs of individual students, is the way to go forward in education. Some have argued that in order to employ the concept effectively a 'paradigm shift' is required to restructure society's thinking, practices, and policies' [25]. These studies fit very well with the IoT which provides a rich and flexible platform for students and faculty alike to explore and flourish. Thus, given that the IoT mirrors these ideas and provides a cheap versatile and highly motivating framework, we argue that it is ideally placed to support these educational reforms. Moreover, because it's based on pervasive networking, we also argue it fits well with the iCampus vision.

[2] www.fortito.com

VI. The Proposed Framework

A. *The Smart-Boxes*

Clearly, wiring up an entire campus or building with Internet-of-Things would be expensive so, as much as we would like to build an entire IoT Building, in the Harlow UTC project we are starting with smaller room and desktop sized spaces which present a more economic option for the education sector which is highly cost sensitive. Some example of previous room-sized approaches include the Essex University iDorm (an IoT based student dormitory) [30]. Another approach was the BT CEL (Customer Evaluation Laboratory), constructed as a 'box' within a lab, that mimicked a living room, which they used in the early millennium as part of their Customer Engagement Model. We have taken inspiration from this earlier work and have proposed a hierarchical Smart-Box (HSB) concept. In this we simply regard a building, city or transport as being a set of high-tech boxes. The function of the 'box' depends simply on the sub-functions provided. A key aspect is to view these 'boxes' as being part of a hierarchy so, for example, that a collection of them forms a building, and a collection of these, a city. With regard to student education that enables us to move in the opposite direction down this hierarchy and create desk-top "smart-Boxes" that are akin to smart-rooms or buildings, which students can use to learn about intelligent environment technology and computer science. A picture of one of these desk-top boxes (used for a plant care application) is show in the following figure (figure 3).

Figure 3 The pDorm

While these desk-top boxes are perfect for student development work they are too small for students to inhabit the spaces, so as part of the new Harlow University Technical College (UTC) building we have also produced a number of conceptual drawings for a much larger (and inhabitable) boxes. One is a box inside the lab, the other a more conventional smart home. At this stage it has not been decided which (if any) of these larger scale concepts to use, although both offer the same type of functionality. These are shown in figure 4a and 4b.

Figure 4a - Architects artistic impression of inhabitable smart box

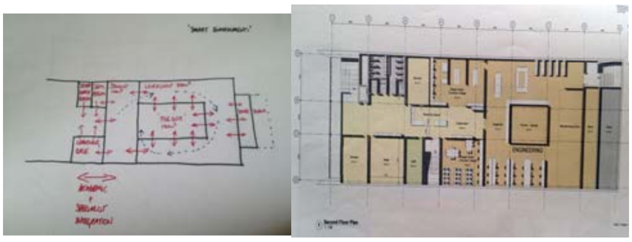

Figure 4b - Architects sketch of inhabitable smart-home

In this way students are able to learn how to develop the basic computer science technologies of intelligent environments and the Internet-of-Things using the desk-top boxes, before evaluating the technologies with real people in the larger space. Using this approach Harlow UTC aims to create a new kind of innovative and imaginative learning model (from curriculum design to delivery of courses) to equip students aged between 14 and 18 years old to meet the employment challenges/needs of the 21st century. Thus, the hierarchical Smart-Box educational model provides an innovative teaching and learning environment that includes a combination of enquiry based hands-on / practical learning, where students can work individually or in teams to explore their interests in order to realize their full potential.

B. The Internet-of -Things as a Teaching Framework

From a computing point of view, programming is a fundamental skill that students will be required to learn and, the exemplar provided in this paper, will focus on how that can be achieved by the use of a new paradigm called *Pervasive-interactive-Programming (PiP).* Programming the IoT is potentially a very challenging task as the IoT is effectively a highly distributed system but we will show in this paper how even novice programmers can use PiP to programme the IoT, advancing from simple to advanced programming concepts needed to tackle university level curriculums. However, before moving to explain the exemplar we will first outline the technical framework in the following section.

C. The Technology Framework

The proposed IoT education framework fits within the iCampus model introduced in section III. At the core of the concept is the notion of inter-device messaging which we facilitate via the Cloud, creating a composite *Cloud-of-Things* (CoT) model, as illustrated in Figure 5. This model forms the basis of the infrastructure we employ which, in addition to the technical benefits, introduces students to the important topics of big-data and cloud-computing. It is useful to note that the broader aims of the Harlow UTC building supports the following components of the iCampus model:

1. iLearning by supporting and inspiring students to learn future cutting edge technologies
2. iSocial by providing a high-end safe environment platform for students and faculty to interact
3. iHealth by providing a high-tech platform suitable to teach and inspire the next generation medical applications/equipment while at the same time providing meaningful/useful access to health related information to students and faculty
4. iGreen by providing technologies to increase energy efficiency and lower carbon waste.

The focus of the work in this paper is on the first of these pillars, iLearning in the area of Science and Technology innovation. In particular, it provides a suitable programming platform to support the currently envisaged GCSE Computer Science curriculum, as championed by the UK government.

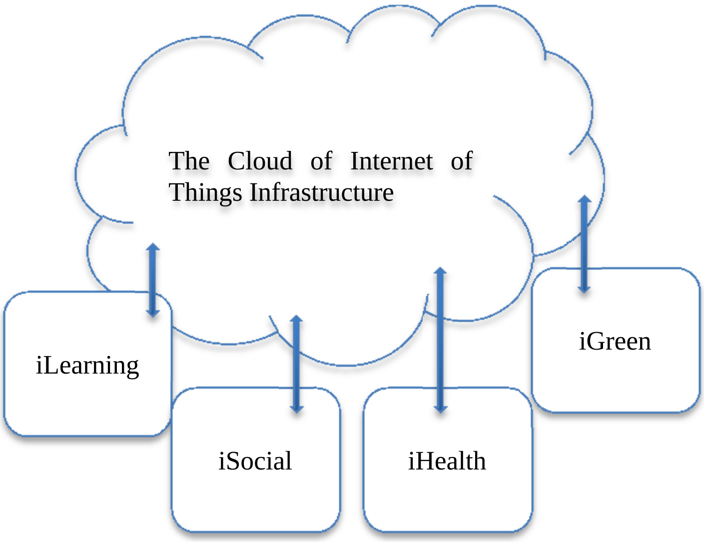

Figure 5 – Composite Cloud of Things Model

VII. TEACHING EXEMPLAR

As explained earlier, PiP [11] employs a show-my-by-example approach to programming which is well suited to the iCampus environment. PiP utilises a mechanism that facilitates end-users (students and faculty staff) to “create” programs themselves, without prior programming skills. As part of the process it enables students to quickly acquire an understanding of how computer programs actually work, by guiding them through a simplified scheme for creating programs leading to them to construct "programs" themselves.

Furthermore, PiP is aimed at programming distributed computers, embedded into real physical appliances, such as those that make up the IoT. The concept underlying PiP is simple in that it mimics the 'playful' method that has been used to teach children for generations – the teacher demonstrates an action by showing an example; the learner then repeats the demonstrated action. In PiP, the 'things' that comprise the programming environment are categorised into 2 types: (a) physical 'things' (including graphical representations of them) which we call "hard things", (b) abstract things (e.g. application software such as email, instant messaging etc.) which we call "soft things". Both types of 'things' provide their functionalities in a form of services that are network discoverable and accessible.

The availability of numerous networked 'things' and their services present an opportunity to connect groups of them together to provide meta-services or meta-appliances. In PiP we called this the "deconstructed appliance/application model" which can be regarded as a form of virtual application/appliance [11]. The representation (specification) of such virtual entities is referred to as a MAp (Meta-Appliance/Application). It contains detailed information about the "things", and rules that govern the functionalities of them. Rules are created by the end user as part of programming the system.

A. *Pervasive interactive Programming (PiP) for IoT*

The inspiration for using PiP to teach students how to program computers arose from two perspectives. First was the observation that much of the learning in early childhood arises from replicating examples of behaviour that people observe in each other [28] and secondly, that at the heart of pervasive interactive programme (a methodology based learning-by-example) are *IF-Then-Else rules,* that are the core construct of procedural programming languages [11]. Thus, by providing a mechanism to translate behaviour examples into rules, we have a natural and intuitive model of procedural programming.

The basic principle of PiP is to capture the macro behaviour of a system as a series of micro tasks, by the user demonstrating to the system examples of the required functionality. PiP captures and describes these micro tasks by means of rules. Sets of these rules are then are combined to form the macro level behaviour of the system. In PiP, rules are normally internalised by the system, and are not usually visible to the user. However, for using PiP as a teaching tool, making the rules visible becomes an important aspect of the pedagogical process. This is important because the core construct of both PiP behaviours and procedural programs are rules. This is illustrated in Figure 6. Rules enable decision-making which is a key property for any entity that purports to be smart. Its contribution to making Von-Neumann style computers powerful and flexible problems solving machines, is hugely attributable to there being a decision making mechanism embedded into its structure.

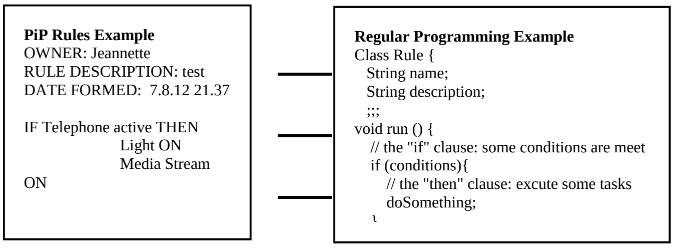

Figure 6 – Equivalence between PiP Rules and Regular Programming Constructs

It would be hard to exaggerate the importance of decision-making functionality in computation. Without the ability to make decisions computers would be relegated to machines that stepped unidirectional through lists of instructions. Computers would still be programmed, in the sense someone wrote out those lists of instructions, but the ability of a computer to restructure its computational strategy and adapt to new problems or contexts would be severely (perhaps fatally) restricted. The use of decision making structures comes at a large cost; understanding, designing and handling the numerous combinations of resulting program control flows, which has the potential to become a task of daunting complexity. In fact, this is such a big issue it is one of the key targets of software engineering [12]. Thus for students learning programming a key, and complex, aspect is to understand the nature of decision constructs (their relationship to sense-action pairs from data and physical domains), is how such decision constructs are formed and how they contribute to the functionality of the overall program. Our approach is to focus on exposing and understanding the role of rule formation as the key construct in computer programming. We do this by employing PiP to show the linkage between sense-action pairs in the real world, and the creation of corresponding rules in the computational machine. We then link these rules to the if-then-else programming constructs in procedural programming languages, which we then use as the launch-pad for students entering and following the more regular route to learning to program. In this sense our technique is to provide an intuitive and interactive (constructionist) pedagogy for introducing programming concepts to students who are not, initially, technically literate. It is important to note the scope of PiP in relation to computer science education; we envisage it as a means to introduce non-programmers to programming, in the initial phase of their computing education. Thus we see it as abridge for non-programmers (buy introducing the nature and role of rules) into programming via traditional languages such as C or Java. As part of a curriculum is might occupy the first few weeks of the students programming education. Beyond education, PiP is a powerful tool for enabling the population at large to customise their computational spaces.

1) An Example

To illustrate the principle of how we use PiP as a tool for teaching computer programming, we will first present a scenario that involves programming a community of ‘things’ to provide some coordinated functionality. In this example there are three main components being controlled; a telephone, light and media player. The basic idea is that the system should be programmed to provide behaviour such that when an incoming call arrives, the lights would raise and the media player would stop. Figure 7 – 9 illustrates PiP in use. In a typical teaching session, before logging the system the student is provided with examples of regular programs (e.g. C and Java) that have the important constructs such as rules, highlighted. At this point they are not expected to understand the program, just to appreciate the main elements. The students are also shown simple examples of programmed coordinated activity (e.g. a media player coordinating actions with a light).

Figure 7 - a) User log in screen b) select ‘things’ by dragging to composing area

Figure 8 - a) constructing rule b) rule formed (Jeannette_Rule708)

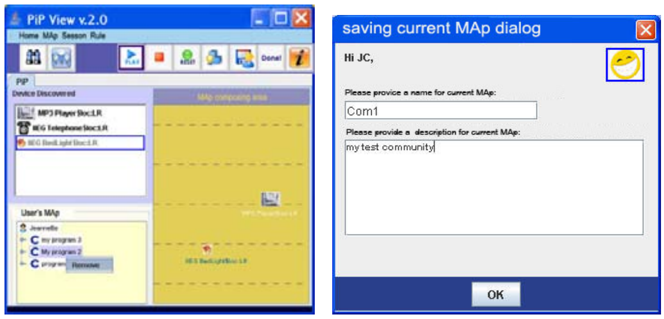

Figure 9 - a) deleting MAps (rules) b) saving MAps (rules)

The student is asked to write down the behaviour of the system they wish to create, as a simple state-action list. The student then logs into the system and begin to translate the specification they written into “programs” using PiP interface – via steps (1) exploring things (2) selecting things by dragging and dropping them into the ‘programming area’ via the graphical interface control panel (Figure 7). In the second phase the student then demonstrates the actions (i.e. the specified behaviour), which PiP translates into a set of rules, see Figure 8. Once the student finishes their “programming exercises” they will able to see the program descriptions in the form of plain text (showing the sense-actions) or actual programming code (the PiP translation comprising rules). The student is then encouraged to replay the program to verify the intended behaviour occurs (ie that the specification is met) before saving their program, see Figure 9. In the third phase the student is invited to look at the constructed rules and relate them to the actions. It is then explained to the student that a simple procedural language uses these rules as the core decision construct.

The student is then invited to re-execute the program to verify its action before being invited to alter the rules manually via the interface, without going through the demonstration cycle again, to make the environment achieve a modified behaviour. The student is then shown how such rules can be translated into actual programming code (by the addition of auxiliary code such as declarations and operators). Finally, students are then encouraged to compare the manual and automatically generated code to deepen their understanding. Thus, this forms the basic introduction to programming and the task may be progressively made more complex to increase the confidence and programming skills of the student. From this the student is then asked to write the conditional constructs in regular programs as part of the transition to understanding and programming in more conventional programs such as C or Java.

VIII. Evaluation

PiP was evaluated with some 18 participants (a mix of students and staff who typified the target group our research aims at) who were invited to programme an open-ended design (of their own choosing) to produce a working IoT system. With a gender mix of 10 females and 8 males and ages ranging from 22 to 65, participants were divided into six groups based on their technical knowledge ranging from have no experience to expert in the field.

The evaluation took place in a box-like space equipped with numerous IoT devices varying from appliances, sensors, and actuators through to special purpose equipment to support evaluations of this type. An evaluation methodology was developed with the assistance of socio-technical research staff [29] to evaluate six usability dimensions of PiP - the overall concept, user controls, cognitive load, information shows in Figure 10. Methods used in the evaluation involved evaluation observations and participants filling in a questionnaire after the session.

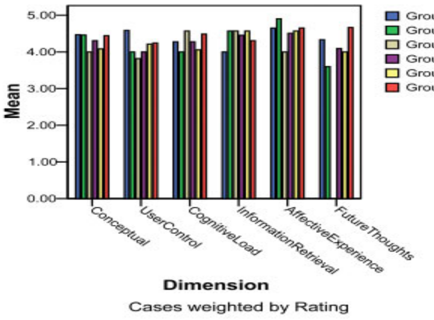

Figure 10 Mean ratings for each individual group of participants evaluating six usability dimensions.

An analysis of the evaluation data using SPSS showed that, in general, all the dimensions rated well (scoring above 4) indicating the users were generally well satisfied with PiP as a method for introducing them to programming IoT devices. Our results showed that 83% were able to use PiP to program their own IoT systems with little or no assistance and reported they found PiP very intuitive to use. In terms of general observations, none of staff or students appeared to find the principles difficult to understand. A 94.4% of all participants stated they felt it rewarding when they were able to design and program their own IoT system. The results are published in greater detail in an earlier paper [13] and showed conclusively that novice programmers found physically demonstrating required system behaviour to be an effective way of learning to program computers.

IX. Conclusion

In this paper we have introduced a novel approach to educating students in computer science based on combining concepts taken from Pervasive interactive Programming, the *Internet-of-Things*, the *iCampus*, *Living Labs* and a concept we term the *hierarchical 'Smart-Box'* model. In doing these we explained the technologies and principles involved. For example we explained how these can be integrated into a framework that we have labelled "*The Cloud of Things*" (CoT). We have argued that the IoT will provide a highly motivating environment for students to learn about Computer Science and programming, as it encapsulates all the key features and issues of modern computing, but wrapped into a flexible and highly motivating application. Moreover, we have discussed how this fits with the current education and technology trends. In order to make this model function effectively, some formidable obstacles need to be overcome. For example, an IoT system is essentially a distributed computer system, which historically have proven difficult to programme. Thus, using this framework to teach programming, especially to introduce novices to programming, is potentially difficult. To overcome this we introduced an end-user programing paradigm that we have developed called *Pervasive interactive Programming (PiP)* and evaluated it with 18 participants. These evaluations have demonstrated that students and staff with diverse backgrounds (including non-programmers) were quickly able to master the skills and understand the concepts involved. Thus we argue that our methods show some good potential for introducing students (especially pre-university level students) to programming in a way that is simple, motivating and eases their path to programming with more regular high-level languages by introducing them to the core program constructs, namely conditional rules. Finally, we proposed a hierarchical smart-box based approach to decomposing the real world into discrete spaces that range from desk-top units to entire buildings or campuses. The desktop units represent a particularly cost effective solution for teaching large numbers of students as it fits easier with the academic needs and budgets of educational establishments. The work has involved a number of threads, each with significant research potential that we hope to explore as part of future work.

Acknowledgment

We are grateful to partners at Harlow UTC for their support in this project, in particular Sue Young (Projecct Manager) and Sam Stevens (Architect). Also, a special acknowledgment goes to Professor Marcian Cirstea and Professor Mike Cole for their helpful advice and encouragement.

References

[1] Jackson M "Intelligent Campus". 1st International Symposium on Pervasive Computing and Applications, pp3-9, 2006

[2] BBC news: http://www.bbc.co.uk/news/education-16493929 Last accessed on 05-03-2013.

[3] BBC news: http://www.bbc.co.uk/news/education-16186705 Last accessed on 05-03-2013

[4] BBC news: http://www.bbc.co.uk/news/education-21250567 Last accessed on 05-03-2013

[5] BBC news: http://www.bbc.co.uk/news/technology-21243825 Last accessed on 05-03-2013

[6] BBC news: http://www.bbc.co.uk/news/technology-20957953 last accessed on 05-03-2013 .

[7] BBC news: http://www.bbc.co.uk/news/technology-20957953 last accessed on 05-03-2013

[8] EU Smart cities project http://www.smart-cities.eu/model.html Last accessed on 06-03-2013

[9] MIT Smart Cities project http://cities.media.mit.edu/ Last accessed on 06-03-2013

[10] European Initiative on Smart Cities http://setis.ec.europa.eu/about-setis/technology-roadmap/european-initiative-on-smart-cities Last accessed on 06-03-2013

[11] J. Chin, V. Callaghan, G. Clarke, "End-user Customisation of Intelligent Environments". In the handbook of Ambient Intelligence and Smart Environments, Editors: H. Nakashima, J. Augusto, H. Aghajan, Springer, 2010, Spring, pp 371-407, ISBN 978-0-387-93807-3

[12] Callaghan V, Barker K, "SAS-‐ an experimental tool for dynamic program structure acquisition and analysis", Journal of Microcomputer Applications (1982).5, .209-‐ 223.

[13] Jeannette Chin, Vic Callaghan, Adrian Winckles, "Personalising the iCampus; an End-User Programming approach", IEEE The International iCampus Symposium 2012 (IC'12), Macau, China, 4-7 December 2012

[14] K. Guild, M. Paredes Farrera, R. Martin, R. Almeida, A. Bontozoglou, M. Patel, K. Yang, V. Callaghan, "STUDENT: Scenarios, Technologies & Users within the Digital Essex Network Testbed", 6th International Conference on Intelligent Environments (IE'10), Kuala Lumpur, Malaysia. 19-21st of July 2010

[15] V Callaghan, "*Buzz-Boarding; Practical Support For Teaching Computing, Based On The Internet-Of-Things*" (presentation), 1st Annual Conference on the Aiming for Excellence in STEM Learning and Teaching, Imperial College, London & The Royal Geographical Society, 12-13 April 2012

[16] BBC news: http://news.bbc.co.uk/1/hi/technology/4440334.stm Last accessed on 06-03-2013

[17] The Internet-of-Things council http://www.theinternetofthings.eu/internet-of-things-what-is-it%3F Last accessed on 06-03-2013

[18] Jason W. P. Ng, "The intelligent campus (iCampus): End-to-end learning lifecycle of a knowledge ecosystem", EBTIC's International iCampus initiative white paper, pp. 1-6, May 2010

[19] Schlautmann A, Levy D, Keeping S, Pankert G, (2011) " Wanted: Smart market-makers for the "Internet of Things", Arthur D. Little, management consultancy, see http://www.adlittle.se/prism_se.html?&view=383 (February 2012)

[20] Vermesan O, Friess P, (2011), "Internet of Things - Global Technological and Societal Trends Smart Environments and Spaces to Green ICT", River Publishers, Denmark, ISBN-10: 8792329675

[21] MIT 2006 - The iCampus http://icampus.mit.edu

[22] Chen YB, Yang CJ, Jong GJ, "Intelligent Campus Multi-application RFID Integration System", Intelligent InformationHiding and Multimedia Signal Processsing, pp 504-507, 2008

[23] Davies M, Callaghan V, Garrdener M, "Towards a Mixed Reality Intelligent Campus", Intelligent Environments 2008

[24] Machado LS, Costa TKL, Moraes RM, "A 3D Intelligent Campus to Support Distance Learning, Information Technology Based Higher Education and Training 2006

[25] Barajas, H. and Higbee, J. Where do we go from here? Universal Design as a model for multicultural education. in Higbee, J. (ed.) Curriculum transformation and disability: implementing Universal Design in higher education. Minnesota: Center for Research on Developmental Education and Urban Literacy, University of Minnesota, 2003.

[26] Higgle, J. (ed) Curriculum transformation and disability: implementing Universal Design in higher education. . Minnesota: Center for Research on Developmental Education and Urban Literacy, University of Minnesota, 2003.

[27] Hall, T. and Stahl, S. Using Universal Design for Learning to expand access to higher education. in Adams, M. and Brown, S. (eds.) Towards inclusive learning and teaching in higher education: developing curricula for disabled students. 2006 Abingdon: Routledge.

[28] Ann L Brown "Preschool children can learn to transfer: Learning to learn and learning from example", Cognitive Psychology, Volume 20, Issue 4, October 1988, Pages 493–523

[29] DiDuca D and Van Helvert J. "User Experience of Intelligent Buildings; A User-Centered Research Framework", Intelligent Environments 2005, Essex, 28-29th June 2005

[30] Callaghan V, Clark G, Colley M, Hagras H Chin JSY, Doctor F "Intelligent Inhabited Environments", BT Technology Journal , Vol.22, No.3 . Klywer Academic Publishers, Dordrecht, Netherlands, July 2004

[31] Raspberry PI http://www.raspberrypi.org/ Last accessed on 06-03-2013