Making the invisible visible in constructionist learning tasks: an explanation framework based on a Pedagogical Virtual Machine (PVM)

By

Malek Qalit Alrashidi

A thesis submitted for the degree of PhD in Computer Science
School of Computer Science and Electronic Engineering
University of Essex
September, 2017

Acknowledgements

First, let me express my gratitude to Almighty God for all the mercy He showered me during my PhD journey by providing me with the strength and perseverance to accomplish this work.

I am very grateful to my supervisors, Dr Michael Gardner and Prof. Victor Callaghan, for their generous support, help, and encouragement through this research. I got my inspiration from their trust and belief in me completing this work. Thanks again.

I cannot express my grateful thanks to my parents, Qalit and Haia, for their patience of me being away from them for almost eight years in the U.K. They have always been supportive and stand with me in every single day. I started this journey as a single person, but I ended up with a lovely wife and two gorgeous daughters. My sincere thanks and gratitude goes to my wife, Taghreed, for her endless love and motivation, to my daughters Maya and Alaa, and to my brothers (Jamal, Aymn, Mohammad, Essam and Sultan) and sisters (Malak, Amani, Alaa and Hala) for their constant care and encouragement.

Special thanks to Dr Anasol Pena Rios and Malcolm Lear for their help during the development of Buzzboards and when I face technical issues. Thanks to those students who participated in the evaluations phase. Not to be forgotten are the students and staff members of the School of Computer Science and Electronic Engineering of Essex University in the United Kingdom. Thanks to the board meeting members (Dr Martin Colley and Dr David Bebbington) for their constructive feedback and comments.

I would like to thanks Tabuk University for giving me the opportunity to complete my master's degree and PhD in the U.K.—without their financial support, this journey would not be completed.

Finally, thanks to all my friends here in the U.K. and to those in Saudi Arabia.

Abstract

In today's digital world, the use of diverse interconnected physical computer based devices, typified by the Internet-of-Things, has increased, leaving their internal functionalities hidden from people. In education, these hidden computational processes leave learners with s vagueness that obscures how these physical devices function and communicate in order to produce the high-level behaviours and actions they observe. The current approach to revealing these hidden worlds involves the use of debugging tools, visualisation, simulation, or augmented-reality views. Even when such advanced technologies are utilised they fail to construct a meaningful view of the hidden worlds that relate to the learning context, leaving learners with formidable challenges to understand the operation of these deep technologies.

In working towards a solution to this challenge, this thesis combines computing and pedagogical models in a novel way to improve learning and teaching of computer science. This framework (a combination of computational and pedagogical models) is the core contribution of my thesis and has been given the name a Pedagogical Virtual Machine or PVM). It aims to extract learning-related information from the underlying computers that make up the education focus by providing a layered analysis of the technical and pedagogical processes that interact together for any given learning activity (in the context of learning about embedded computing). It adopts an object-oriented perspective that deconstructs computation and learning into objects, while taking inspiration from the Java Virtual Machine ideas, thereby building on existing paradigms of 'learning objects', 'object oriented programming' and virtual machines. In this way it addresses the challenge of linking both computing and learning activities in a standardised way across a multiplicity of computing and learning environments.

The use of augmented reality (AR), and its ability to reveal deep technologies, further improves the effectiveness of the PVM framework introduced above by superimposing data, in real-time, concerning the invisible computational processes being explored by the learners. Applications that learners and developers might use this PVM tool for are typified by topics such as the Internet of Things, pervasive computing and robotics. The study presented in this thesis is based on the latter, robotics.

The learning effectiveness of an AR based PVM approach was evaluated in two educational experiments that concerned students learning to program a desk-based robot (which was used as an example of an embedded-computer). The two experiments were conducted with computer-science students from the University of Essex and differed in the level of complexity. The results showed that PVM with AR significantly improved the students' learning achievement and performance than those who used traditional learning environments. In addition, the PVM with AR made a positive difference to students learning experience, supporting the use of PVM with AR in educational activities that involved dealing with abstract technologies.

Table of Contents

List of Figures

List of Tables

Acronyms

Term	Description
MR	Mixed Reality
AR	Augmented Reality
IoT	Internet-of-Things
PVM	Pedagogical Virtual Machine
VM	Virtual Machine
ASM	Algorithmic State Machine
ARLO	Augmented Reality Learning Object
ARLE	Augmented Reality Learning Experience
LO	Learning Object

Chapter One

1. Introduction

> *"Have a vision. It is the ability to see the invisible. If you can see the invisible, you can achieve the impossible"*
>
> *-Shiv Khera*

Learning is a natural aspect of being human. People like to be involved in processes whereby they can gain knowledge and skills and improve themselves. Pena-Rios et al. (2012) describe some of the recent digital technologies used in the educational setting, such as mixed reality, augmented reality, and virtual environments, which have transformed learning and teaching from traditional methods to new high-tech methods. These emerging advanced learning technologies has shown great potential for teaching and learning (Dede, 2009; Kerawalla et al., 2006; Ştefan, 2012). They can provide new and more context-sensitive learning experiences for students. Moreover, it is now possible to extend traditional teaching methods to become more engaging and immersive for learners, making use of a range of different devices such as tablets, smartphones, electronic glasses, and head-mounted displays. These devices may allow learners to discover, visualise, engage and interact more meaningfully within the learning context (Cheng and Tsai, 2012). The learning context is now more embedded in the learning activities and is, consequently, less abstract and situated within a more holistic process. Thus, learners can become more immersed in the learning process and better informed about their performance (based on the expected learning outcomes and objectives).

Embedded computing is often considered as a hidden technology where learners can require more assistance to inspect processes and activities hidden within the technologies, making use of debugging, monitoring, and visual tools. This can range from working on a single small processor chip to more complex systems such as robotics, cars, and aeroplanes, all of which are embedded seamlessly within products and services (Suh et al., 2012). To the student, this kind of technology often has an abstract behaviour where the only information/things students can see is the final action, and they do not know how the internal processes work and communicate inside the embedded computing device to achieve the desired result. Learning the way various devices work in embedded systems is challenging and requires students to visualise how the devices work without seeing them (Anastassova et al., 2014). The same case applies to laboratory practice, where students only use various input and output nodes to manipulate how the devices work and cannot see the operations taking place inside the devices (Andujar et al., 2011). Thus, students only have a partial understanding of the various concepts they study in the classroom and the laboratory. Students are required to have an explanation of how abstract concepts work (Dede et al., 1999). Students' views should involve the invisible things that have no real representation when learning abstract concepts (Maloney et al., 2001).

However, augmented reality (AR) can overcome this issue and produce a magic-lens view for revealing hidden embedded computing activities. Linking real and virtual worlds allows augmented reality to create a reality that is both enhanced and augmented (Bronack, 2011). In addition, learners can benefit from the coexistence of virtual objects and the real environment in several aspects. First, it allows learners to visualise complex spatial relationships and abstract concepts (Arvanitis et al., 2009). Second, learners can interact with 2D and 3D synthetic objects (Kerawalla et al., 2006). Third, it allows phenomena that are non-existent or impossible in the real environment to be experienced by learners (Klopfer and Squire, 2008). Finally, it allows learners to develop important practices that cannot be developed in other learning technology environments (Squire and Klopfer, 2007). These advantages have made augmented reality one of the key emerging technologies for learning and teaching over the last five years (Johnson et al., 2010). However, the educational benefits of augmented reality not only concern the use of the technologies but also how augmented reality is designed, implemented, and integrated into learning settings, both formal and informal (Wu et al., 2013). AR might be based on technology, but it should be conceptualised beyond technology (Wu et al., 2013). These aspects can result in learners achieving a good level of knowledge and awareness of the technology, as well as higher achievement of learning outcomes. Yet, AR on its own will not improve the learning processes without first considering how to manage and represent the hidden information in a way that improves learning and teaching.

To this end, this research aims to construct a technical framework (the pedagogical virtual machine (PVM)) that provides a layered analysis of the technical and pedagogical processes that are interacting together for any given learning activity (e.g. learning about embedded computing). Besides that, the PVM makes use of AR technology to allow students to visualise the hidden information within and provides mechanisms for interacting with the underlying computing environment.

1.2 Motivation

One motivation that drives this research is related to augmented reality. The affordance of augmented reality (AR) by making the invisible visible could change how learners learn and gain knowledge about abstract concepts (Cheng and Tsai, 2012; Dunleavy et al., 2009). This could be true when adopted in appropriate curricula or activities, such as embedded computing. In computer science (CS), many of the computational processes are hidden inside the computer (Callaghan, 2012). As a human, it is often difficult for us to see these processes, as they are invisible. Often, all that we can see are the final results from a computing process, with very little information about the underlying computational process that caused the result. This is particularly true for embedded computing projects where students often will be constructing applications by assembling computing components that have a very limited user interface. Thus, a student might take an action that causes a particular result, but from an educational point of view, there may be very little explanation for how the internal processes have operated to achieve the result. The traditional approach of discovering these processes could be using traditional debugging tools or simulated environments. These approaches still separate the physical object and its activities from learners' views. Thus, adapting AR for the hidden worlds (e.g. embedded system) may enhance the learning experience. Johnson et al. (2011) stated, based on their Horizon Report, that augmented reality will be adopted in the educational sector in the following 2-3 years, as it has the ability to support/enhance teaching, learning, research, and creative inquiry. Further, in 2016, the Horizon Report expects augmented reality and virtual reality to have a large impact on teaching and learning in higher education in the next 2-3 years (Horizon Report, 2016).

The second motivation is related to the enormous growth of connected devices via the Internet, or the 'Internet of Things.' The estimated number of connected devices could reach 16 billion by 2020 (Vermesan and Friess, 2011). Callaghan

(2012) indicated the possibility of incorporating these devices in the educational sector, specifically in teaching fundamental computer science concepts and having students understand how they work. In support of Callaghan’s view, Brittain (2011) states “Children are taught physics and biology because we live in a physical and biological world. We now live in a digital world and children should be taught how it works. This will allow them to manipulate computers for their own ends”. In addition, Hatcher (2012) had the vision of teaching with objects and photographs (Figure 1-1). She believes incorporating objects and photographs into curricula by teachers would engage students, increase interest and curiosity, connect them with their environment, and provide a higher level of visual literacy and reasoning. This view could be valid with teaching computer science, especially hidden things. These things should also be treated as objects that students can explore and discover. Based on these inspiring views, this works aims to provide a mechanism that enriches learners experience while working on such hidden worlds. However, the next section describes the high-level vision of the work.

Figure 1-1 Teaching with objects and photographs, image courtesy Hatcher (2012)

1.2.1 Big vision

The inspiration for the pedagogical virtual machine (PVM) was derived from the Java virtual machine (JVM), which has the virtue of being independent of the underlying operating system. This makes the JVM portable and interoperable, as it can be deployed on any platform. JVM follows the notation of “write once, run anywhere”, that means users can write a Java code, compile it, and run it anywhere without the need to know the operating system or the platform (Venners, 1996). Thus, JVM forms a platform-agnostic interface, which gives technological companies, developers, and users the ability to make use of JVM in their platform/system.

Similarly, the pedagogical virtual adopts this principle as it provides a platform-independent interface for students and teachers to access information specifically designed for learning and teaching. As education is an ongoing process, educational technologies may change over the time. This can lead teachers to keep their knowledge and understating up to date with current technology. Teachers may want to present to students several educational technologies during the course. Thus, teachers do not want the clutter of technological understanding. Also, students want to learn the principle and do not care about the system. Thus, the vision of the PVM seeks to achieve the same principles of platform independence but for pedagogical tools (lesson design or assessment) rather than computer programming, which is the focus of JVM. As our example is based on teaching computer science, it's inevitable the question of language and hardware arises, so we have adopted an object-oriented model for our PVM, which has enabled these lower-level (hardware & software) details to be separated from the operation of the pedagogical machine, which simply concerns itself with external behaviours and messaging as a means to determine whether the pedagogical learning goals have been met. Finally, we state that "*As any computer can have a Java virtual machine; also, any computer-based system that is used for education can have a pedagogical virtual machine*". Hence, the vision is wider, but this thesis tries to address the main principle of the PVM, which could be then extended further by inspiring researchers.

1.3 Hypothesis

This research proposes the following hypothesis:

1. It is possible to create a synchronous real-time computational architecture that embeds pedagogical processes into the technology being learnt, so as to reveal the hidden computational and learning processes to the students & developers. In particular, it will be able to harness the concepts of objectifying and virtualisation as means of unifying pedagogical and computational thinking to improve learning and teaching.
2. Using the PVM framework within structured AR learning environments (specified in 1) would make the technical activities/information from the

embedded devices visible and meaningful to the student. This will lead to improved learning outcomes for learning activities and give the learners greater awareness of the activities inside structured AR with PVM learning environments.

3. That such structured PVM with AR learning environments (specified in 1) would enable learners to acquire new knowledge more quickly and with fewer misunderstandings.
4. That using structured PVM with AR learning environments (specified in 1) would provide assistance that reduces the load in learning.
5. That using structured PVM with AR learning environments (specified in 1) would increase students' enjoyment, perceived competence and usefulness while performing laboratory hands-on-activities.
6. That using structured PVM with AR learning environments (specified in 1) would increase learners' curiosity, as compared to traditional laboratory learning environments.
7. That using structured PVM with AR learning environments (specified in 1) would be preferred over traditional learning environments.
8. That learners' user experience while using structured PVM with AR learning environments (specified in 1) would show acceptance of PVM system.

1.4 Contributions

This thesis presented the following contributions:

1. Created a novel computational model and architecture (PVM model) for extracting pedagogical and computing activities within structured learning environments. This model was complemented with a learning design strategy

for constructing learning activities that involve both computational and learning processes (PVM model – *Chapter Three*).

2. Developed a pedagogical explanation approach to engineering laboratory based learning activity that utilises a layered architectural dichotomy. This was complemented with the augmented reality technology to make both computational and learning processes visible to learners and developers by constructing a robust meaningful view of invisible processes in the world (PVM model – *Chapter Three*).
3. A prototype system that implements the PVM model components, employing different learning activities task based on a pedagogical framework (The PVM with AR- *Chapter Four*).
4. Evaluating the pedagogical effectiveness of PVM with AR prototypes through two user studies comparing them to equivalent traditional approaches. The first study involved constructing and exploring embedded computing learning tasks targeting the lower level of the pedagogical framework (*Chapter Five*), whereas the second study addressed the challenge of controlling real-time systems focusing on the high level of the pedagogical framework (*Chapter Six*).

A secondary contribution is included as follows:

- Designing AR user interfaces driven by the structured information architecture to support complex learning tasks.

1.5 List of publications

Part of the contributions described in this thesis have been published and presented in the following publications:

1) Elliott, J.B., Gardner, M. and **Alrashidi, M**., 2012. “Towards a framework for the design of mixed reality immersive education spaces”. *2nd European Immersive Education Summit*, pp.63-76.
2) **Malek. Alrashidi**, V. Callaghan, M. Gardner & E. Jennifer, “ViewPoint: An Augmented Reality Tool for Viewing and Understanding Deep Technology*” on 1st Workshop on The Cloud of Things (CoT13’s), July 16-17 Athens, Greece, 2013.*
3) **Malek. Alrashidi**, V. Callaghan, M. Gardner & E. Jennifer, “The Pedagogical Virtual Machine: Supporting Learning Computer Hardware and Software via Augmented Reality” on *3rd European Immersive Education Summit (EiED’12)*, London, UK, 2013.
4) **Malek Alrashidi**, Vic Callaghan, Michael Gardner *“An Object-Oriented Pedagogical Model for Mixed Reality Teaching and Learning”,* on Intelligent Environments 2014, Shanghai Jiaotong University, China, 2-4 July 2014
5) **Malek. Alrashidi**, V. Callaghan, M. Gardner & E. Jennifer, “Structured Learning Activities in Embedded Computing Using a Pedagogical Virtual Machine (PVM)” on iLRN conference 2015, 13th - 14th July 2015 Prague, Czech Republic
6) Ahmed ALZAHRANI, Michael GARDNER, Vic CALLAGHAN, and **Malek ALRASHIDI**, “Towards Measuring Learning Effectiveness considering Presence, Engagement and Immersion in a Mixed and Augmented Reality

Learning Environment", on iLRN'15, Prague, Czech Republic, 15-17 July 2015.

7) **Malek Alrashidi**, Ahmed Alzahrani, Michael Gardner, and Vic Callaghan, "A Pedagogical Virtual Machine for Assembling Mobile Robot using Augmented Reality." On Proceedings of the 7th Augmented Human International Conference (AH '16), Geneva, Switzerland on February 25 - 27, 2016.
8) **Malek Alrashidi**, Michael Gardner, and Vic Callaghan, "Evaluating the Use of Pedagogical Virtual Machine with Augmented Reality to Support Learning Embedded Computing Activity", the 9th International Conference on Computer and Automation Engineering, ICCAE '17, February 18-21, 2017, Sydney, Australia.
9) **Malek Alrashidi**, Khalid Almohammadi, Michael Gardner, and Vic Callaghan, "Making the Invisible Visible: Real-Time Feedback for Embedded Computing Learning Activity using Pedagogical Virtual Machine with Augmented Reality", the 4th International Conference on Augmented Reality, Virtual Reality and Computer Graphics SALENTO AVR 2017, Ugento (Lecce), Italy, June 12-15, 2017.

Also, the proposed PVM model was described in the following journal paper:

- Gardner, M. and J. Elliott. "The Immersive Education Laboratory: understanding affordances, structuring experiences, and creating constructivist, collaborative processes, in mixed-reality smart environments." EAI Endorsed Transactions on Future Intelligent Educational Environments, 2014. 14(1): p. e6.

1.6 Thesis Outline

The remaining chapters in this thesis are organised as below:

- Chapter two presents the concept of mixed reality and augmented reality. It focuses on augmented reality, and its affordances, issues, and uses in an educational setting, particularly focusing on the use of augmented reality to support learning and teaching understanding of abstract concepts in STEM laboratory education (e.g. making the invisible visible of STEM concept). The section identifies some of the challenges faced by learners when performing learning activities that involve dealing with hidden worlds and proposes the use of AR in a structured learning environment, establishing possible benefits of using such environments over the use of current alternatives for hands-on engineering-style construction activities. In addition, this chapter provides insight on different smart technology that can be incorporated with AR, such as mobile learning, intelligent environments, and the Internet of things (IoT). In regards to the pedagogical approach, it introduces the concept of constructivism, deconstructivism, and learning objects, a core principle used in constructing learning activities that unify technical and learning processes within structured learning environments.
- Chapter three introduces the pedagogical virtual machine model (PVM), a conceptual architectural model for embedding learning processes within technology. This chapter presents the principle of using the PVM in lab-based computing engineering activity conducted by learners and developers. The PVM model consists of a layered approach that aims to provide pedagogical explanations regarding the technology being learnt. The PVM layers are data, aggregation, pedagogy, and user interface. At each layer, the PVM

incorporates augmented reality to visualise the computing and learning activities. Finally, the chapter proposes a design strategy for constructing learning activities that can utilise the PVM affordances.

- Chapter four starts by describing the implementation of the proof-of-concept PVM with AR systems, which is based on the conceptual architectural model, defined in the previous chapter. Additionally, this section explains the implementation of the physical objects as well as implementing two learning activities based on the PVM design strategy proposed in chapter four.
- Chapter five discusses the different evaluation techniques used for augmented reality user studies before explaining the specific method used for the evaluation of the first embedded computing learning activity scenario (assembling and exploring learning activity) that utilises the proof-of-concept PVM with AR system. This section describes the experimental design and strategy used in the evaluations. Later, the chapter presents statistical results for the user evaluations, together with an in-depth analysis.
- Chapter six starts by explaining the specific method used for the evaluation of the second embedded computing learning activity scenario (designing behaviour based robotics activity) that utilises the PVM with AR system. This section describes the experimental design and strategy used in the evaluations. Then, the chapter presents statistical results for the user evaluations, together with an in-depth analysis.
- Chapter seven discusses the results/findings of both evaluations (chapters five & six) and their wider consequence to the research study.
- Finally, chapter eight concludes the thesis by summarising the achievements of the thesis, discussing main educational and technological issues that arose

from the research. It then highlights the future work and finishes the thesis by drawing an image of the future of the field.

Chapter Two

2 Background and Related Work

"Experience is the teacher of all things."

- Julius Caesar (100 BC, 44 BC)

This chapter provides an overview of related topics that contribute to this research, such as the use of mixed-reality technologies, augmented reality, learning theory and the concept of smart things (intelligent-environment). In general, it examines the use of these technologies from technical and pedagogical points of view and their use in education. Specifically, it looks at the current methods of mapping physical object activities within structured learning environments to improve learning and teaching.

2.1 Mixed Reality

Mixed reality is considered one of the most advanced methods to connect the real environment with virtual spaces (Milgram and Kishino, 1994). It helps solve challenges pertaining to the physical/virtual in one shared environment. In addition, it ties both worlds, virtual and physical, together to form multiple views to people. Therefore, it allows people to switch from one world to another or to view both worlds at the same time but with different granularities of virtual or physical (whether physical space is dominant or virtual space is dominant). Milgram has divided mixed reality into four subsets by using a Reality-Virtuality Continuum (Figure **2-1**). Milgram and Kishino (1994) stated that "the most straightforward way to view a mixed reality environment, therefore, is one in which real world and virtual world objects are presented together within a single display, i.e., anywhere between the extremes of the Virtuality Continuum". In this case, mixed reality (MR) is considered a spectrum that connects physical environments not present in any kind of virtual representation to virtual ones, facilitating the co-existence of computer-generated and physical elements in the real world. Its potential could be determined by the possibility of facilitating reality, thus making invisible things visible (Pastoor and Conomis, 2005). Nevertheless, MR potential can also be determined by its natural synthesis and modification of physical laws, thus governing reality and implementing metaphors, such as auditory, visual and haptic, that could not exist in the physical world (Ellis and Bucher, 1994).

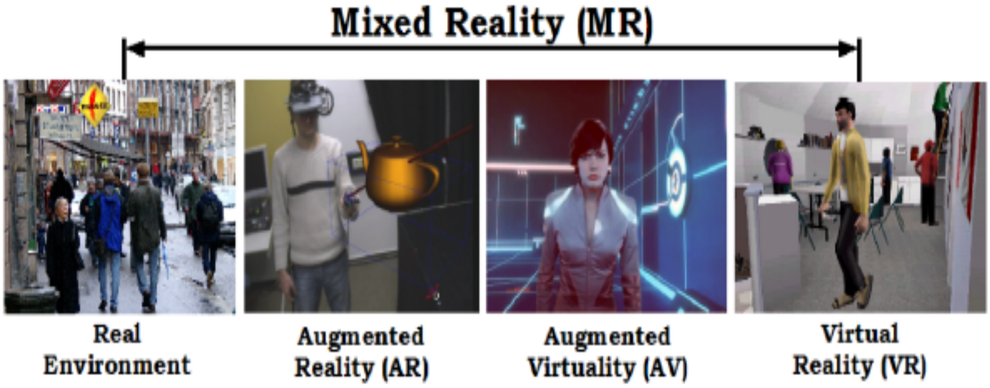
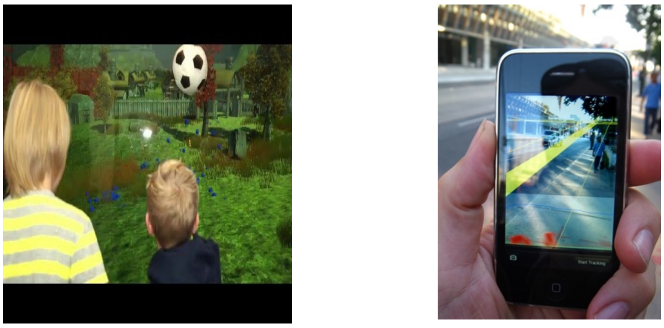


Figure 2-1 Milgram's Reality-Virtuality Continuum

In the Reality-Virtuality Continuum's scope, augmented reality (AR) and augmented virtuality (AV) develop MR. As illustrated in Figure **2-2**a, the AV primary environment is virtual and improved by the addition of the physical world's data and objects. It is worth noting that a virtual environment involves a computer-generated interactive space based on non-visual and visual mechanisms, such as auditory and haptic, that convince users to believe they are immersed in synthetic space (Ellis and Bucher, 1994). Additionally, Wann and Mon-Williams (1996) suggested the existence of a three-dimensional virtual environment (3D VE) within the Reality-Virtuality Continuum's scope. A 3D VE is an environment used to capitalise on natural human aspects of perception through the extension of visible information in three dimensions. Dalgarno et al. (2002) further noted that 3D VE has three major characteristics:

- The three dimensions illusion,
- Physical and smooth temporal changes, and
- A high interactivity level.

The AR illustrated in Figure 2-2b shows an environment with a physical world enhanced through the addition of computer-developed objects using computer methods of vision to make them look as if the objects and physical world co-exist within one environment (Pastoor and Conomis, 2005). Therefore, AR provides reality to the environment and not replacement. In this case, AR could provide necessary information that cannot be detected directly by users, helping them execute world tasks. In addition, AR can enable users to understand a complex scenario (Azuma, 1997).



(a) Augmented Virtuality

(b) Augmented Reality

Figure 2-2 Mixed Reality

AR technology has the capability to supplement the physical object with virtual information rather than replacing existing teaching and learning material. Thus, the study will concentrate more on AR in Milgram's Reality-Virtuality Continuum and will indicate related studies conducted on the other dimensions when applicable. The following section gives an overview of augmented reality technology and its affordances and use in education.

2.2 Augmented Reality

2.2.1 Definition

The term "augmented reality" simply refers to the real world (environment) that is augmented with virtual objects (Milgram and Kishino, 1994). Azuma (1997) defined AR as a system consisting of three fundamental aspects: a combination of both the virtual and the real world, real-time interactions and 3D registration of virtual and real objects. Klopfer and Squire (2008) defined AR as "a situation in which a real world context is dynamically overlaid with coherent location or context sensitive virtual information". Also, AR can be defined as an approach centred on the overlay of virtual objects in a real-world context and having the ability to induce in users feelings of sub-immersion through facilitated interactions between the virtual and the actual world (Uematsu and Saito, 2008). In the context of this thesis, AR can be defined as a system in which physical things around students are blended with real-time virtual information in order to support students' perceptions of the surrounding environment.

2.2.2 Brief History of Augmented Reality

The display technology for AR, which blends the real and the virtual environment, can be divided into two main categories (Table 2-1 gives a brief history of AR). The first category is visual display, which consists of three ways to visually augment reality: video see-through, optical see-through, and projective. The second category is display positioning, which refers to the display position between the real world and the viewer: handheld, spatial, and head-mounted.

AR has several inputs and tracking devices. For inputs, Reitmayr and Schmalstieg (2003) used gloves for their mobile augmented system, whereas wristband wireless devices were used by Feldman et al. (2005). Modern smartphones can be used as a pointing device; for example, the Google Sky map allows users to direct their phones to the sky in order to know the names of stars and planets (Carmigniani et al., 2011). The requirements of this type of input device depend on the application requirements. For example, touchscreen input can be used for handheld displays. On the other hand, tracking technologies contain several devices including cameras, sensors, GPS, accelerometers, ultrasound and optics. A comparison has been made between these technologies according to their range, setup time, precision, time and environment (Carmigniani et al., 2011).

Table 2-1 History of Augmented Reality

Year	Augmented Reality Development
1968	Ivan Sutherland invented the first prototypes, which were based on a head-mounted display (Sutherland, 1968).
1985	Users interacted with virtual objects inside a room in a system called

	Videoplace created by Krueger et al. (1985).
1989-1993	AR coined by Caudell and Mizell, who developed a system that aided employees to bring together wires and cables of an aircraft (Caudell and Mizell, 1992; Janin et al., 1993).
1993	The first function of an AR system called Virtual Fixtures was developed by Rosenberg (1993), which showed its advantages for human performance.
1999	ARtoolKit, a library used to build AR applications, was developed by Kato and Billinghurst (1999).
2000	ARQuake, an AR outdoor mobile game, was developed by Thomas et al. (2000).
2009	Sixth sense prototype produced by MIT was based on wearable AR devices (Mistry and Maes, 2009).

Over recent decades, augmented reality has been used in several areas, such as for military and medical purposes, advertising and commercials, entertainment, education, and mobile applications. In the military, it has been used for head-up displays and helmet-mounted sight “to superimpose vector graphics upon the pilot’s view of the real world” (Azuma, 1997). In medicine, it has been used to help doctors view the patient's status in the form of 3D images and to give doctors an enhanced version of x-ray vision to examine patients (Chastine et al., 2007). Furthermore, in advertising and commercials, it has been used as a marker to allow users to see advertisements through their webcams.

2.2.3 Augmented Reality Technologies

The development of handheld computers and smartphones has granted new possibilities for augmented reality (Martin et al., 2011; Squire and Jan, 2007). These handhelds offer mobility, which could leverage the validity/authenticity of the learning environment and increase the learner's interactions with others (Klopfer and Sheldon, 2010). Dunleavy et al. (2009) indicated that the implementations of augmented reality result in student motivation. Both teachers and students showed more engagement, especially when using handhelds, adopting roles, negotiating meaning within activities, solving authentic problems and performing physical tasks (Dunleavy et al., 2009; Dunleavy and Simmons, 2011). Making pervasive AR systems is possible by using mobile devices (Broll et al., 2008). Handheld computers with location-based technology, such as GPS, allow pervasive augmented reality systems to be run on these technologies, as opposed to head-mounted displays.

Moreover, Mobile AR is recognised as being an organic platform geared toward various 'killer apps', as they are named. It has been highlighted by Wagner and Schmalstieg (2006), for instance, that an interactive AR museum can be described as 'a virtual media that annotates and complements real-world exhibits'. In this same vein, a training application is introduced by Träskbäack and Haller (2004), which enables the staff of oil refineries to observe instructional diagrams positioned on top of the tools being utilised and learned. A number of other applications are known, such as multi-user AR on handheld devices(Wagner et al., 2005), equipment maintenance (Feiner et al., 1993), and document annotation (Rekimoto and Ayatsuka, 2000), amongst others. Regardless of the application, however, there are a number of characteristics in common. For example, all of these applications depend on vast distributed and dynamic data, and there is a need for all links between relevant data and recognisable visual targets to be maintained. Such links will ultimately alter through application development or with the growth of the underlying data.

Essentially, there is the need for the presence of a presentation layer, which describes how data can be rendered as virtual media. In some instances—although this rests on the data's nature—it may be sensible to render various mixes of icons, images, texts or 3D objects. The exact conversion from information to virtual content ultimately rests on the type of application. Various users adopting different mobile devices could be able to collaborate and share data. This, therefore, suggests the requirement of a central data store with the ability to monitor the actions of users as well as the system state overall (Mooser et al., 2007). Schmalstieg et al. (2007) proposed a 3-tier data model for managing data in a mobile augmented reality application. The first layer was a database, whereas the second layer linked the database and application by translating raw data from the database to a specified data structure. The third tier contained all the applications. In addition, the second tier hides data from presentation so applications did not have to understand the data details. Application types were derived from basic abstract types, such as SpatialObjectType and ObjectType, which were predefined. Data storage and presentation layers were linked, ensuring virtual representations were consistent with the monitored technology.

Nicklas et al. (2001) also proposed a three-layer model that consisted of a client device layer, server layer and federation layer. All system resources were stored in the server layer, which could come in different forms, e.g. geographical data, users' location or virtual objects. A top-level object Nexus Object was designed, from which all objects, such as sensors, spatial objects and event objects, could inherit. The federation layer provided transparent data access to the upper layer by use of a register mechanism. It decomposed queries from the client layer and then dispatched them to registers for information access. It guaranteed consistent presentation, even if data servers supplied inconsistent data. The model increased access delays because of the delegation mechanism and separated underlying data operations from the client layer. In addition, multiple copies of the object on different servers caused data inconsistency.

Similarly, Schmalstieg and Hesina (2002) produced a 4-layer data model for mobile augmented reality. The lower layer was a dynamic peer-to-peer system that allowed both communication and connectivity services. The second layer provided general mobile augmented reality functions such as tracking, sensor management and environmental presentation. The third layer included a high-level functional module that was composed of sub-layer components, which offered application related functions for the higher layer that interacted with users. Object identifiers and their types were used to represent the virtual objects; these were bound to a table data structure that contained linking information. In addition, to describe object relationships, a data structure was used as well as a special template to store representative information.

2.2.4 Augmented Reality in Education

The focus here is to provide additional information students might not have access to or might be missing. An example of the effective application of AR to teach students and provide them with additional information is illustrated in anatomy education, which does require not only expertise and huge expenses but also a lot of effort to effectively deliver (Blum et al., 2012).

According to Munnerley et al. (2015), individuals always strive to look for ways to simultaneously bring together the various elements in their physical and digital contexts and integrate them in new and creative ways. It is in this context the AR becomes very resourceful by providing alternative perspectives. These perspectives may be alternative physical reality differentiated by time or place or that which is created by the thoughts of others. Through the use of AR, one can expand the mind from its initial state of lack of some beliefs or memory to include perspectives that are much wider and more advanced (Clark and Chalmers, 1998). Simultaneous access to many perspectives is provided by AR (Munnerley et al., 2015). These perspectives include those that have been generated both inside and outside of the person's mind. The viewer can then combine these perspectives to view an object in an integrated way. Munnerley et al. stated that augmented reality can create a shared story, a collective memory or a meta-narrative. This capacity for associating the various individual stories and putting them together into a shared experience network gives ways to new channels that create a common understanding and educational liberation.

A report by Johnson et al., (2011) showed that AR has important pedagogical applications. As such, studies conducted by several authors examine the effects AR can have on various aspects of science education, including laboratory work (Andujar et al., 2011; Benito et al., 2014), conceptual change, ecological preservation (Lin et al., 2011), inquiry-based learning (Squire and Klopfer, 2007b), spatial ability (Martín-Gutiérrez et al., 2010) and scientific argumentation (Squire and Jan, 2007). The majority of these studies found that learners regard AR in a positive way. Some of the learners were satisfied with AR; others found it useful, and some improved in their learning (Cheng and Tsai, 2012).

Several possibilities can be realised by using AR in education (Wu et al., 2013). Wu et al. (2013) provide the advantages of AR in education: First, AR makes it possible to visualise relationships that are complex and concepts that are hard to understand (Arvanitis et al., 2007; Shelton, 2002). Secondly, AR can facilitate the timely presentation of information at the right place, contributing to reducing time searching for such information, reducing errors, and improving the ability to memorise and recall information (Cooperstock, 2001; Neumann and Majoros, 1998). Additionally, by using AR, there is the possibility of experiencing phenomena that may not be experienced in real world situations (Neumann and Majoros, 1998; Klopfer and Squire, 2007).

Billinghurst and Dünser (2012) indicated that through AR, complex phenomena could be easily understood. This is because AR provides unique visual and interactive experiences that integrate both virtual and real information and, thus, facilitate the communication of abstract concepts to learners. This capability allows the superimposing of virtual graphics over real objects by designers, thus making it possible to manipulate digital content in a physical manner and interact with it. As such, spatial and temporal concepts are demonstrated more effectively in addition to the relationship between virtual and real objects. For example, through reading in combination with a 2D picture, a student can theoretically understand the position of the earth relative to that of the sun. A much better and practical understanding can be gained by using a 3D visual system. AR has made it possible to animate dynamic processes to provide a direct, tangible interaction that makes it possible for learners to interact intuitively with digital content.

From a pedagogical perspective, AR provides a student-centred approach to learning in addition to a flexible space for opportunities for learning (Munnerley et al., 2015; Novak et al., 2012). AR makes this possible by taking learning wherever the student goes by decoupling it from the traditional classroom, lecture halls and labs. AR makes it possible to learn from home, at the workplace or even on the public transportation system. Moreover, AR is developing more interactive applications that make learners become co-creators and critics, thus leaving a record of their learning based on their experiences. In the framework of teaching and learning, AR plays a general role. Several important elements of traditional learning, such as reflection, a questioning attitude, integration, critical thinking, and real learning goals, are all promoted by augmented reality. Augmented reality provides various opportunities for learning and teaching:

a) Visualisation (subject to user manipulation),
b) Mobility,
c) Content (general and student constructed),
d) Alternative perspectives,
e) Contrasting and comparing the various perspectives, and
f) Integration of multiple perspectives.

As reported by Munnerley et al.(2015), research on the application of AR in the educational sector has mainly focussed on the visualization and mobility to provide flexible learning opportunities by using handheld and portable devices to deliver information and on the visualization of information learners may have challenges in accessing because of financial or physical constraints. It is time that opportunities (c)-(e) are given the required attention. However, this may require the consideration of the principal purpose, which is to come up with student-centred learning experiences where there is the connection, integration, construction, and deconstruction of the students’ own meaning based on their experiences.

2.2.5 Augmented Reality Affordances

AR is known to comprise of a number of features and affordances falling into the following six different elements (Wu et al., 2013). With this noted, AR is recognised as facilitating the following:

- Bridging formal and informal learning,
- Learning content in 3D perspectives,
- Learners' sense of presence,
- Immediacy and immersion,
- Ubiquitous, situated and collaborative learning, and
- Visualising the invisible.

Regarding the use of 3D synthetic objects, AR facilities this, enabling the visual perception of the target environment or system to be more developed, as highlighted by Arvanitis et al. (2007). Another element is linked to the adoption of handheld computers in AR, with the facilitation of such technology enabling ubiquitous, situated and collaborative learning seen to be able to develop learning within a real environment (Broll et al., 2008; Dunleavy et al., 2009). Moreover, it has been recognised by Bronack (2011) that AR, along with other immersive media, has the potential to deliver affordances in terms of immediacy, immersion and presence, with AR delivering a mediated space that provides learners with the feeling they are with others, thus reinforcing and enhancing the learners' community of students (Squire and Klopfer, 2007). In the same regard, it is further emphasised that an AR system has the capacity to comprise and deliver both verbal and non-verbal cues and feedback, which together aid in establishing and maintaining a sense of immediacy in students (Kotranza et al., 2009). Considering that immediacy is fundamental in terms of developing learning from an affective perspective, AR, when bringing together students, data or virtual objects within a real environment, can significantly enhance immediacy.

One further element of AR affordance is the ability to facilitate the visualisation of invisible events or concepts through superimposing virtual information or objects onto physical environments or objects (Arvanitis et al., 2007; Dunleavy et al., 2009). Essentially, when taking this into account, AR systems could provide students with support and assistance in terms of enabling and helping them visualise unobservable phenomena or abstract scientific concepts through the adoption of virtual objects, including—but not limited to—molecules, symbols and vectors. For instance, through the application of augmented chemistry, students are able to choose chemical elements, add them into a framework of a 3D molecular model, and revolve and pivot the model (Kotranza et al., 2009).

In addition, AR is known to bridge the gap in relation to learning environments, both formal and informal. For instance, AR, as well as a number of other technologies, was employed in the CONNECT initiative technology with the aim of developing a virtual science thematic park environment (Sotiriou and Bogner, 2008). Two different modes were encompassed: the school mode and the museum mode. Various scenarios considered in the environment comprise both conventional and virtual field trips to museums, curricular activities carried out both before and after the visit, and modelling and experiment activities. In this case, science learning within the school was linked with the learning-related experiences associated with conventional and virtual museum visits, facilitated through the use of AR. This helped develop the experiments, models and visualisations of students. Nevertheless, Wu et al. (2013) indicated that AR may not be the only system delivering such elements; some may be provided by other environments or systems in the context of other relevant and comparable concepts and technologies. Therefore, in order to highlight the affordances of AR, it is essential to examine the way in which AR could be positioned within various instructional approaches so as to fulfil teachers educational aims and goals (Bronack, 2011).

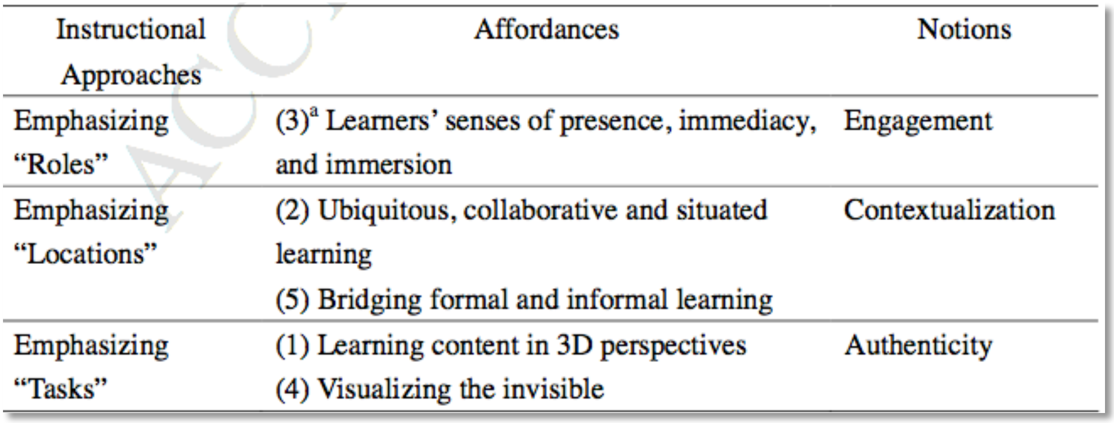
Finally, Dunleavy et al. (2009) stated in their literature review, which mainly concentrated on augmented reality for learning in formal and informal learning environments, that the usage of mobile and context-aware technologies such as tablets and smartphones allows users/students to interact with the digital information embedded in the physical world. In addition, they highlighted the affordances and limitations for AR in three aspects: teaching, learning and instructional design. The common affordance of AR is the competence to provide to a group of learners multiple incomplete yet balancing perspectives on a problem situated within the physical world (Dunleavy et al., 2009; Facer et al., 2004; Perry et al., 2008; Squire et al., 2007; Squire and Jan, 2007). This is a result of the one-to-one ratio between the device and the student within such environments, which enables each student to interact with the GPS device in order to contribute to the activity/task. In addition, this affordance allows educators to use one of the appropriate collaborative pedagogical approaches with the design experience approach, such as jigsaw and differentiated role-play. AR has the potential to provide multiple points of view of the same object, which can aid learners to go further than the information available to them (Milgram and Kishino, 1994). Combining these multiple viewpoints with the environment and placing them within a problem-based augmented reality can afford teachers the ability to augment the physical world with digital information and transform it into a place that allows students to manipulate, observe and analyse. In addition, they allow learners to explore, construct and manipulate virtual objects (Dalgarno and Lee, 2010).

2.2.6 Augmented Reality Issues

The issues related to AR are categorised under three major aspects: technology, pedagogy and learning (Wu et al., 2013). The head-mounted display(HMD) is one of the technologies that has some issues such as their cumbersome and expensive design, which causes discomfort and poor depth perception (Kerawalla et al., 2006). This issue can be overcome with the new generation of HMD, such as glasses, or the new portable technologies for AR, as they are more comfortable and enhance the immersive capacity of both sense and presence (Fiorentino et al., 2014).In addition, stability (Squire and Klopfer, 2007) and interfacing between multiple devices (Klopfer and Squire, 2008) are issues that appear when integrating several portable hardware and software devices. Learners can face difficulties in the absence of a well-designed interface to guide their actions in real-world environments and to allow for the smooth flow of information from one device to another as well as for the transfer from the virtual to the reality. Wu et al. (2013) also pointed out the high risk of failure when multiple devices are used.

Instructional design is a pedagogical issue for AR. Wu et al. (2013) posed the important question of "How *should the information be distributed and flowed between two realities and among different devices?"* when designing the learning activity and augmented reality system. Furthermore, Klopfer and Squire (2008) asked how one may "*balance competing drives for individuality with distribution and decentralized information flows with guided educational activities may be tensions central to the platform*". Wu et al. (2013) suggested empirical evidence is needed as well as a set of design guidelines based on learning theories, such as situated learning, to help teachers and designers eliminate this tension. To design an effective learning environment, both designers and researchers can precisely describe not only the technologies and their uses but also their affordances from the learners' view (Elliott et al., 2012).

In terms of the learning process, learners could be cognitively overloaded in an AR learning environment by encountering a huge amount of information, the number of devices they have to use, and the difficult tasks that the learners need to accomplish (Wu et al., 2013). This leads students to multitask in an augmented reality environment. Dealing with a complex task and unfamiliar technologies often results in confusion and feeling overwhelmed when learners engage in a multiuser AR simulation (Dunleavy et al., 2009). In addition to supporting this claim, many researchers stated students can be cognitively overloaded in three main aspects: 1) the task's complexity; 2) both scientific enquiry process and navigation (Klopfer and Squire, 2008); and 3) making decisions as a group (Perry et al., 2008). Furthermore, Perry et al. (2008) indicated one of the key instructional issues related to managing the level of the complexity is that AR designer experience has attempted to reduce cognitive load by simplifying the initial structure and increasing the complexity as the experience progresses. Wu et al. (2013) summarised the important notions with their affordances and instructional approaches in their paper that suggested further investigation of how AR environments support learning and teaching by reconceptualising these notions (Figure **2-3**).



Instructional Approaches	Affordances	Notions
Emphasizing "Roles"	(3)^a Learners' senses of presence, immediacy, and immersion	Engagement
Emphasizing "Locations"	(2) Ubiquitous, collaborative and situated learning (5) Bridging formal and informal learning	Contextualization
Emphasizing "Tasks"	(1) Learning content in 3D perspectives (4) Visualizing the invisible	Authenticity

Figure 2-3 Possible Alignments of Instructional Approaches, Affordances and Notions in Education (adapted from Wu et al. (2013))

2.3 AR in Science, Technology, Engineering and Mathematics Laboratories (STEM Education)

Blending both mixed reality and laboratories is another application of AR. By superimposing virtual elements on devices, learners could interact and manipulate both real and virtual objects. By combining these AR technologies, pervasive learning can be enhanced by 3D or virtual objects, physical models, remote laboratories and computer simulations (Broll et al., 2008; Dunleavy et al., 2009).

Wu et al. (2013) posed the important question of how AR technology can be used for educational purposes. Dede (2009) stated that AR technologies aid students to engage in true exploration in the real environment, and virtual objects such as text, video and pictures are all supplemented with elements for the learner to discover the real-world surroundings. The most frequent uses of AR are to explain spaces with the superimposing of location-based information (Johnson et al., 2010). Second, the usage of AR can further extend to the combination of the real world and digital learning resources. Learners can experience the scientific phenomena that are not possible in a real environment, such as chemical reactions.

The use of AR in education can make it possible to construct knowledge in an active and autonomous manner and, thus, facilitate practical learning. Furthermore, Zhong et al. (2003) note that learner motivation can be improved through AR because it raises the enthusiasm of students because of the interaction with new technologies. These benefits of AR in general education can be applied to technical and engineering education and specifically to courses in embedded systems (Anastassova et al., 2016).

Kaufmann and Schmalstieg (2003) developed an AR system for mathematics and geometry education. The 3D geometric construction system aimed to improve the spatial abilities of learners and to maximise the transfer of knowledge in real/practical contexts. However, an informal evaluation showed the students were stimulated to use it and needed very little time to familiarise with it and apply it in practical settings. The system had several problems, such as fatigue from its use and eye-hand coordination minus haptic feedback.

Another example of the AR in education is the use of tangible interfaces and AR models of 3D objects (Chen et al., 2011). In the tangible interfaces, physical objects are coupled to a digital source of information. The AR models are used in engineering courses containing graphics so as to promote the students' understanding of the relationship that exists between 3D objects and their projection. The tangible interfaces system was tested with 35 students who were majors in engineering. From the tests, the students' performance and ability to transfer 3D objects into 2D projections significantly improved. Higher rates of engagement with the AR models were also realised.

Additionally, AR can be applied in teaching courses in embedded electronics (Anastassova et al., 2016). Learning the way the various devices work in embedded systems is challenging and requires students to visualise how the devices work without actually seeing them. The same case applies to laboratory practice where students only use the various input and output nodes to manipulate how the devices work and cannot see the operations taking place inside the devices. As a result, students only have a partial understanding of the various concepts they study in the classroom and in the laboratory. The use of augmented reality in engineering education aims at overcoming challenges such as those described above, particularly in the first years of studies in computer engineering. For this reason, it was proposed by Müller et al. (2007) and Andujar et al. (2011) that an AR system be used for enhancing the students' interactions with the various laboratory works.

Andujar et al. (2011) considered a digital control system design based on a field-programmable gate array (FPGA) development board. Here, AR is used to demonstrate how the various activities carried out in the lab can be performed in the same way it is done in the workplace. The system provides physical contact and, thus, reduces cases of student discouragement because of lack of it. The study used 10 teachers and 36 students in evaluating the system. For both the students and teachers, there were higher engagement, greater motivation, and significant improvement in learnability of abstract concepts when compared to the traditional learning methods. In this AR prototype and many others, designers focussed much on the visual aspects of embedded electronics. In contrast, a unified embedded engineering learning platform (E2LP) AR system has been developed which focusses on other factors beyond the visual aspects by coming up with a multisensory AR system for teaching electronics (Anastassova et al., 2016; Miodrag Temerinac et al., 2013). The E2LP system has a camera that takes a video of the board, which is then projected on a touchscreen that is supported like an electronic lamp (**Figure 2-4**). Additionally, the system has a tactile pointer used in showing its position on the board. With the information from the camera and the pointer, an enhanced visualisation of the real view is displayed by the AR software on the tactile screen. From the view and with the aid of the pointer and the camera, students get quick access to the components of the board and their specifications and work on them. The AR software can be explained from the pedagogical point of view, which structures it into three primary levels: exercises, problems and projects.

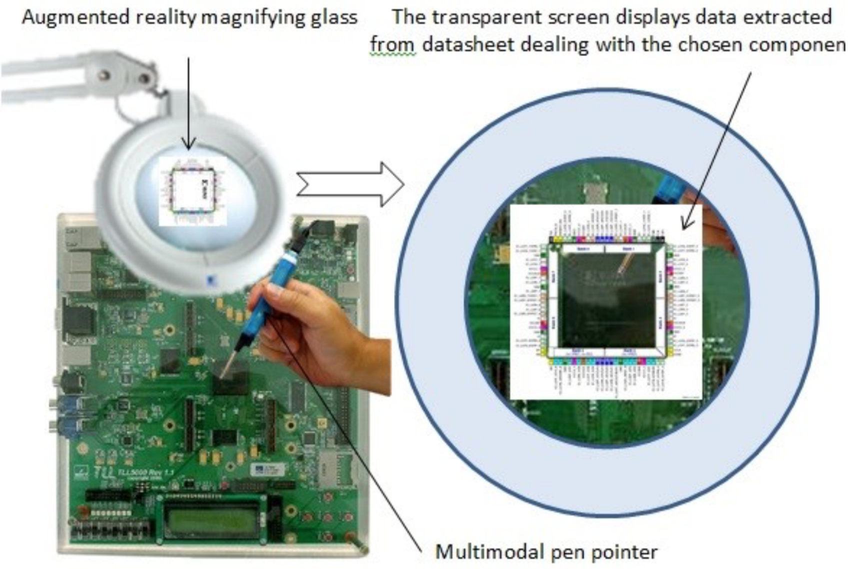


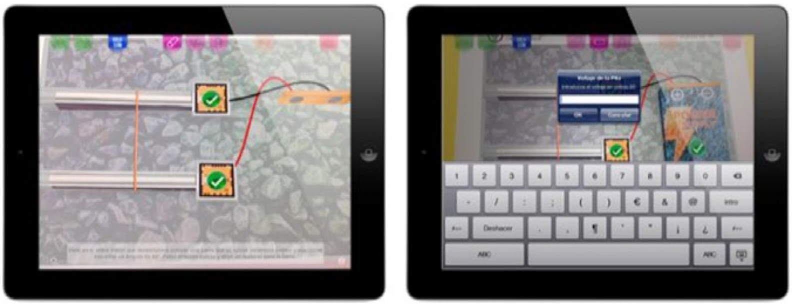
Figure 2-4 E2LP AR system (Curtsey to M. Temerinac et al.,(2013))

- **Exercises**: In this AR case, exercises are the basic tasks that have well-structured solutions supporting the superimposing of the various hardware components in a visual way. Additionally, students have to resolve the various tasks before starting a new course. For this reason, AR is an important tool that can be used to facilitate the engagement of students and improve their understanding of the various concepts.

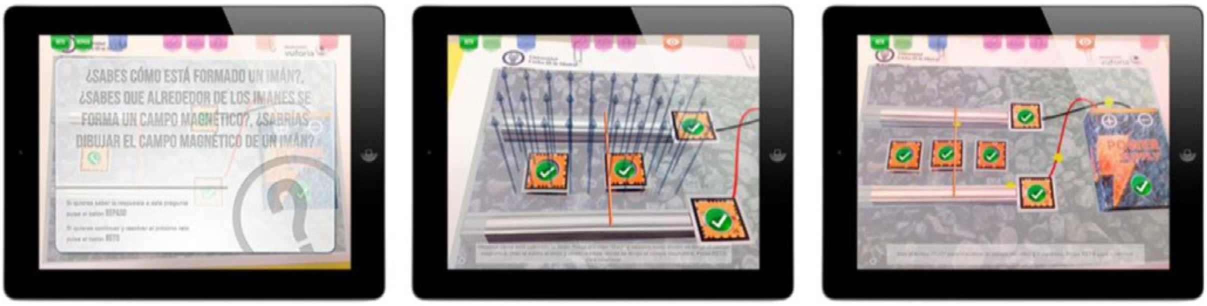
- **Problems**: The AR software has several open-ended tasks with solutions that not only differ but also use different methodologies. Problems are challenging in comparison to exercises and in terms of the information displayed on the AR system. However, various clues are provided by the AR system to enable the solving of the different tasks.
- **Projects**: Projects are more challenging than exercises and problems because they require students to come up with their own projects, define the objectives of the projects, and find sources for the various components needed to complete the project. As such, a project does not have a pre-defined path students can follow. The AR software can be instrumental in providing information about the various resources the students need to use.

Chan et al. (2013), in a different study, proposed and evaluated the LightUp design, which is an augmented reality learning system for electronics. Several parts, like wire and bulb as well as motor and microcontroller, make up the LightUp. The learner, in forming circuits, needs to have these parts connected magnetically. The implementation of LightUp is in the form of a mobile application that makes "informational lenses" utilising computer recognition to identify electrical parts available and to supplement the image with visualisations by making the invisible circuit visible. This system helps children with real time learning, understanding, and constructing of circuits through simulation. The disadvantage of this study is its reliance on using a simulation in extracting information for the learning activity while failing to utilise physical objects.

Similarly, Ibáñez et al. (2014) designed and evaluated an augmented reality learning application to learn electromagnetism concepts. This application allowed students to manipulate 3D shapes and emulate the circuit elements. Each element was attached a fiducial marker to make it possible for recognition. Each element was associated with a particular learning material of problem-solving for the manipulation of the students (Figure **2-5**). Such aided the students in discovering the circuit's behaviour or in the visualisation of the electromagnetic forces. Their research contributed to a much better comprehension of the effects of AR on learning results, specifically those that require electromagnetic invisible forces understanding. However, this study was not focused on a real physical object and did not look at revealing real-time data nor at inspecting the communication process inside the object; it used AR to help simulation for building the circuit step-by-step.



CABLE **BATTERY**



¿SABES CÓMO ESTÁ FORMADO UN IMÁN?, ¿SABES QUE ALREDEDOR DE LOS IMANES SE FORMA UN CAMPO MAGNÉTICO?, ¿SABRÍAS DIBUJAR EL CAMPO MAGNÉTICO DE UN IMÁN?

MAGNET_1 **MAGNET_2** **MAGNET_3**

Figure 2-5 Stages of the learning activities within the AR application (Curtsey to Ibáñez et al. (2014))

Onime and Abiona (2016) built an augmented reality mobile system to handle practical laboratory experiments in science, technology and engineering. The system replicated existing hands-on experiments using photographic markers of laboratory kits. The researchers examined the system based on two learning scenarios in the field of microelectronic and communication engineering. In the engineering scenario, they used an Arduino-compatible board called Seeding Stalker v2, which has embedded sensors for the hands-on experiment. The learning goal was to connect the resistor and light emitting diode (LED) to the board and to pulsate the LED. They used a low-cost 2D photographic mock-up of the same hardware board as a photo-realistic marker for the augmented reality mobile experiment. They built a 3D virtual object for the LED and the resistor and produced step-by-step instructions to connect them to the board using the aid of AR. In the communication scenario, they introduced three types of antennae communication, and learners were asked to establish bi-directional wireless radio links. In contrast, the AR version involved three mock-up marker objects for all antennae, and when the learner pointed at the marker, the radiation appeared in 2D or 3D mode. In both scenarios, 148 students participated in both experiments. After finishing the experiment, they stated that the augmented reality mobile application made a positive impression and was helpful for learning and grades compared to the existing hands-on experiment.

Another study conducted by Jara et al. (2008) took a teleoperation approach using an augmented reality application that gave users the ability to operate and work remotely with real robots with the aid of virtual data supported via a virtual environment. This study highlights the positive aspect of real-time feedback when operating robots remotely. They used online video streaming and updated 3D simulation of current robot positions as feedback options for users. This gave users the competence to verify the work and that it was correct. In the same vein, Ligeour et al. (2005) proposed an augmented reality system that helps human operators achieve complex tasks remotely. Users can control and operate telerobots with the aid of virtual information supplied via augmented reality. However, the abovementioned telerobot studies focused only on remote operation and on investigating the value of using augmented reality as an assistant technology to help them achieve their task. In addition, these studies did not look at how real robots operate or communicate. Instead, they relied on the results of actions. Learners and users in labs have the required equipment but, in turn, need further assistance to help them understand and accomplish their tasks. This assistance could range from very simple tasks, such as assembling components, to more complex tasks, such as robot communications and processes.

Tang et al. (2003) designed an HMD-based augmented reality system for assembling object tasks, and they compared the efficiency of the system to printed manuals and overlaid instructions on LCD displays. They revealed that error rate and cognitive overload were reduced using augmented reality systems. In addition, augmented reality assists assembling and operational tasks by minimising alternating attention tasks, so users do not require shifting between two or more sub-tasks.

Henderson and Feiner (2011) proposed an HMD AR system that supports users in procedural tasks, especially in the psychomotor phase, and helps retrieve the objects of interest. They compared the system with 3D instructions on a monitor and found that HMD AR was better than static 3D instructions were. Similarly, Billinghurst et al. (2008) presented a mobile augmented reality system that enables users to assemble tasks, and the system displays a series of images adapted to the real world view. This approach helps users view complex models on their smartphones, and it enables them to receive a step-by-step guidance for completing the task. This guidance uses animated images linked to the real objects and placed in the right positions. Users found the augmented reality system very useful in assisting them to complete assembling tasks. The limitation of their study was they did not look at assembling smart objects that could communicate and interact with both physical and virtual (computing) objects. Instead, their focus was on physical objects that had no interaction with the computing environment, such as puzzles.

In terms of real-time feedback, Liu et al. (2012) proposed an augmented reality prototype system that provides real-time feedback for operational tasks. They created a generic controller for the manipulated object, so it can communicate with a mobile device. With a mobile phone camera pointed at the marker, it presents an augmented reality view with outlines of the physical controls as well as instructions to complete the task, such as entering values or pressing buttons. The system provides colour overlay as real-time visual feedback that indicates correct or wrong uses. This approach improves significantly the task performance and learning experience compared to text, picture and augmented reality without real-time feedback.

Daily et al. (2003) used augmented reality to visualise information from distributed robot swarms. The robots are used to communicate and work cooperatively to provide information about intruders. Users employ AR to collect and overlay information from each robot and to highlight the shortest path to the intruder. Similarly, Amstutz and Fagg (2002) use augmented reality to present information from large numbers of sensors and robots. Collecting the state of these objects and visualising them via augmented reality helped users in search and rescue tasks. In Addition, Chen et al. (2009) proposed a system that makes it easier for researchers to visualise and interpret complex robot data by augmenting the robot's information (such as sensor map and task relevant data) with the real world environment, maximising the shared perceptual space between the user and the robot. Thus, AR technology can assist humans not only in testing robots' systems but also in understanding complex robot information and improving human-robot interaction.

Collett and MacDonald (2010) argue that mobile robot programming lacks suitable tools to assist developers in the debugging phase. Robots interact with the environment, which makes programming challenging. This interaction causes additional complexity to programmers to first understand what data the robot is receiving and, second, how the robot is interpreting that data. To overcome this problem, they proposed an augmented reality debugging tool that supports programmer views of the robot's world. Instead of using a simulated world by developers, augmented reality allows them to view the robot world with additional virtual objects in the real environment. Thus, developers do not need to shift between the real environment and the robot world because augmented reality brings the robot data in in real world context. One advantage of this approach is to help developers find the source of the errors in the robot application. One example of errors is that robots may miscalculate the orientation and the distance to the nearest obstacle because of incorrect values entered by developers. In this case, augmented reality can immediately show this error in graphical representation. However, the limitation of this study is they used augmented reality as visualisation tools that represent robot data only in 3D or 2D graphical views and placed the abstract data, such as current state and task progress, to the console in plain text output. The system lacks text to support that, if considered, could add more value to discover errors and bugs. Visualisation without meaningful explanation could impede developers in identifying the cause of the problem immediately. Thus, there is a need to structure robot data in an augmented reality view and explain how these data help developers and learners achieve their task goals.

Magnenat et al. (2015) claim that augmented reality and visual feedback enhance high school students' ability to understand event-handling concepts in computer science. They proposed an augmented reality system with integrated visual feedback that overlays the executed events on robots in real-time. The system provides timelines that show location and time of the execution at the physical location. This helps students understand what the robot is doing and helps students trace their program. By using the system, students were able to identify errors in their programs faster and minimise the time between runs. However, students who used augmented reality systems were stressed due to the AR system complexity, such as the complex setup and the system sometimes losing tracking.

2.4 AR Within Smart Technology

2.4.1 Intelligent Environments

The vision of pervasive/ubiquitous computing is to embed intelligence into our everyday lives, such as at work and at home (Suh et al., 2012). This includes the use of embedded computing and network devices that are controlled by intelligent agents in order to enhance the user experience as well as the general quality of life (Ball et al., 2010). This technology has introduced a new relationship between users and the underlying technology, with the objective of making the technology both transparent and prevalent to the users (Chin et al., 2010; Kurz and Benhimane, 2012). In addition, Suh et al. (2012) pointed at the need to understand users' objectives and provide collaborative services by structuring available resources in the environments. They suggested the use of a mediator between users and intelligent systems. Usually, a mediator is integrated seamlessly and invisibly to the users, particularly in the early systems (Weiser, 1999). Carolis and Cozzolongo (2009) found the absence of clear interfaces for controlling device services may result in increasing users' difficulties, especially when interacting with an invisible presence. To overcome this issue, Suh et al., (2012) have proposed the development of Integrated Control Architecture (ICARS), a novel software framework that can be applied as a robotic mediator to collaborate with smart environments. This software allows providers to implement and include numerous collaborative services that require expandability and makes it possible for programmers to develop advanced collaborative applications. Similarly, Chin et al. (2010, 2009, 2008) introduced a vision applicable to a home appliance, known as a soft appliance, which was developed by aggregating basic network services. This vision developed a theory in which home appliances, such as TVs, are decomposed into basic functions. Afterwards, these services can be aggregated in various new ways, with the rest of the deconstructed services used to produce soft appliances that are based on user preferences. A central element in the vision includes a concept that is referred to as a meta-application/appliance (MAp). The MAp includes a template of sematic data that describes the virtual (or soft) appliance that end-users can configure in a manner that reconstructs the composition of an appliance. It is possible to generate such MAps through an explicit end-use programming process that utilizes Pervasive interactive Programming (PiP). PiP seeks real (not graphical) objects and is directed towards distributed computing instead of a single processor.

AR within a pervasive computing space also now provides an opportunity for exposing and explaining some of the underlying functionality within an intelligent environment, for example, making the invisible visible, and using it as an interface to the intelligent environment itself. Mayer et al. (2013, 2012) illustrated a system that enables end users to monitor the flow of information in smart homes with ease and in a non-invasive and intuitive fashion, which allows inhabitants to have a better understanding of the interactions that devices have in smart homes (Figure **2-6**). A mobile service is used in this system to visualize the network traffic between connections and devices in real time. When utilizing the techniques provided by domains of augmented reality and network sniffing, the uncovered data flows tend to overlap in the live camera view of the mobile device.

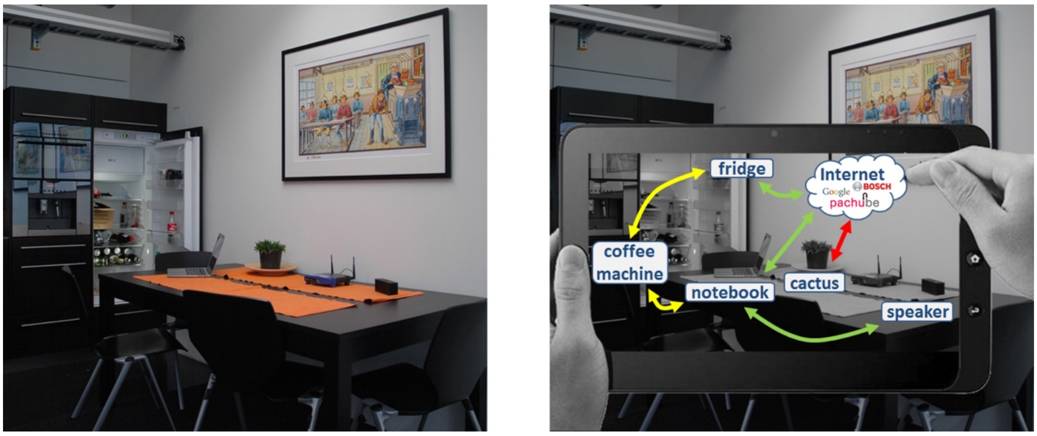


Figure 2-6 Monitoring home devices (Curtsey to Mayer et al. (2013))

This can make it possible for smart homes' inhabitants to closely track the communication behaviours of their smart devices. Communication herein refers to the interactions of smart devices amongst themselves and their connections to the remote services with which they share information. The system monitors the flow of data that occurs between devices while visualizing the connectivity that the devices have with each other. However, the device does not offer any views of either the internal communication or computation processes.

Sato et al. (2013) suggested a novel interface paradigm, known as the cyber-physical user interaction, which can construct a cyber/virtual space, send commands and receive responses from physical/networked appliances via space using AR technology (Figure **2-7**). Control of the networked appliances can occur via a tablet computer or smartphone that is being used as a WiFi controller. Because the paradigm enables appliances from different vendors to be interconnected, it becomes possible for users to operate their home appliances intuitively. Furthermore, the Sato et al. (2013) evaluated the Embodied Visualization with Augmented-Reality for Networked Systems (EVANS) programme to control various home appliances as well as sensor devices at the same time. This was done via a cyber-physical user interaction (CyPhy-UI) paradigm that uses web cameras to retrieve real-life information. A touchscreen display for showcasing the AR visualization as well as the components of user interaction were used to retrieve user inputs.

Figure 2-7 Display image on a tablet computer for EVANS (Curtsey to Sato et al. (2013))

Likewise, Heun et al. (2013) explored a new approach to programming and interaction with physical objects using AR technology (**Figure 2-8**). This approach is based on a smarter objects system that associates a virtual object with every physical object. This provides users with an AR graphical interface that enables them to program or modify the interface and the behaviour of the physical objects. In addition, it enables the sharing of smarter objects functionalities with other objects to make collaborative environments.

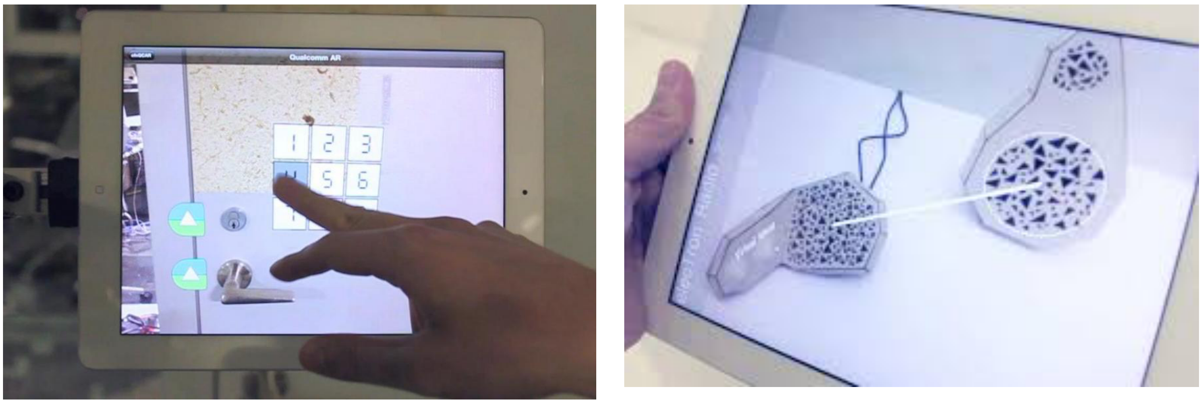


Figure 2-8 AR with smarter objects (Sato et al. (2013))

2.4.2 Internet of Things

Ferscha and Keller (2003) stated that "smart things are commonly understood as wireless ad-hoc networks, mobile, autonomous and special purpose computing appliances, usually interacting with their environment implicitly via a variety of sensors on the input side and actuators on the output side". In addition, the hidden functionalities of any system humans cannot see are considered deep technologies. These hidden technologies are embedded in the environment and are invisible to human sight, although they are still there. They can increase users' perceptions of the environment surrounding them if presented to them in a natural way. Thus, linking the physical world with the virtual one is an important aspect, and this can be achieved by using means such as mixed reality or augmented reality. For example, Ferscha et al. (2003) developed a 6DOF DigiScope, a visual see-through tablet that supports "invisible world" inspection.

The IoT has been established using the concept of smart things and smart objects. Miorandi et al. (2012) consider smart objects as entities that are physical and focused on making communication easier. The objects are designed to have a unique identifier, linked to one name and address. Some of these objects have computing capabilities, given they have sensors to detect physical phenomena. In addition, they are capable of triggering actions that have an effect on the physical reality. Based on these properties, smart objects are known to be context-aware (Plauska and Damaševičius, 2014). For instance, these objects are able to analyse data they received from sensors and can use recognition algorithms in detecting activities and events. In this sense, it is easier for these objects to share data and be able to perform intelligent behaviours based on how each element is designed (Kortuem et al., 2010). It should be noted that the structure of IoT encompasses three layers: hardware, middleware and presentation (Plauska and Damaševičius, 2014). The hardware layer includes actuators, sensors and communication devices. These components have been developed for the existing global Internet infrastructure, designed to link physical and virtual services. The sensors incorporated into these devices provide crucial information about the environment, encourage interaction among users, and provide information to the world. The middleware layer is useful in capturing data, analysis and aggregation. The sensors provide secondary information, which could be useful in synchronising the learning process. The data keeps in mind the physical interactions and the feedback from interacting objects. The presentation or web service layer provides a chance for the things to participate in the business process, and the users have a chance to inquire into the things and their states.

The concept of IoT has similarities with the Object-Oriented Programming (OOP) (see Section 2.5.4). The "things" have a state and a representation of the real-world entities. However, these "things" can only be accessed through interfaces (services). Today, OOP is dominant in software programming. However, students have faced challenges as they try to assimilate the concepts of OOP. It is essential to remember that several approaches have been adopted to help students visualise the learning content, material and tools. As such, students develop a great understanding of the environment, which is significant in visualising the structure of the program (Henriksen and Kölling, 2004). In some cases, abstract visual programming languages (VPLs) could be adopted, which make use of visual elements as opposed to machine instructions (Bentrad et al., 2011). The preference of the VPL is attributed to its simpler domain description (Clarisse and Chang, 1986) and the immediate visual feedback (Burnett, 2001), making it easier for students to grasp concepts.

It is critical to connect learning services and materials to the physical objects where sensors are an important part of the connection. This is important in learning objects and creating an e-learning environment (Plauska and Damaševičius, 2014). Specht (2009) stated that the connection between digital and physical objects is building a new setting for learning. The following are the main three contributions of IoT in education. First, IoT provides a technological background for contextualised learning. It allows embedding smart objects in the environments where learning takes place. The learning content is easily synchronised to reflect the needs of the students.

In addition, learning has been personalised through the help of IoT. Second, it has enabled learners to be immersed in the learning environment, using the physical mobile robotic learning objects. Instant feedback within interactive environments is the fundamental component of effective learning. The adoption of immersion as a learning technique has played a pivotal role in knowledge construction that is determined by pre-existing students' knowledge. Third, IoT has been instrumental in promoting student engagement. It is noted that student engagement helps a great deal in skill development because feedback is immediate, giving students a chance to learn and correct errors (Liu et al., 2013).

2.4.3 Virtual Machine

A virtual machine (VM) is a system that emulates a real machine. It also defined as software implementation of physical devices, capable of running applications, similar to real machines (Mateljan et al., 2014; Nasereddin et al., 2014). The first VM was developed by IBM Corporation in the early 1960s. In principle, a VM works as an abstraction layer on top of the physical machine and isolates the hardware from the operating system (Schoeberl et al., 2011). It also works as a standard application that can be deployed on a real operating system (McEwan, 2002). Usually, a physical system acts as a host and grants a VM the ability to access its underlying functionalities. Thus, users can perform the same functions on a VM as they would on a real system, such as installing the operating system and running programs (Li and Mohammed, 2008) . Some VMs have their own operating system, hard drives and virtual devices.

Chen et al. (2010) discuss the two forms of VM are commonly used today. The first is the system virtual machine, which allows different operating systems to run simultaneously on a single computer. This is typically used in a teaching environment for learning about computer and network concepts (Aagren, 1999; Nabhen and Maziero, 2006). The second form is the application virtual machine which follows the principle of write-application-once-and-run-everywhere. It is designed as a portable runtime environment for certain programming languages (e.g. Java virtual machine (Venners, 1996)).

Although VMs have been used for years by scientists, their virtualisation potential has not received much attention until recently. Virtualisation applications have become popular because of the availability of advanced computer technology (Mateljan et al., 2014). Education is an example of a field that can benefit greatly from such applications as they can simply administration, maximise scalability, improve teaching and minimise the cost of using multiple physical machines (Li and Mohammed, 2008; Mateljan et al., 2014; Nabhen and Maziero, 2006).

Often, incorporating a variety of configurations within classroom or laboratory environments is a valuable approach to enriching student experience, especially in computer science courses. Students usually can choose from many software packages and hardware systems to achieve the learning objectives of curriculums (Stoker et al., 2013). However, it becomes difficult to obtain, maintain and configure these packages as some institutions or organisations may lack resources, especially the financial resources (Qian et al., 2011). In addition, in case of specific learning objectives, a physical classroom or laboratory may not have the required computer science equipment.

A key benefit of the VM is that it reduces the cost of purchasing classroom and laboratory equipment (e.g. hardware and software). Moreover, it gives students the flexibility to perform laboratory activities on any computer (school or personal computers) at any place (campus or home) (Gaspar et al., 2008; Stoker et al., 2013). Further, through a VM, students can access the underlying functionalities of school computer, particularly for courses such as operating systems, security and networking (Gaspar et al., 2008). At the same time, unauthorised user access can also be prevented, especially if the system is made accessible via the internet (Cervera et al., 2016). Cervera et al. developed a virtual learning environment that allows remote students to program and control a simulated or physical robot. The system utilises VM for the execution phase, so instead of executing the code directly into the physical machine, it is executed on the VM. This decreases the risk of controlling sensitive processes within the real machine, and it also restricts student access to certain hardware processes.

VMs provide teachers with the ability to include software packages and hardware systems within their curriculum objectives without the fear of resource constraints (Stoker et al., 2013). They enable teachers and students to experience a wide range of platforms or environments. For instance, Delman et al. (2009) explained problem that students face with simple robotic environments, whether for home or laboratory use, particularly if they are based on the latest operating systems (e.g. Vista and Linux). Most educational robotic systems have their own specifications and programming language; they require students to be familiar with both (Powers et al., 2006). A student may use his/her own personal computer or the laboratory PC to work on educational robotic systems, and these computers may have different operating systems. Thus, students encounter problems with installing or configuring the robotic platforms on multiple computers. To overcome this problem, Delman et al. (2009) developed an educational robotic virtual platform called Code::Blocks, which supports a wide range of operating systems such Windows, Mac and Linux. The developed system makes use of the VM to provide robot programming environments that work with diverse platforms. Thus, students can use Code::Blocks to program, debug and run virtual and physical robotic systems on any operating system at any place (e.g. home or schools).

Similarly, Java technology utilises a VM to create an independent platform that debugs and executes Java programs on any operating system, machine or device (Abenza et al., 2008). A survey of higher education institutions in the UK showed that Java is the most commonly language used for teaching introductory courses on programming (Chalk and Clements, 2006; Chalk and Fraser, 2006). This is possibly because Java is considered an object-oriented language. It is open source, freely available and can run on any machine that has a Java virtual machine (JVM) (Chalk and Clements, 2006). A Java runtime environment (JREs) is a special platform that runs java applications. The main element of a JRE is the JVM, which makes Java a popular and portable programming language that can be deployed across a wide range of devices and machines (Lambert and Power, 2008). JVM hides its source code from researchers or users. One benefit of this approach is that it gives users the ability to use JVM within their preferred platforms and within a variety of programs. On the other hand, it requires users to depend on the standard Java library. JVM works as an interpreter that analyses and executes a sequence of bytes that is produced by a bytecode (Java compiler). This is considered the main component for developing an independent platform for a Java program.

2.4.4 Pedagogical Agents

Feedback, guidance, motivation and encouragement are all often required by learners within their learning environments (Kizilkaya and Askar, 2008). The learners interact with the computer and use it as a means of support to help them learn about certain phenomena that surround them (Soliman and Guetl, 2010). This interaction is facilitated by interface agents, which are sometimes known within educational environments as pedagogical agents (Hewett et al., 1992; Morozov et al., 2003). Pedagogical agents are virtual characters that are used as a means of pedagogical support within a learning environment (Holz et al., 2009; Landowska, 2010). These agents aim to provide assistance to learners based on their knowledge and requirements, as well as responding to the learners' actions (Bartneck and Hostlaan, 2003; Kizilkaya and Askar, 2008). In addition, pedagogical agents have the ability to comprehend the learning context and they can play valuable roles in learning scenarios in order to improve learners' knowledge (Landowska, 2010; Shaw et al., 1999).

The affordance of these pedagogical agents was categorised into five aspects by Veletsianos and Russell (2014). The first aspect is that pedagogical agents are adaptable, which can serve to help with learning, delivering content, and assisting in the development of cognitive processing and metacognitive skills (Clarebout, 2007). The second aspect is that pedagogical agents can simulate human behaviour and communicate with learners (Sklar and Richards, 2010). The third aspect is that pedagogical agents can accommodate learners' preferences and needs, for instance, they can act as peers during collaborative tasks. The fourth aspect is that pedagogical agents can promote engagement, motivation and responsibility among learners (Kim and Wei, 2011; Kim et al., 2006). The final aspect is that pedagogical agents can serve to improve learning and performance outcomes (Krämer and Bente, 2010; Murray and Tenenbaum, 2010).

Soliman and Guetl (2010) developed intelligent pedagogical agents within virtual learning environments that aim to guide and help learners within such environments. They are also intended to explain subjects, provide feedback and respond to learners at any time in virtual environments. Additionally, Qu et al. (2004) designed pedagogical agents that interact and communicate with learners using eye-gaze detection to resolve any confusion or uncertainty that arises during the learning process. The use of these pedagogical agents resulted in learners' motivation being enhanced.

Intelligent pedagogical agents can serve to improve the interaction with learners when providing narration and/or conversation (Soliman and Guetl, 2010). Doswell (2005) developed a pedagogical embodied conversational agent (PECA) to assist and interact with learners during informal learning scenarios such as visits to historical sites. Moreover, Martins et al. (2016) designed animated pedagogical agents to provide hints to learners throughout the learning process. These agents use gesture as a feedback approach to highlight incorrect answers and help learners when they are required to complete a learning task.

Further, the intelligent tutoring system and virtual reality system research has been integrated within the context of education by some researchers in order to enhance learning and teaching (Fardinpour and Dreher, 2012). An AR learning system was developed that incorporated an interactive agent intended to create problem-solving scenarios, assist learners during the performance of an activity and inform them about the state of their learning process (Oh and Byun, 2012). This interactive agent assists learners throughout the learning processes and it responds to the learners' actions. The provision of real-time feedback is vital during the learning process, especially when learners are required to complete a number of actions in order to accomplish their task (Omoda-Onyait et al., 2012; Vasilyeva et al., 2007).

Furthermore, Shirazi and Behzadan (2015) designed an augmented reality pedagogical tool that motivates and engages learners during the learning process. The AR pedagogical tool helps learners to understand abstract concepts within the topics of construction and civil engineering. The tool provides a virtual instructor that aims to assist learners with completing their learning task. Moreover, an autonomous pedagogical agent known as 'Steve' was designed by Rickel and Johnson (1998) to be used as a virtual instructor in a virtual reality environment to assist learners with machine maintenance tasks. Similarly, Barakonyi et al. (2004) designed 'Puppet', an autonomous animated agent that can communicate and respond to learners' actions. They created a hierarchy framework that combined augmented reality, an animated agent and sentient computing within a single user interface. The framework turned physical objects, for example, LEGO, into responsive agents and then used them in AR environments. This allowed learners to be more aware of the impact of the physical objects. Additionally, the animated agent was employed to teach, assist and track learners throughout the learning process.

The use of pedagogical agents in learning environments could result in numerous potential benefits, since they can provide assistance, guidance and feedback to learners while they are performing learning activities. However, the use of pedagogical agents embedded within the technology in order to reveal the computational processes of the physical objects related to the learning process has not previously been investigated. Thus, this thesis tries to overcome this gap in the literature by creating a pedagogical virtual machine that links computational processes with learning processes so as to improve both learning and teaching.

2.5 Linking AR to Learning Paradigm

2.5.1 Learning Objects for AR

The instructional design is currently trending on the ‘learning object’ (Alharbi et al., 2012). Learning objects can be defined as learning materials having pedagogical goals capable of applying and reapplying to different contexts of learning (Sosteric and Hesemeier, 2002). There is a dramatic increase in the repositories of learning objects that store the learning resources such as ARIADNE, SMETE, Learning Matrix, iLumina, HEAL, MERLOT, LearnAlbert, EDNA and Lydia (Neven and Duval, 2002). The availability of several learning object repositories is very common to the students and instructors. Thus, the factor of pedagogy should be primarily considered while designing and providing learning objects (Alharbi et al., 2012).

The learning objects (LOs) concepts were developed in the 90s and are motivated by the necessity of reducing the cost of development and maintenance of digital learning and its reuse and modularization (Han and Krämer, 2009). Learning objects have been differently defined in the literature. The IEEE Standard for Learning Object Metadata (2002) defined learning objects as any referenced, non-digital, reused and digital entity that supports learning at the time of technology. This definition is wide enough to indicate that a learning object can be anything that can be used in education. Nevertheless, there are definitions, as well, that try to compress the scope of the IEEE definition in respect to learning objects. The non-digital items were excluded from the IEEE definition by Wiley (2000), who defined the learning object as only a reusable digital resource supporting learning. Several definitions of learning object have been combined by Sosteric and Hesemeier (2002) to state it as anything with a pedagogical objective that can be applied and reapplied in other learning contexts as well. In addition, Han and Krämer (2009) stated the aim of LOs is to pave a new path for constructing and mediating educational content related to smaller learning units. These self-contained units can be reapplied in numerous contexts and educational environments and can be combined with a coherent collection of learning resources. Han and Krämer indicated that students will be able to understand the concepts of complicated procedures and internal operations with pedagogically designed and interactive learning objects. This can be specifically incorporated into self-paced learning situations where greater degrees of cognitive skills can be ignited with the help of interactive LOs as students will be allowed to conduct processes of arranging concept components or virtual materials or to offer new solutions.

The first attempt of transferring particular principles of software engineering such as decoupling and cohesion to LOs was made by Boyle (2003), who wanted to motivate reusable production. The focus of each component is a single learning objective, and this helps achieve the cohesion among various components in a compound LO of Boyle's approach. LOs are made pedagogically purposeful by combining learning activities and informational objects (IOs) with dynamic objects. Additionally, the unit size of learning objects has been incorporated for defining the granularity of learning objects (Wiley et al., 2000). The aim of learning objects is to decrease the traditional size of the learning material, but the appropriate criteria for small are yet to be decided (Allen and Mugisa, 2010).

The concept of learning objects can be applied to AR learning experiences (ARLOs). The components of learning objects are inherited in ARLOs (Santos et al., 2013). Nevertheless, there are also instructional activities, content and context elements which are specified for ARLEs (Figure **2-9**). The instructional framework and teacher's objectives (such as laboratory, homework, self-study and lecture) would consist of important context elements (Kenkre et al., 2012).

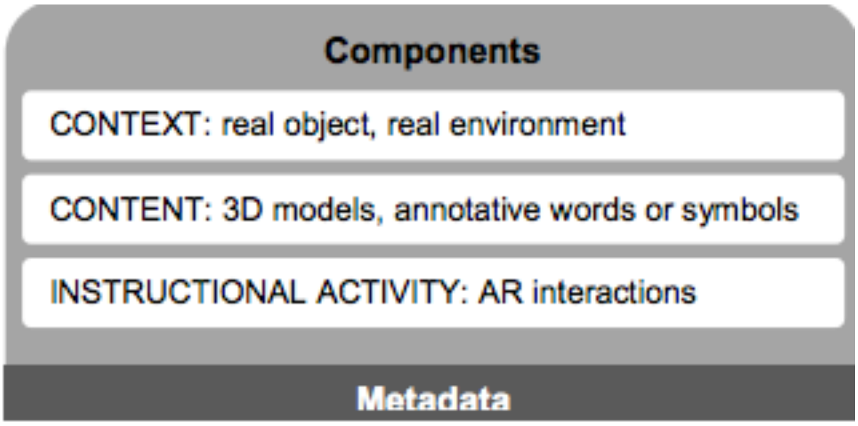


Figure 2-9 ARLOs components (Adapted from Santos et al., 2013)

The context of visualisation is another significant context. Context visualisation is one of the affordances of AR; this means that the virtual information is presented in a rich context of a real environment. The result of context visualisation is advantageous impacts on learning because of two reasons. First, there is an alignment between virtual information and real objects, and this requires reducing the necessity of shifting the attention to various media. Second, the multimodal cues are applied, which are available in a real and familiar environment. Therefore, this helps the students in associating and developing their knowledge. Teachers should adapt ARLOs in relation to environments and objects having the familiarity and accessibility for students to benefit students. Content in learning objects refers to the interface, information (e.g. sound, image and text) and its features. Information adopts the forms of annotative symbols or words such as arrows, circles and numbers and 3D computer graphic models in ARLOs. The teacher must be allowed to measure, interpret and either include or exclude the used annotations and 3D models in the ARLO. The ARLO feature seamlessly integrates the virtual elements into the real environment and adjusts the ARLO according to the students' needs and teacher objectives. Instructional activity in relation to learning objects indicates the discrete steps in the learning process using the learning objects. This step may include or exclude other learning materials. The learning objective of a learning object is explained to the students.

2.5.1.1 Bloom Taxonomy

Learning objects have an ability to support individual instructional objectives, which could be useful in different learning activities (Alonso Amo et al., 2008). Besides, learning objectives describe the specific knowledge needed by the learner, in particular, a learner's willingness to acquire skills and competence. The learning objectives are composed of a set of interrelated shareable content objectives (SCOs), where each item has its importance regarding an item of knowledge. During learning, learners acquire three levels of knowledge. These include syntactic level, pragmatic level and semantic level. These elements were outlined by Bloom as he defined the taxonomy of learning objectives (Bloom et al., 1956). He described the learning objectives as knowledge, application, synthesis, comprehension, analysis and evaluation. In the last level, it is possible for learners to apply the acquired knowledge, which would help them solve their problems. Learners have a chance to evaluate methods, tools and processes, which could be applied quantitatively and qualitatively. Thus, Bloom's Taxonomy has been identified as an instructional model for setting learning objectives for different learning stages. The first version of Bloom's Taxonomy learning objective classifications were introduced in 1956.

- **KNOWLEDGE**: Provides learners with the ability to recognise information, principles and ideas in a form that makes it easier for individuals to learn.
- **COMPREHENSION**: This describes an individual's ability to comprehend and interpret information, according to the previous learning.
- **APPLICATION:** It refers to a learner's ability to transfer, select and use data and principles, in order to complete problems and task with minimum interruptions.
- **ANALYSIS:** This is the learner's capability to distinguish, classify and relate to the assumption, hypotheses, evidence and structure of the statement or question.
- **SYNTHESIS:** It describes a learner's capability to integrate and combine ideas into a product, proposal and plan that could be new to a learner.
- **EVALUATION:** This entails the learner's ability to appraise, assess or come up with a review based on the specific standards and meeting a given criteria.

The taxonomy of cognitive layers of Benjamin Bloom, which is five decades older, has undergone revision so that new understanding related to curriculum and instructional design, evaluation and cognitive psychology can be included (Conklin, 2005; Han and Krämer, 2009). There are six layers of cognitive procedures to elevate the labelled complexity in Bloom's Taxonomy: 'understand', 'analyse', 'create', 'remember', 'apply' and 'evaluate'. Students require identifying useful knowledge or retrieving knowledge from distant memory in 'remember'. Students are able to develop a plan in 'create', such as arranging the building blocks with each other to create a functional and coherent whole, identifying components for a new structure or producing new artefacts. The new terms are defined and explained as follows.

- **Remembering**: This describes the process of retrieving, recognising and memorising of relevant knowledge drawn from long-term memory.
- **Understanding**: It involves a learner constructing meaning from the written, oral and graphic information, making it easier to interpret, classify, summarise and exemplify learned ideas.
- **Applying**: It involves carrying out or using procedures precisely in executing and implementing identified procedures.
- **Analysing**: It entails breaking materials into smaller parts in order to understand how different parts relate as well as knowing the overall structure or the purpose.
- **Evaluating:** It primarily entails making judgements based on criteria and standards, and this is possible through checking and reviewing.
- **Creating**: In this case, the learners put the elements together, forming a coherent and functional whole. Learners reorganise elements in new patterns or structures, and this is possible through generating, producing or planning.

The cognitive skill taxonomy is incorporated in several educational fields, for example, the computer science field. Bloom's Taxonomy is widely accepted, proving a success in teaching learning. (Khairuddin and Hashim, 2008). The incorporation of Bloom's Taxonomy had shown great potential in teaching computer science (Thompson et al., 2008). Several studies concentrated on computer science education, outlining Bloom's Taxonomy effectiveness, in particular, the automatic grading approach in programming. It shows the potential in helping computer science instructors design and evaluate the learning activities (Scott, 2003). Indeed, Bloom's Taxonomy has been efficient in structuring assessments (Lister and Leaney, 2003), helping the experts compare cognitive difficulty levels of computer science courses

(Oliver et al., 2004). The revised Bloom’s Taxonomy has been used in planning diagnostic assessments, particularly in programming, system design and system analysis (Shneider and Gladkikh, 2006).

2.5.2 Constructivism and Constructivism

A good description of a learning theory is an effort directed toward explaining the ways in which people learn, thus facilitating understanding of and insight into the intrinsically complicated learning process. Human learning is undoubtedly complicated, and a number of different scholars have suggested various theories on the different types of learning (Bransford et al., 2006). The basic learning theories centred on people can be broken down into three main categories: behaviourism, cognitivism and constructivism (Saengsook, 2006).

Constructivism is a philosophical approach to supporting knowledge through real-life or alike experiments that promote learning through education (Boytchev, 2015). Constructivism asserts that students enjoy learning in an environment where they have the freedom to explore and create knowledge (Burbaitė et al., 2013), which helps students as they work on projects. While working in such environments, students are given the chance to test their ideas (O’Shea and Koschmann, 1997). Curiosity, which leads to exploration, is a significant factor in developing creative thinking (Boytchev, 2015). Learning is a procedure during which students accumulate knowledge entities and produce knowledge constructs. Overall, constructivism provides views about students’ interests, what they can accomplish and how they develop over time, especially in their methods for doing things and thinking. In contrast to constructivism, constructionism concentrates on how students construct knowledge and acquire new information. In constructionism, students learn by making and engaging with objects. The similarities between constructivism and constructionism are threefold (Ackermann, 2001). First, both are considered constructions in that students can create their own knowledge through personal experience and that this knowledge can be constructed and deconstructed. Second, both theories are developmentalist theories, as obtaining knowledge is an ongoing process; in addition, both indicate the growth of students at different stages. Third, stability and change; closure and openness; and continuity and diversity are all maintained in both theories. Combining both theories can allow for insight into the ways in which people learn and develop at different stages.

The learning process via constructionism can be divided into two phases: construction and deconstruction. The basis of deconstructionism is the constructionist ideas of Papert regarding building tangible artefacts to obtain an understanding of the world (Papert, 1991). The focus of deconstructionism is the contrast procedure of deconstructing real-life artefacts in order to acquire knowledge. Boytchev (2015) employed the term deconstruction to suggest the decomposition of something into separate reusable units. A piece of knowledge regarding a phenomenon or an object is shown in the left image of Figure **2-10**. The learning must be decomposed into smaller and meaningful units of knowledge for the learners in the first stage. After that, those units are used as building blocks to develop personal knowledge (and it does not need to be equivalent to the original). The third phase directs the learner to rearrange the units differently to generate new knowledge.

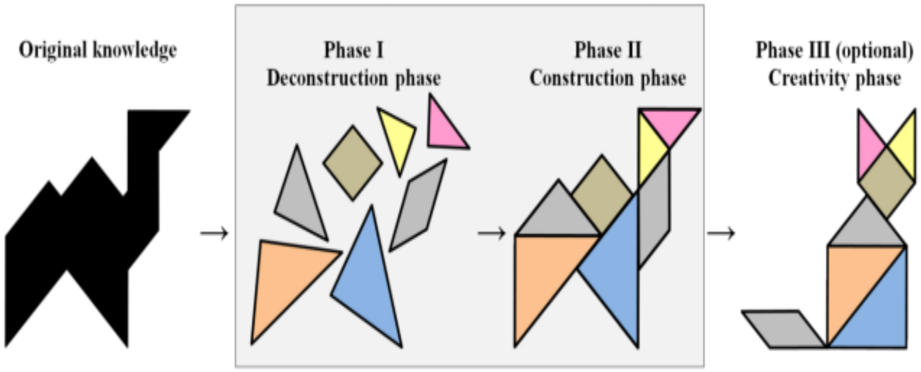


Figure 2-10 Phases of learning through construction, image courtesy Boytchev (2015)

Deconstructionism can also be perceived as a method of problem-solving consisting of two activities: analysis and design. Decomposition of the problems into simpler sub-problems through the typical assistance of formalised rules is involved in the analysis process (Resnick et al., 1988). Its application as a method of problem-solving can be exemplified in software engineering in which cognitive complexity of software systems is handled through modularization and functional decomposition, and a problem is recursively decomposed into sub-problems until those can be solved directly (Wang, 2007).

Extensive use of deconstruction has been observed for learning and teaching in education. From the point of view of teaching, Macdonald (2004) stated that students would obtain a better understanding of the tasks if those are divided in teaching. From the point of view of learning, deconstructionism has been applied to the learning of computer science fundamentals (Self, 1996). Self (1996) explained the source of learning as the differences between the model and situation for the application instead of similarities emphasised in the abstractions of rationalism emphasised in the deconstructionist perspective. Resnick (1990) has given another example where an environment of computer-based robotics (LEGO/Logo) emphasising learning via the design phase of the procedure of problem solving is presented. Students would be able to develop tangible objects with the use of ‘Logo blocks’ (code snippets) and LEGO2 blocks. The basis of Logo blocks is Papert’s Logo Programming Language (Resnick et al., 1988). Considering only children, a text-based computer language has been designed. Later, LEGO/Logo evolved as Scratch, which is a visual programming environment that specifically focuses on teenagers and children, and it is designed for teaching computer programming with the use of animated games and stories (Resnick et al., 2009).

2.5.3 Robotics as a Learning Objects

Robotics is considered an attractive multidisciplinary area that is anticipated to prevail in the 21st century (Shukla and Shukla, 2012; Štuikys, 2015). The current high school and university students will be highly technological, being exposed to service and industrial robots, educational robots, assistive robots and domestic or entertainment robots at work, educational institutions, hospitals and care centres and home, respectively. Currently, students are surrounded by an absolute digital world, and its working techniques should be taught to them. Thus, the educational priorities should be shifted to teaching the students about the operations of those digital devices (such as robots, high-tech gadgets, computers and smart TVs), which are their daily needs (Štuikys, 2015). On the other hand, robots can be perceived as specialised computers having both mechanical and computational facilities so they can carry out tasks involving physical movements. Robots allow demonstrating the capabilities of electronics technology and offer the opportunities for project-based learning to the students (Štuikys, 2015). Taking e-learning into consideration, the use of robots has increased in promoting excitement, fun and involvement in learning, highlighting interests in science, engineering and mathematics careers, and increasing learners’ achievement scores, motivating problem-solving and promoting cooperative learning (Barker and Ansorge, 2007; Beer et al., 1999; Mauch, 2001; Rogers and Portsmore, 2004).

The new advanced learning and teaching technology are becoming more popular (Štuikys, 2015). Learning is being transformed from conventional classroom-centred education to the web-based resources (e-learning; Rosenberg, 2002) and mobile devices (m-learning; Banister, 2010) and to immersive learning within ubiquitous learning environments (u-learning; Jones and Jo, 2004), to context-aware environments with universally personalized content (i-learning; Kim et al., 2011), and to context-aware environments that can overlay virtual educational information related to the real world based on learner location and needs (Tanner et al., 2014).

Educational robots have become more beneficial in the last twenty years as they can now perform the most useful and active learning methods and use the supporting tools to teach science, technology, engineering and mathematics (STEM; Štuikys, 2015). The educational possibilities of robotics in schools have been summarized by Benitti (2012) and found the following: (1) most of the related studies are focused on the areas of robotics (such as mechatronics, robot programming and robot construction); (2) Lego robots have been predominantly used (90%), and (3) learning achievements increased with the use of robotics with respect to the STEM concept, specifically in schools.

An approach has been proposed by Burbaitė et al. (2013) where they considered learning objects in terms of physical entities with two components: software (or robot control programs) and hardware (or mechanical/electrical parts of the robot). It is possible to use both components to transfer knowledge and learning in aggregation and separation. While considering programs in terms of LOs, the robot functions as the environment of e-learning that will translate programs and change them into real-world tasks, such as solving maze tasks. If the hardware is only considered as an LO, the construction of the robotic hardware is focused on demonstrating the mechanical construction principles (such as stability, the centre of gravity and rigidity) along with the laws and concepts of kinematics (such as inverse kinematics, forward kinematics, steering geometries and degree of freedom). However, when the robot is entirely considered as LO, the focus is on the behaviour of the robot to perform specific tasks (wall following, collision avoidance, line following and roaming; Burbaite et al., 2013).

Additionally, Burbaitė et al. have defined five fundamental components of the framework and their interactions. Those components are abstractly identified as technology driven processes, knowledge transfer channels with actors involved, a set of tools, facilities used and a pedagogical outcome. The pedagogical outcome is responsible for implementing learning goals (or objectives) by using a framework of real e-learning in teaching settings. Likewise, it is necessary to evaluate the pedagogical outcome achieved for any product. Three forms of evaluation are anticipated: expert assessment, teacher self-assessment and student self-assessment. However, they did not clearly distinguish between hardware and software domains, rather using the robot as a physical learning object for validating, demonstrating and deploying the 'units of knowledge' (e.g. robot program or an algorithm in the context of CS).

2.5.4 Object Oriented Paradigm

The object-oriented (OO) paradigm is a well-known model used widely in the fields of both artificial intelligence (AI) and software engineering. The core abstraction of object-oriented programming (OOP) is an ‘object’, with associated properties, behaviours and interactions with other objects (Booch, 1994; Cox, 1984; Cox and Novobilski, 1991a; Pokkunuri, 1989). Cox (1984) stated ‘an object oriented program is structured as a community of acting agents, called objects. Each object has a role to play. Each object provides a service, or performs an action, that is used by other members of the community.’ Object-oriented models have been shown to be very powerful tools for dealing with complex human oriented activities. For instance, one view of the world is that people, companies and other organisations are objects, billions of interacting objects, and by properly structuring those objects and their relationships, we end up with the world that functions relatively well, despite the huge complexities involved. OOP adjusts very well, being able to deal with the simplest problems to the most complex tasks. It gives a form of abstraction that vibrates with methods people use to solve problems in their everyday lives (Cox, 1984). Moreover, in the world of technology, OOP has been shown to be a very effective way of dealing with the complexity of programming advanced software applications. A key concept underpinning OOP is the modularity of the object, in which objects act as independent entities that coordinate actions by exchanging messages. Each object is independently implemented and has the required resources to manage its state and behaviour while shielding its implementation details from other objects. This is called ‘encapsulation’, as it hides the user from the need to understand the system at a detailed code level. The user only needs to know what the object does, not how it does it (Cox, 1984).

2.6 Discussion

This thesis proposes to utilise virtualisation principles to deliver a shared pedagogical learning model across a wide range of computer based educational environments by abstracting, combining and unifying ideas drawn from virtual-machines, agents and augmented-reality. This presents a technical and pedagogical challenge of creating, organising and synchronising learning processes with physical objects to be used in laboratory activities. The proposed solution utilises a blend of four dimensions. The first dimension is related to pedagogical views based on the foundation of constructivists theory for creating laboratory learning activities. It uses the concept of deconstruction and construction learning in order to create laboratory learning activities that allow learners to acquire new knowledge. Boytchev (2015) suggested decomposing learning activities into a small number of learning objects that assists learners during learning phases, and after completing phases, learners construct knowledge. The learning objective of the learning activity is based on a pedagogical framework (e.g. Bloom’s Taxonomy) that helps in assessing the learning outcomes for students. Likewise, the second dimension is related to technical views, the vision of Chin et al. (2010), regarding decomposing home appliances into their internal functionalities, which may address the same issues when instructors map their learning activity based on computational objects. Thus, by using the same view, physical objects can be decomposed into software and hardware services that allow integration with learning objects. In addition, IoT can expose physical object (both hardware and software) functionalities to the world, making it available for communication with other worlds, such as an AR learning environment. Thus, computational objects can be smarter and have the possibility to interact with other smart components. Hence, both learning objects and computational objects are derived from an OOP vision as the interest here is related to the communication messages and events that are happening among objects. The third dimension is based on the foundation of visualisation where the communication processes within computational objects are exposed to learners. This uses AR as a means for creating an interactive learning environment that can help students in understanding the invisible entities while performing laboratory learning activities using instant feedback. This can enable learners to construct a meaningful view of the physical objects related to learning and teaching. Especially, the focus is on revealing the abstract concepts learners encounter when dealing with computational objects. Similar to Magnenat et al. (2015), where they demonstrate the use of AR to teach learners about computer science programming concepts, such as event-handling and using robots as learning objects. The last dimension is the use of constructing meaningful information layered architecture that helps learners understand the invisible computational processes. Providing a layered explanation approach for the information gathered from computational objects and generating a pedagogical semantic meaning of this information related to designed learning objects can enhance learners in terms of gaining better insight into the abstract concepts of the technology. Thus, combining the aforementioned dimensions within AR structured learning environments allows learners understanding of the physical world in hands-on activities.

2.7 Summary

This chapter started by introducing fundamental concepts in mixed reality and augmented reality, identifying the affordance and the use of AR within education settings. Additionally, it described the use of AR in STEM laboratory tasks where AR assists learners and users to view information that would not be possible to experience in reality. One of the fundamental affordances of AR is to make the invisible visible, and this shows positive potential on learning. Along with AR, this chapter presented four technological paradigms that could incorporate with AR: mobile learning technology, intelligent environment, Internet of Things (IoT) and virtual machine. It introduced each paradigm and the ability to integrate it with AR to obtain a high-level of awareness of things surrounding us. This chapter illustrated that virtual machines (particularly in educational environments) can maximise learning for diverse platforms, while offering a common interface and language to learners and teachers. Virtualisation can embed pedagogical processes within computational processes to improve learning and teaching. The chapter also introduced pedagogical agents that, when embedded within learning environments, will assist, instruct and provide feedback to learners. Moreover, this chapter presented a review of fundamental pedagogical paradigms that can be used with AR to create learning environments. It introduced the concept of deconstructionism, constructionism and learning objects as well as their use in teaching and learning. From a pedagogical point of view, the use of constructionism allows decomposing learning processes into smaller numbers of learning objects with assigned learning objectives that learners need to achieve. From a technical point of view, the use of constructionism allows decomposing computational objects into their internal services. This can unify both learning and computational objects within an AR structured learning environment to enrich learners' experiences while performing laboratories and hands-on activities. In addition, the use of virtualisation can make learning possible on a wide range of computational objects by providing learners with a common interface. This interface can allow learners to gain knowledge and understand the computational processes in a pedagogical manner.

The following chapters present the theoretical and architectural framework for the implementation of a structured AR learning environment that unifies learning processes with computational objects to construct a robust meaningful view for developers/learners of things in the world.

Chapter Three

3 Pedagogical Virtual Machine Model (Pedagogical and Computational Objects)

> *"If you don't know where you want to go, you won't know which road to take and you won't know if you have arrived"*
>
> *-Truism*

This section introduces a new paradigm, which I refer to as the 'pedagogical virtual machine' (PVM) model, which acts as a manager for revealing educational learning-related functions in the computer. Additionally, it explains the workflow of the model from both pedagogical and computational views as well as structured learning activities within the PVM model. To support the model, augmented reality is used as an assistance technology to help learners/teachers visualise and reveal the processes inside the embedded computing system. Two paradigms are grounded in this model, namely, object-oriented and learning objects.

3.1 Pedagogical Virtual Machine Principles

3.1.1 Virtulisation

Although traditional computer science and engineering laboratories provide students with the equipment needed for their practical work or assignments, students face problems with hidden technologies such as embedded computing. The growth of embedded systems in industry has encouraged higher educational institutions to enhance their embedded-system curriculums to cope with this growth (Qian et al., 2010). This can allow teachers to integrate a variety of embedded computing systems into their curriculums. However, these embedded computing systems often have requirements that could add extra workloads to both teachers and students. Teachers may have to configure the curricula to meet learning objectives or even integrate it in other CS courses. Students may need to cope with the environments, especially if they were exposed to a variety of different embedded systems during their study. Thus, the generalisation of embedded-computing systems in higher education can be challenged due to the following (Ricks et al., 2005):

1. Institutions adapting embedded-computing systems based on the curricula objectives
2. Different technologies used in embedded systems
3. Different environments and architecture between embedded systems
4. Availability of a wide range of embedded applications

Virtualisation can overcome the aforementioned difficulties by providing a portable and adaptable embedded-computing lab platform to students (Hu et al., 2012; Qian et al., 2010). However, virtualisation can do more than provide an inexpensive laboratory platform; it can also allow students to work in a variety of educational environments and receive the same level of learning across these environments without technology or equipment constraints. Virtualisation can also enable teachers to integrate different types of embedded-computing systems within the curriculum and ensure that students gain the same level of learning regardless of the embedded systems used. Thus, creating a platform-agnostic mechanism for teaching across heterogeneous educational environments is the main principle of the pedagogical virtual machine (PVM). The PVM aims to translate students driven computer activity into learning outcomes. Like the Java virtual machine (JVM), it responds to messages but, rather than returniung computational states, returns information relating to pedagogical achievements. Being a virtual entity, it can run on diverse platforms while offering a common interface and language to students and teachers.

It is worth noting that teaching embedded software and systems requires two aspects of thinking: abstraction, and a pedagogical approach. Computer science students learn to make abstractions for computational processes and link abstract layers together (Sztipanovits et al.,2005). Hence, they encounter a problem in understanding the relationship between physical processes and systems. Making clean abstraction hierarchies for physicality can bring problems such as “crosscutting”, which makes conceptualisation difficult. In addition, students’ experiences while working with physical objects or artefacts on laboratory exercises can lack ideation, and the effect of physicality (Srivastava and Yammiyavar, 2016). Therefore, appropriate pedagogical approaches should be used to teach abstraction, and enrich learner experience and knowledge of physical objects. Dede (2008) stated that retaining the effect of the physicality of hardware can be accomplished by using smart objects with embedded intelligence. Therefore, Srivastava and Yammiyavar (2016) defined embedded intelligence as any smart learning object that makes sense of user context and provides instructions as needed. Mattern and Floerkemeier (2010) highlighted that physical objects can exploit the advantages of the Internet of Things (IoT) with regard to embedded information processing, context awareness, and intuitive user interfaces. Thus, the PVM makes such abstract information visible it augments reality, thereby providing a means to supplement students learning by making hitherto invisible computer processes, and pedagogical activities visible to the student (and teacher) advancing both educational technology and augmented reality.

3.1.2 Objectifying

In support of the PVM, Cox (1984) explained that when he started thinking about object-oriented programming, he had the vision that everything in this world can be regarded as an object. This inspired me to think about hardware and software in embedded computers as objects as well. Therefore, the pedagogical virtual machine model implies that all computer objects (hardware or software) contain data that represent the object state and can be communicated with other objects. Thus, the PVM follows the object-oriented principle of representing data gathered from smart physical objects. In addition, the PVM aims to structure both data learning and computational objects within the learning environment. In terms of visualisation, the key aspect of the PVM is to make the invisible visible in structured learning tasks, by exploiting the capability of AR to overlay data from computational objects related to pedagogical processes, from abstract data to more meaningful ones.

3.1.3 PVM definition

Thus, the definition of the *PVM* is*, in simple terms,* that it *is* an entity that interprets and communicates the hidden (deep) computational processes for the purpose of helping students or developers visualise functions in a computer. Its acts as an interpreter for managing educational learning-related functions on the computer, and it promotes a platform-independent interface for students to access information that is pertinent to learning. An important principle of this machine is the unification of pedagogical needs and architectural capability. For instance, a student/learner would need to be aware (via visualisation) of the active software and hardware behaviours. In this respect, it has some similarities with the ideas of virtual machines used to support mobile code in web systems (e.g., the Java Virtual Machine). However, it does not execute code (in a programming language sense) but, rather, responds to a set of generic commands that gather system information (or instrumented data) from the underlying hardware about the software executing. It aims to provide students with a portable, common, and familiar interface irrespective of the underlying hardware (in that sense, acting as a virtual machine – the machine being the monitoring apparatus)

The next section introduces the computational architecture for the pedagogical virtual machine model, which consists of four main layers that range from low-level data collection to high-level data presentation. It then describes the model from two perspectives: an augmented reality view, and an object-oriented view.

3.2 The PVM computational architecture model

The intelligent world can construct a virtual space by integrating ubiquitous devices such as sensors, actuators, digital devices, and legacy systems, which are embedded seamlessly in a physical space (Feiner et al., 1993). In embedded computing, the real world is somewhat messy as it often contains different types of devices that can be interconnected in ad-hoc ways. Neither are their forms structured, as each one can have a different design. The PVM uses an object-oriented approach to structure this data into more meaningful concepts (Cox and Novobilski, 1991).

Figure 3-1 shows the PVM computational architecture model that unifies pedagogical and computational objects within structured learning environments. This model is composed of four layers that provide real-time information regarding the computational objects being learnt. The root of the information in PVM is data derived from the computational objects. Thus, the PVM starts its processes from computational objects, where its capturing software and hardware behaviour are hidden from the user. The PVM has the capability to sense and interpret computational objects' local situation and status. Thus, if an object changes its state, or an event occurs, the PVM captures it and begins to process them using PVM components (layers). The computational objects could range from the things that people use in their daily life, such as cars, washing machines, TVs, aeroplanes, robotics, to mobile technologies. Therefore, by embedding PVM within these computational objects, the PVM can obtain soft and hard behaviour signals to gather essential system information for learning objects. The following sections explain the processes of the main components that contribute to making the PVM system.

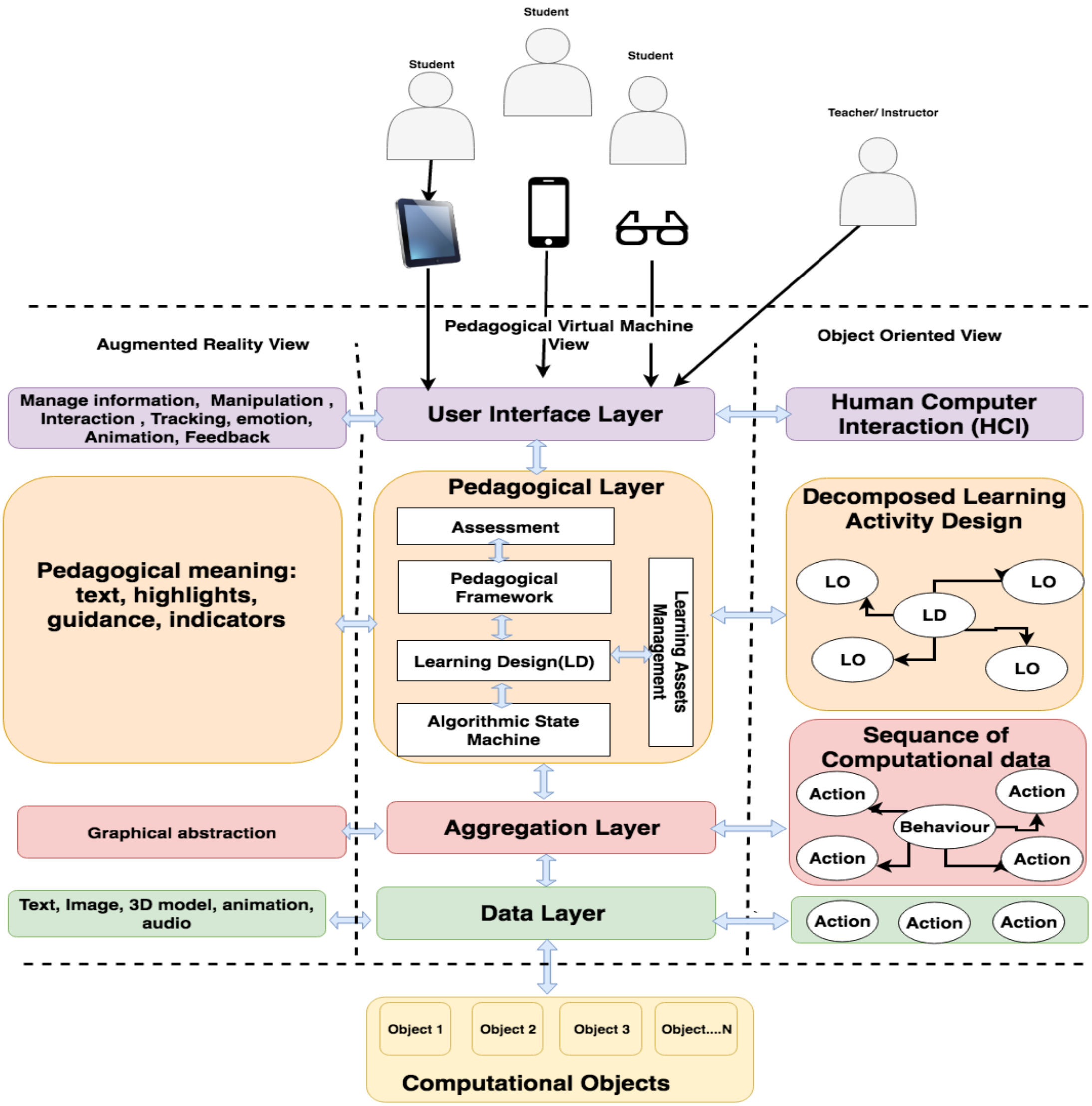
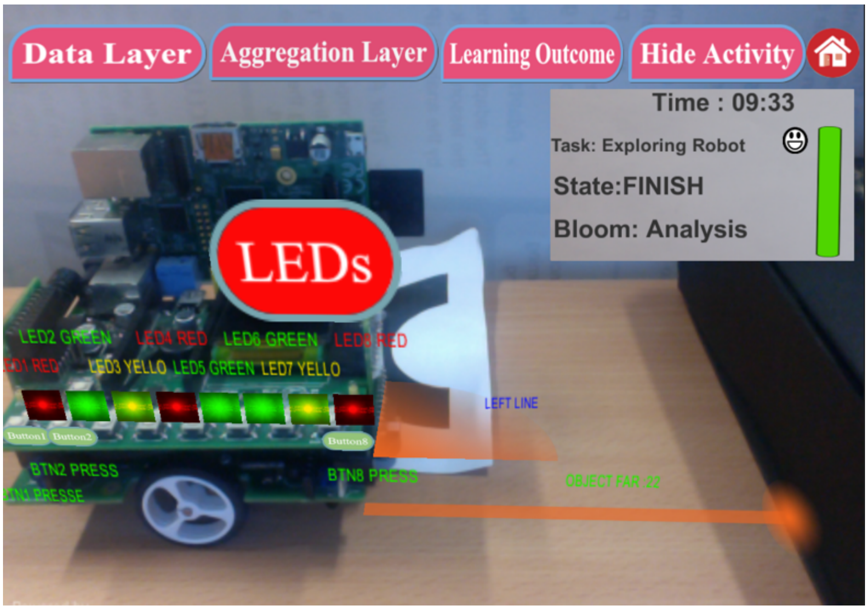


Figure 3-1 Pedagogical Virtual Machine Model

3.2.1 Data layer

The principle of the data layer is that nothing above this layer needs to know any details about the computational objects at all. At this point, this layer presents data in a simplified way. The most obvious way is to encapsulate devices and computer activities (hardware/software) as an object, as the user does not care about what is inside them or what the objects are, but only about the messages. The data obtained (messages) are usually synchronised to specific actions (events). This layer turns these raw data (messages) into meaning (e.g., Motor On/Off, Obstacle far, Light On etc.) by using the instruction or datasheet that describes them. It then encapsulates these data objects and makes them accessible to the layers above. From OO view, these messages (data) may correspond to data member value or output of member function. One of the features of the PVM model is to make the invisible visible; thus, from an AR view, the data are linked to their virtual elements (e.g., 3D virtual object, 2D text, 2D image, animation). In this mapping, the user/learner can have a clear view of the messages that are communicating and interacting within computational objects by using their AR displays, such as tablets, smartphones, HMD, etc. (Figure 3-2). Thus, the main functions of this layer are as follows:

- Translate received messages into meaning (e.g., based on datasheet, tables, instructions).
- Encapsulate messages as an object.
- Make the objects accessible to the layer above.
- Overlay messages to the learners/users via AR.



Time : 09:33
Task: Exploring Robot
State:FINISH
Bloom: Analysis

Figure 3-2 AR data presentation example

3.2.2 Aggregation layer

The principle of the aggregation layer is to produce meaningful information (behaviour) from the data gathered on the layer below (the data layer). Thus, it groups data from the lower level data objects to provide higher value information to the layer above (the pedagogical layer). Aggregating objects can be viewed as a form of composition. This layer enhances the functionality of the lower level objects by creating compound objects (behaviour). The aggregation layer is intended to collect object data from the data layer. The reason for aggregating is to help close the understanding gap, by making the information that comes from the lower level of the model more suitable to that needed for learning in the pedagogical layer above. In this sense, the layer packages the lower level data and makes more meaningful sense out of the data sequences (sequence of actions). In terms of the AR view, every evaluated datum (behaviour) is represented graphically to the learners, so they can understand the received messages in high-level view (Figure 3-2). Thus, the main function of this layer is as follows:

- Retrieve data from the data layer
- Evaluate the data based on a set of rules (e.g., rules set by instructors/teachers)
- Create a compound object that contains the evaluated data (behaviour)
- Overlay behaviour to the users via AR.
- Make the compound objects accessible to the layer above (the pedagogical layer)

3.2.3 Pedagogical layer

The aim of this layer is to provide a structured description of the pedagogical context (i.e., the learning activities), which is used to manage and map the computational (compound technical) objects that come from the lower layer, to support the teaching and learning activities which are then presented through the layer above. By correlating learning and computational objects, this layer can make sense of a learning activity, providing guidance or feedback to the various learning stakeholders (e.g., teachers and learners) via the user interface layer. From the object-oriented perspective, the pedagogical layer utilises the principle of the OO schema to represent a network or society of objects, both learning and computational, although not as an explicit notation but, rather, implicitly. The augmented reality view can then provide a pedagogical meaning to the physical objects used in the student learning activities by overlaying information on the physical views in the form of text, image, highlights, graphics, and 3D objects. This layer consists of five main sublayers: the pedagogical context, the learning design description, an algorithmic state machine, learning assets management, and assessment, which are explained as follows:

3.2.3.1 Algorithmic State Machine (ASM)

This sublayer utilises the ASM methodology to organise the state flow of the compound objects and the state of the learning activity (Levin et al., 2004). Therefore, this sublayer takes every compound object that comes from the aggregation layer and represents it as a state that indicates the current state of the physical object. It then maps the state to the related learning object steps, so the state is actually a compound of two things: the step of the learning activity, and the state of the compound object itself. Finally, it checks each state to determine whether all or one of the learning outcomes of the learning object have been met or not (Figure 3-3).

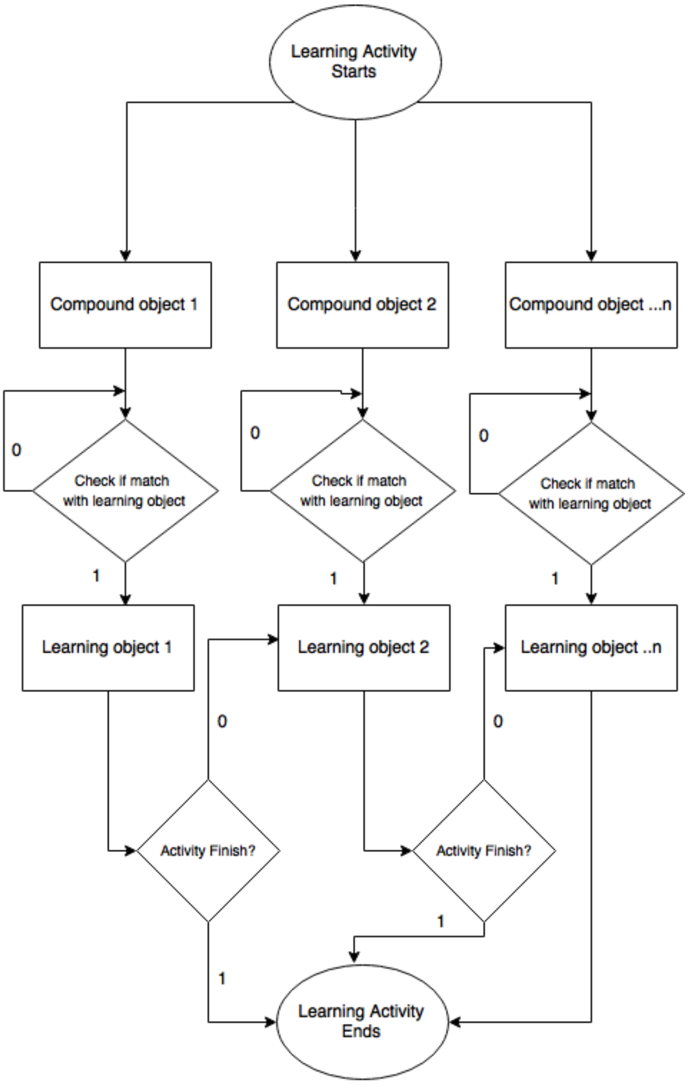


Figure 3-3 ASM example

3.2.3.2 Learning Design

This is based on the concept of 'learning objects' (a well-established scheme for creating and delivering bite-sized lessons, frequently referred to as units of learning) (Koohang & Harman, 2007). The main benefit of designing the learning activity in this way is to maximise its portability and re-usability. Furthermore, it simplifies the structure of the learning activity, so it can be more easily modified. Thus, in this sublayer, it follows a well-known learning design specification called IMS (Instructional Management System) to define learning object structure (Consortium, 2003). This allows the teacher/instructor to define the learning activity, the task steps, the learning objectives, the description of each task, and the expected outcomes. The learning object can have one or more steps in order to accomplish the learning objectives. This layer uses the states provided by the ASM to map the technical state of the equipment to the correct stage in the learning activity (Figure 3-3).

3.2.3.3 Learning assets management

The aim of this sublayer is to store and retrieve all learning material that is designed by the instructors. In addition, the augmented reality virtual elements (e.g., images, 3D objects, shapes, audio, etc.) that correspond to the learning objects are stored to allow easy linking to the computer activities (e.g., data and behaviour).

3.2.3.4 Pedagogical Framework

This sublayer can make use of a variety of useful pedagogical frameworks that are mapped to the learning design layer below. For instance, Bloom's taxonomy of the cognitive domain can be used to describe how the learning objectives can be arranged in a hierarchy from less to more complex (Bloom, 1956). The levels of Bloom's original taxonomy, in ascending order from simple to complex, are Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. Therefore, each learning design (learning object) can correspond to one or more level in Bloom's taxonomy. Using the PVM, it should also be possible to make use of other structured pedagogical frameworks, if this is required.

3.2.3.5 Assessment

There are two types of assessments for evaluating learners'/students' learning activity achievement. The first assessment consists of checking the state of the learning objects which indicate to what extent the learner has met the learning objective (as described in ASM). The second assessment is designed by the instructors/teachers to measure learner knowledge and is based on a common assessment approach, such as open-ended questions, open questions, etc. Both types of assessment can be designed and stored in learning assets management, and the outcome of the assessment is presented to the user via user interface.

3.2.4 User interface layer

This layer provides an augmented reality interface for teachers, students, and examiners to the learning activity. For the students, the interface can be used to guide them through the required sequence of actions needed to achieve the learning goals, as well as to provide them with supporting pedagogical information, through the use of information overlays. For the teachers, it can enable them to set up the learning tasks, as well as provide a record of how well the student has achieved the learning goals. This information can also be accessed by examiners or other moderators. The most visually striking feature of this layer is the image processing aspects connected to views derived from the device's camera. This is based on the use of augmented reality, which combines both virtual and real worlds in a single display (Azuma, 1997). For example, artefacts can be rendered, recognised, and tracked in order to overlay virtual content in the user's display, such as highlighted text, icons, video, graphical images, and 3D models. In addition, it allows learners to manipulate and interact with the tracked objects. The following are the main required components for the AR user interface.

3.2.4.1 Augmented Reality Interface

This is the main user interface where users/learners can see things on the device's camera, and these things/artefacts can be rendered, recognised, and tracked in order to overlay virtual content in the user display, such as text annotation, icons, video, images, and 3D models. In addition, the AR display could range from smartphones, tablets, HMD, projectors, PC cameras, and glasses.

3.2.4.2 Visual Targets

These are the techniques/targets to be used in order to access, and interact between, the real and virtual objects. The interaction could be undertaken by diverse technologies, such as Quick Response code (QR), Bar Code, Near Field Communication (NFC), Video Markers, Computer Vision (object recognition), Global Position System (GPS), interactive sensor/effector systems, and computer networks (e.g., micro sub-nets).

3.2.4.3 PVM components

The AR user interface provides learners and teachers with PVM graphical elements, such as icons and menus. This can make users inspect the information related to the learning objects from different levels of view.

The next section presents the mechanism approach for designing a learning activity to be used within the PVM model. It provides guidance for the construction of a learning activity that involves the use of computational objects.

3.3 Engineering laboratory learning activities design

One of the abstract technologies in computer and engineering science is the embedded computing system. The concept of embedded computing is concerned with computing power embedded within the real environment. Often, it involves the use of small computers or microprocessors which are part of a larger system. From the pedagogical perspective, the internal communication and data inside embedded computing devices is usually hidden, which can make it difficult for learners, as they want to know more about how these devices work and operate together. Therefore, the creation of learning activities based on computational objects and learning design within augmented reality space requires classifying both computational and learning objects to reveal the affordances of the proposed PVM model. Lee et al. (2009) categorised three main factors for ubiquitous virtual reality (U-VR) that can be employed within a mixed-reality environment:

- Reality, which refers to the point where the implementation is located in relation to Milgram's virtuality continuum (Milgram and Kishino, 1994).
- Context, which refers to the flexibility to change and adapt in time and space. Context can be presented as a continuum ranging from static to dynamic.
- Activity, which refers to the number of people who will execute an activity within the implementation, going from a single user to a large community.

Similarly, Pena-Rios et al. (2012, 2013) classified learning activities within a mixed-reality environment into five dimensions:

- A Virtuality Continuum, which refers to activities that involve interaction and manipulation in real time between physical and virtual objects.
- Timing, which refers to the execution time of the activity, whether synchronous or asynchronous.
- Function, which refers to the type of activity, i.e., whether it is a main learning activity (e.g., lab session) or a support activity (e.g., coursework).
- Action, which refers to the work being undertaken in the activity, i.e., whether it is task-based, simulation-based, or role-play.
- Participants, which refers to the activity being designed for an individual or a group of people.

Alrashidi et al. (2013) proposed a four-dimensional learning activity framework for classifying activities from single-user discrete tasks to multi-user sequenced-tasks within an augmented reality learning environment (Figure 3-4).

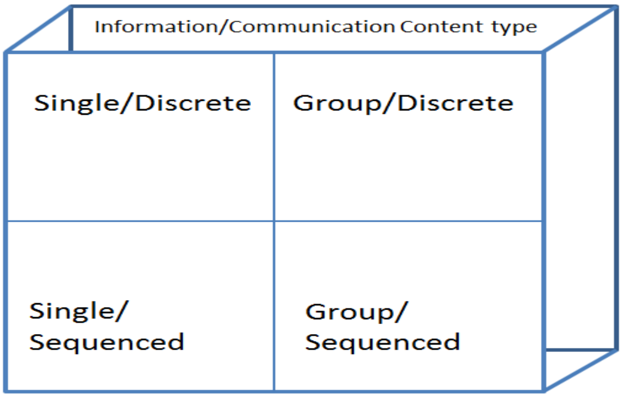


Figure 3-4 4-Dimensional Learning Activity Framework

- **Single-user discrete task:** the learners will be assigned by the teacher to work on one objective of the learning activity that provides information/data about the physical object.
- **Single-user sequenced task:** the learners will be assigned by the teacher to work on several objectives of the learning activity that provide information/data about the physical object. When the learners finish the first objective, he/she goes to the next objective until the unit of learning (UoL) is completed.
- **Group-user discrete task:** A group of learners will be assigned by the teacher to work on one objective of the learning activity that provides information/data about the physical object. Each learner in the group is responsible for performing a task within the learning activity, and each

learner can see related information regarding his/her task. Then, as a group, they can combine/discuss the whole task.

- **Group-user sequenced task:** This is similar to the above description, except that it is based on step by step instructions which go from one learning objective to the next until they complete the UoL.

Based on the aforementioned classification factors of learning activities within mixed-reality environments, the PVM-AR learning environment system is capable of handling the following factors:

- Augmented reality, which refers to the learning environment that enables learners to carry out the activities. It involves the superimposition, interaction, and manipulation of virtual information/elements that are related to the physical object in real time.
- Learning Method, which refers to the type of approach used to carry out the activity, whether synchronous or asynchronous learning.
- Engineering Laboratories Activities, which refers to the type of activities being undertaken by learners, whether assignments, projects, or practices.
- 4-Dimensional Learning Activities, which refers to the number of learners involved in the activities, and the number of learning objectives they are required to achieve.
- Context, which refers to the information that can be observed by learners, whether static context such as images, or dynamic context that analyses a sequence of information/events to deduce sense from them.
- Assessment, which refers to the type of assessment used to examine learners' performance and achievement, based on the activities undertaken. Two types of assessment can be employed. First, automated

assessment, which provides learners with instant feedback (e.g. evaluate and assess students when designing and programming robot behaviour task). Second, teachers/instructors design assessments to examine learners' knowledge based on the learned learning objects (activities).

As stated in previous sections, the design of learning activities in the PVM model was based on the learning objects paradigm, where decomposition is the key aspect of bridging the gap between computational design and learning design. The use of decomposition exposes computational and pedagogy mechanisms within the PVM model. Thus, instructors or teachers should take two aspects into consideration, i.e., pedagogical and computational, when designing engineering laboratories learning activities.

The first aspect is related to the learning activity design, and the instructors/teachers follow a learning objects paradigm to create the activities (Figure 3-5).

- Breaking learning activities into a small number of learning objects.
- Each learning object should have at least one learning objective, based on Bloom's Taxonomy.
- Each learning object should have at least one condition or requirement.
- Each learning object should define implicitly the computational states (behaviour) that will be evaluated with collected computational processes.

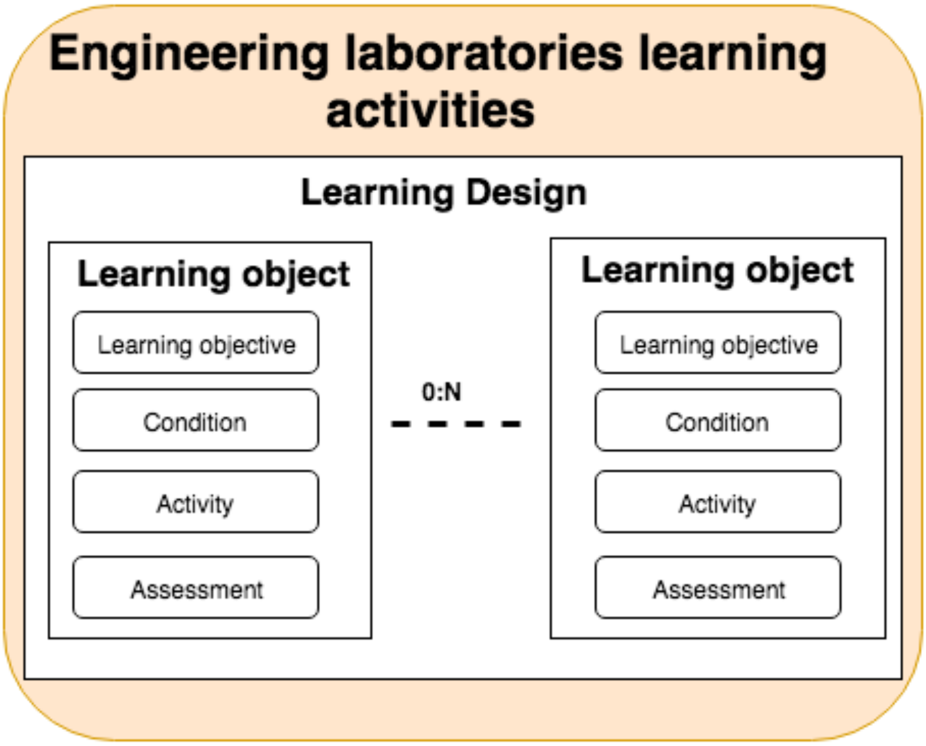


Figure 3-5 Engineering laboratories learning activities design

The second aspect is related to the computational object that is being used by or taught to, learners through the learning activity, and the instructors/teachers should define the following instructions within the system:

- Define all expected atomics operations (e.g., hardware component or software program) that can be received from physical objects (low-level object data).
- Each atomic operation should be associated with a visual representation in AR view.
- Form behaviours or instructions based on the atomics operations (high-level abstraction).
- Every behaviour or instruction can be treated as a state, represented graphically in AR view within the system.

- Use an algorithmic state machine (ASM) as formalism to represent relationships between objects (computational and learning).

By unifying computational and learning objects in laboratory activities, and revealing objects communication and processes using augmented reality, learners enrich their experience by constructing a meaningful view of the invisible concepts in such activities, especially embedded computing.

3.4 Summary

This chapter introduced the pedagogical virtual machine (PVM) computational model that includes components such as data layer, aggregation layer, pedagogical layer, and user interface layer. The proposed model aims to harness computational objects activities with learning processes within structured learning environments, in order to provide support and explanation for learners conducting lab-based computer-engineering learning. Along with the proposed layers, augmented reality was utilised to help reveal hidden computational processes in a way that can improve learning. In addition, this chapter proposed a learning activities design approach for lab-based computer-engineering that is supported by the PVM with an AR learning environment and identified the capability of the model. The PVM model was introduced to contextualise the research presented in this thesis, which focuses on providing a computational architecture that improves learning and teaching, by gathering essential educational-related information from computational objects. Based on this, the chapter provided a description of the PVM layers. As the PVM acts as an interface for obtaining computing activities from computational objects, it turns the collected information into their semantic meaning, using the data layer. These data are then grouped together in order to deduce meaningful information (e.g., behaviour) that can be used within the pedagogical layer. After that, the pedagogical layer takes the evaluated information and relates it to the corresponding learning objects via the use of ASM. Later, the pedagogical layer indicates the outcomes of the learning activity, based on the technical activities gathered. Lastly, all the previous layers are made accessible for the learner to explore while performing the activity, using the user interface. The PVM model was proposed as a solution to bridge the gap between hidden worlds and learning, providing a real-time analysis and pedagogical explanation about the technology being studied. The PVM presented in this chapter introduced the key principles of virtualisation which could be applied to any learning scenarios that involve computational objects.

The following chapter presents the implementation of the PVM model and the architecture that connects learning processes with computational activities. Along with this, it proposes two lab-based computer-engineering educational activities based on the design principles introduced in this chapter. These activities will then be used for the experimental evaluations to explore the pedagogical benefits of using the PVM.

Chapter Four

4 Proof-of-concept PVM system

Chapter three presented the architecture of the pedagogical virtual machine (PVM). To examine the feasibility and the pedagogical effectiveness of the PVM in computing embedded systems, specifically robotics, a proof of concept system was developed to assess learners' understanding of robots' behaviours and actions. This chapter describes the implementation of the system, spanning four phases. The first phase is concerned with the implementation of the physical objects (Fortito's BuzzBot educational robot). The second phase deals with building a distributed architecture for synchronising the physical objects' processes and communication with PVM's augmented reality learning environments. The third phase is concerned with the processing mechanisms of the implemented augmented reality environment and mapping the computational objects' processes to pedagogical processes. The last phase explains the approach to constructing two learning activity scenarios for embedded computing systems that will be used in the experimental evaluation.

4.1 Implementation of physical objects

The pedagogical focus of this research is embedded-computing, an area where the physical elements of the system are as important, if not more, than the software. Interaction with physical computational objects or hardware is part of my evaluation. The physical object needs to be smart to gather information about its processes in real time. Hence, Fortito's BuzzBoard educational components were utilised (Figure 4-1). Callaghan (2012) described Fortito's BuzzBoard as a set of diverse pluggable network-aware hardware boards that can be interconnected, which allows for the creation of quick Internet-of-Things (IoT) prototypes using combinations of plugged modules.

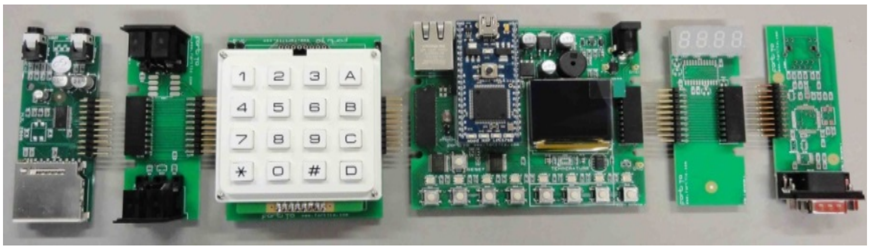


Figure 4-1 Some BuzzBoard Internet-of-Things Components (an Internet Radio)

BuzzBoard was chosen because it uses inter-integrated circuit (I^2C) buses for an intercomponent connect scheme. To detecting Fortito's BuzzBoard components, Raspberry Pi's (RPi) was used. This made it possible to implement discovery and communication between BuzzBoard's components and RPi, which reported on processes and status. However, to do that, it was necessary to develop and implement an API based on the Python-SMBus module, which allowed SMBus access through I^2C /dev interface on Linux hosts (The Linux Kernel Archives, 2009). This ensured detection and monitoring of computational processes and communication between BuzzBoard components. Each BuzzBoard component was represented as a *Class* that contained state and behaviour, using the Python programming language. Python is an open-source general purpose multi-paradigm programming language that promotes simplicity and code readability (Python Software Foundation, 2001). I^2C is a multi-master serial single-ended computer bus created by Philips in 1982 for attaching low-speed peripherals (NXP Semiconductors, 2014). The system management bus (SMBus) is a subset of I^2C defined by Intel in 1995 (SMBus, 2009).

4.2 A distributed architecture implementation

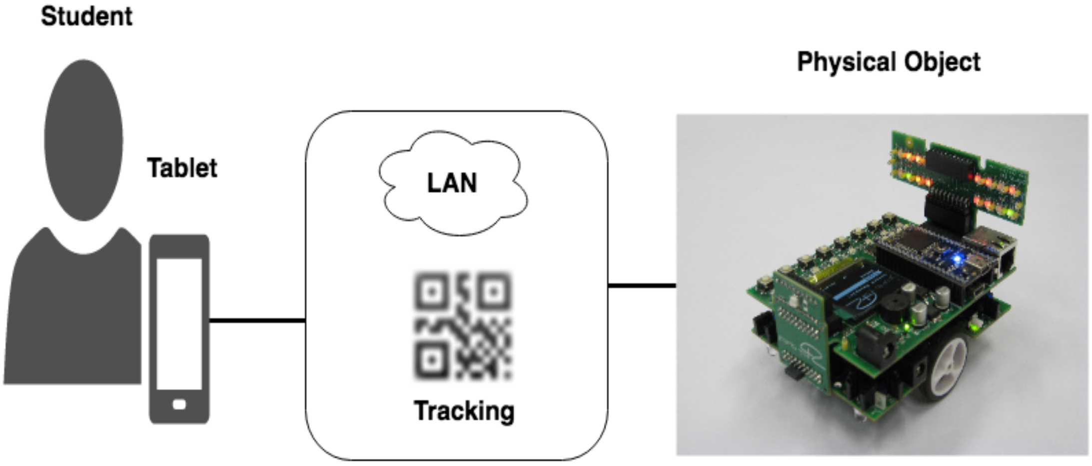


Figure 4-2 Distribution architecture implementation

Figure 4-2 illustrates the architecture for synchronising the physical object's (i.e. BuzzBoards) processes and communication with PVM augmented reality learning environments. Passing computational processes and statuses into PVM augmented reality learning environments requires a combination of two techniques.

The first technique is based on the transmission control protocol/internet protocol (TPC/IP) communication, in line with the paradigm of the Internet-of-Things (IoT), which enables everyday objects to have network connectivity and the ability to send and receive data (Li et al., 2014; Plauska and Damaševičius, 2014; Whitmore et al., 2014). As the system depends on a high level of synchronisation between the physical object and the PVM augmented reality environment, persistent TCP connections were used in the WebSockets implementation. WebSockets are used for managing event-based communications where the channel is kept open on both sides for as long as possible (Bovet and Hennebert, 2013). This was achieved using Tornado, an open-source event-driven networking engine written in Python and used to implement custom network applications. Socket implementation inside the PVM augmented reality learning environment was done using C# libraries in Unity3D.

The second technique is based on the augmented reality tracking, which allows learners to track the physical object using augmented reality displays to overlay relevant virtual information. In an augmented reality field, there are three types of tracking techniques: sensor-based (such as magnetic, acoustic, inertial, optical and mechanical sensors), vision-based (2D image, square and non-square markers, markerless) and hybrid (combination of sensors and vision) (van Krevelen and Poelman, 2010; Zhou et al., 2008). With physical objects (i.e. BuzzBoards), a markerless vision-based technique is used. A key advantage of this technique is that it shows high robustness; it allows for real-time tracking by users using handheld tablet cameras to point at objects and does not require any visual markers (Azzari and Stefano, 2009; Bostanci et al., 2013; Carozza et al., 2014; van Krevelen and Poelman, 2010). Object recognition was achieved with the Vuforia object scanner (Figure 4-3). Vuforia enables enable an AR system to detect and track 3D objects, which increases the interactive experience. In addition, it allows the AR system to augment 3D content and align it to the object. Each BuzzBoard to be used in the system was scanned and overlaid with relevant information.

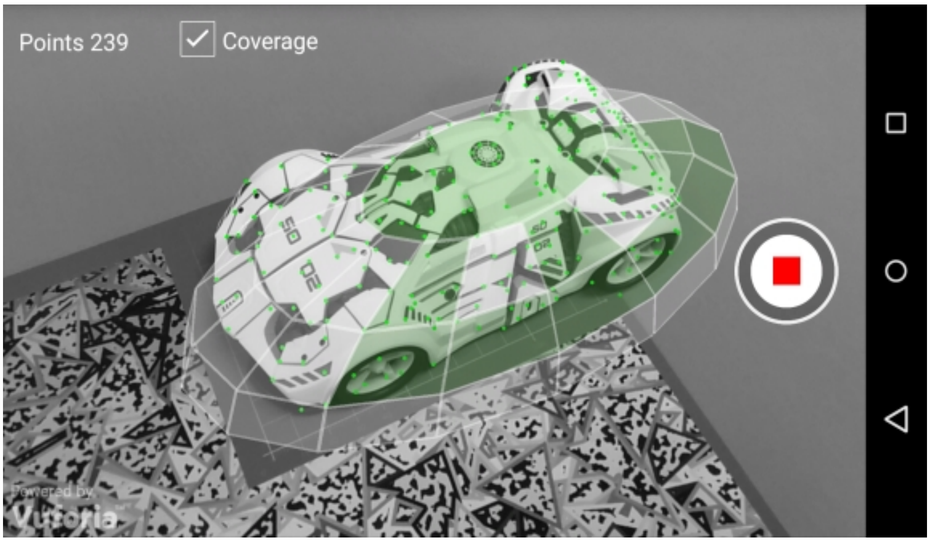


Figure 4-3 Vuforia object scanner example (Courtesy to Vuforia object recognition at www.vuforia.com)

The implementation of both techniques should be checked in the system, and any loss in any technique should be reported on learners' tablets. For example, if the connection is lost, the system should relay the information to the learners and start to reconnect. Whereas, if the tracking is lost, the system should relay information, asking learners to point the tablet's camera at the target object.

4.3 The PVM augmented reality learning environment

Once the connection and tracking were secured, the PVM augmented reality system was ready to listen to and interpret data derived from the physical objects and associate it with learning activity. Bottom-up processes were used, where isolated abstract actions are gathered and analysed to make meaningful pedagogy. The learning activity was based on modularised components (e.g. BuzzBoard), and students were asked to assemble, create, explore, design or build programming behaviour according to the learning task. Once students developed and created the programming behaviour, they executed the behaviour using the programming tool (e.g. Python language). They used the PVM model with AR to inspect the processes and communication of the physical object at the time of execution, as explained in chapter three.

4.3.1 Data layer implementation

The first layer in the PVM model was the data layer that derived object data from the physical data. That is, the data layer only sniffed data without producing a meaningful pedagogical interpretation. The raw digital data (e.g. bytes, hex) provided some meaningful semantics (e.g. the robot module plugged into the system, obstacle detected, the motor rotating clockwise, etc.) before being passed upwards in the pedagogical machine. Every chunk of received data was perceived as object data that contained two or three fragment values, such as the name of the communicating object, the command and value, and the data layer mapped to its semantic meaning. The data layer overlaid every data chunk received from physical objects with visual representation via AR (Figure 4-4). This layer made sensor and actuator information visible. For instance, the BuzzBot robot included a hardware component that could be visible (e.g. LED, BUTTON, MOTOR) or invisible (e.g. LED, IR, LINE SENSOR), and both were visualised and their values updated in real time. Both the sensor data and the action of the robot were overlaid visually on the learners' tablets.

Table 4-1 shows a sample of data objects received from the BuzzBot educational mobile robot's components, along with their produced meaningful action/state. The received data specified the object (sensor or actuator) that was communicating inside the BuzzBot robot (i.e. LED, BUTTON, LEFT MOTOR, RIGHT MOTOR, INFRA-RED(IR)), the command performed, and the value assigned or returned, if any (e.g. LED_1 RED, LEFT_MOTOR FORWARD, IR5 10).

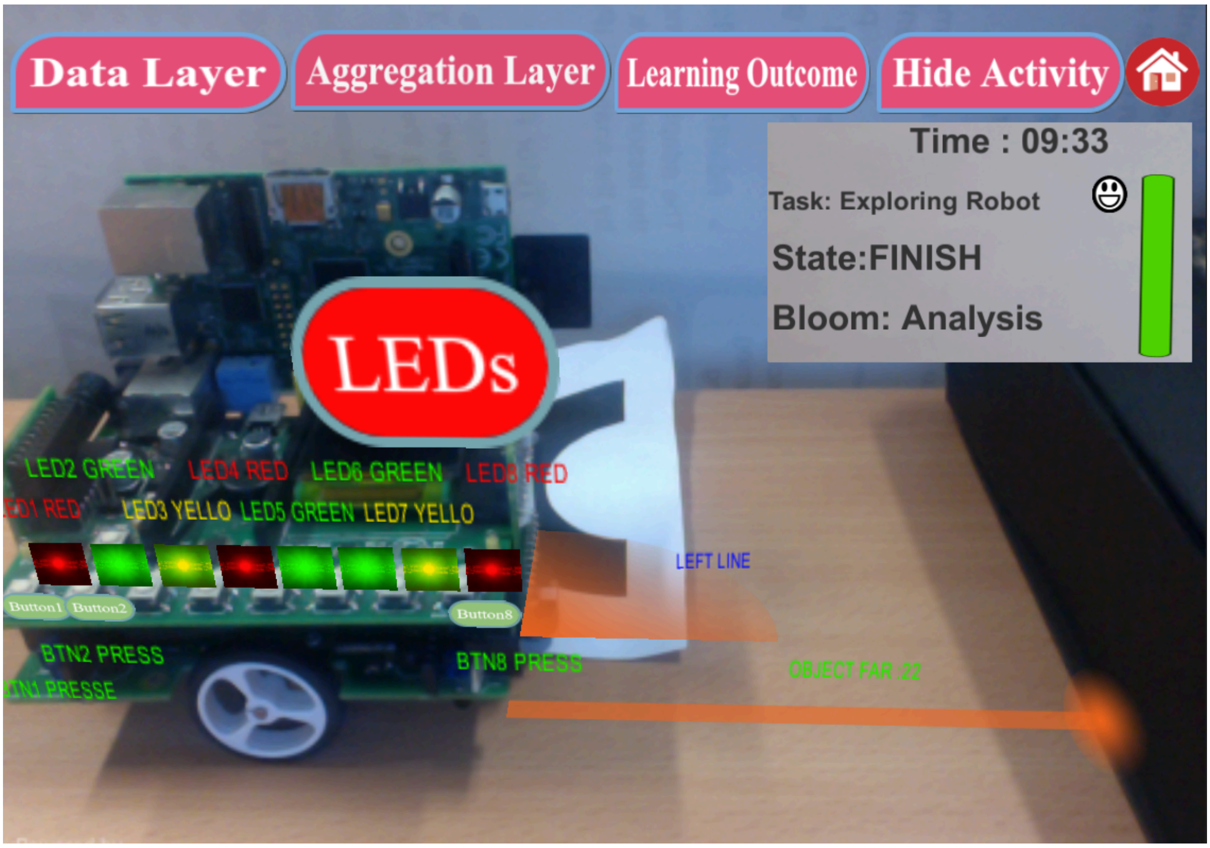
Thus, both hardware and software components could be identified, as well as properties and services.

The data layer overlaid every data chunk received from physical objects with visual representation via AR (Figure 4-4). This layer made sensor and actuator information visible. For instance, the BuzzBot robot included a hardware component that could be visible (e.g. LED, BUTTON, MOTOR) or invisible (e.g. LED, IR, LINE SENSOR), and both were visualised and their values updated in real time. Both the sensor data and the action of the robot were overlaid visually on the learners' tablets.

Table 4-1 Sample of received data with explanations

Received Data			**Meaning produced by data layer**
Object Address (addr)	**Command(cmd)**	**Value**	
0x23	0xFE		LED1 RED
0x23	0xFF	0xFD	LED5 GREEN
0x23	0xFF		LED4 OFF
0x23	0xFF	0xFF	LED8 OFF
0x32	0x00		LEFT MOTOR BACKWARD
0x32	0x02		LEFT MOTOR FORWARD
0x60	0x05	0xF5	STOP MOTOR

0x33	0x00		RIGHT MOTOR BACKWARD
0x33	0x02		RIGHT MOTOR FORWARD
0x28		In range (1-10)	DETECTED OBJECT AT NEAR DISTANCE FROM LEFT SIDE TOP
0x29		In range (11-20)	DETECTED OBJECT AT MODERATE DISTANCE FROM FRONT LEFT
0x2B		In range (21-31)	DETECTED OBJECT AT FAR DISTANCE FROM RIGHT SIDE_TOP
0x22		254	BUTTON1 1PRESSED



Time : 09:33

Task: Exploring Robot

State:FINISH

Bloom: Analysis

Figure 4-4 Example of robot actions (Data Layer)

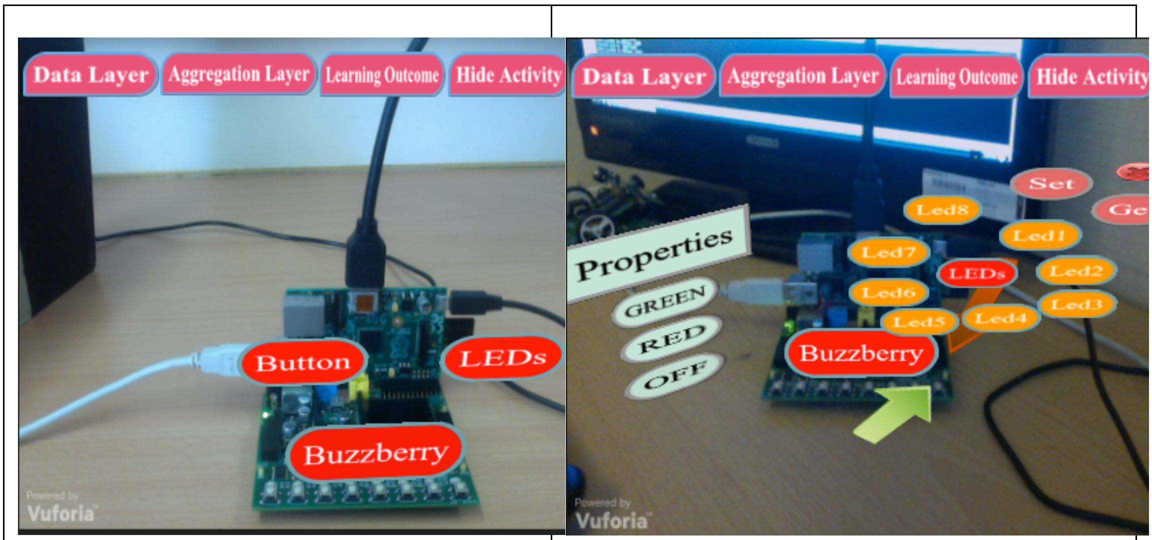
Coloured text, images, and lines and shapes were used to superimpose object data and align them to the physical locations of the object. These allowed learners to explore actions performed on physical objects in real time. All the actions performed were stored on the data object in a data layer class and were available to the above layer (aggregation) for further interpretation. The PVM model used the feature of encapsulation in an object-oriented programming language, wherein the object data is accessible via method/behaviour in the data layer class.

4.3.2 Aggregation layer implementation

The next layer in the PVM model was the aggregation layer, which analysed the sequence combinations of actions or states performed on physical objects by invoking the data layer object. This layer communicated with the data layer object to check for new action received or new components identified. Collected actions or components were evaluated based on a rule-based behaviour approach in order to deduce meaningful behaviours (compound sequence or states, without explicit pedagogical value) (Table 4-2). These behaviours were then stored on the aggregated object in an aggregated class and made available to the pedagogical layer so that they could be correlated with learning objects (learning activities) to produce meaningful pedagogical achievements. Interactivity with the aggregated object was implemented to allow learners to inspect or browse the actions or components belonging to it (Figure 4-5). Every aggregated object was presented as a 2D image with a 'click event' on it for exploring its corresponding actions or states.

Table 4-2 Example of aggregated data

Action	Evaluated meaning
Detect object at moderate distance from left side bottom	**Robot follow wall**
Detect object at moderate distance from left side top	
Left motor forward	
Right motor backward	



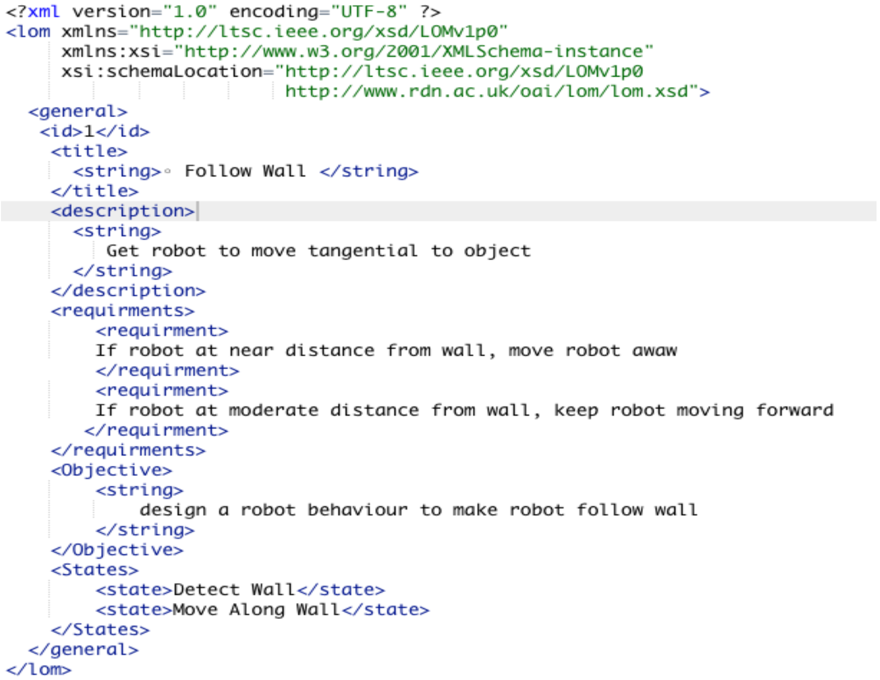
(a) AR shows a hardware component with two actuators belonging to it.

(b) Clicking on one actuator (LED's) reveals properties and services.

Figure 4-5 AR shows aggregated components (a) with interactivity (b) (Aggregation Layer)

4.3.3 Pedagogical layer implementation

The pedagogical layer was the key element linking computational objects with learning objects. The first element in this layer was the algorithmic state machine, which formalised relationships between objects. The pedagogical layer requested the evaluated aggregated objects (e.g. follow the wall, detect the obstacle) from the aggregation layer and each aggregated object was considered as a *state* in this layer. The transition between states for the learning objects was implemented using the state machine approach. The pedagogical layer updated the status of a learning object and evaluated it against the learning task requirement to examine if the learning goal had been met or not. The learning design of the activity was decomposed into a reasonable number of learning objects. Each learning object had its own description, requirements, specifications and objective, and the implementation followed the IEEE learning object metadata. The learning object was stored in the XML file as part of the learning management assets inside the pedagogical layer (Figure 4-6). The assessment of the learning objects was based on Bloom taxonomy, where each learning object is mapped to a level in the taxonomy. The learning management assets included the specifications of the learning objects as well as all the visual elements in the environment, such as images, shapes, 2D text and audio. The achieved learning object and states were stored in the object and were available to learners for self-monitoring their progress on the user interface (UI). This also allowed instructors to view the learners' progress based on what had been achieved for one or all learning objects.



```
<?xml version="1.0" encoding="UTF-8" ?>
<lom xmlns="http://ltsc.ieee.org/xsd/LOMv1p0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOMv1p0
                         http://www.rdn.ac.uk/oai/lom/lom.xsd">
  <general>
   <id>1</id>
    <title>
      <string>◦ Follow Wall </string>
    </title>
    <description>
      <string>
         Get robot to move tangential to object
      </string>
    </description>
    <requirments>
        <requirment>
        If robot at near distance from wall, move robot awaw
        </requirment>
        <requirment>
        If robot at moderate distance from wall, keep robot moving forward
       </requirment>
    </requirments>
    <Objective>
        <string>
            design a robot behaviour to make robot follow wall
        </string>
    </Objective>
    <States>
        <state>Detect Wall</state>
        <state>Move Along Wall</state>
    </States>
  </general>
</lom>
```

Figure 4-6 Follow wall learning object design example

4.3.4 Interface design

4.3.4.1 Human-computer-interaction design principles

Human-computer interaction (HCI) principles play an important role in the design of the pedagogical virtual machine (PVM) system because it incorporates augmented reality (AR) in its interface. As the proposed system provides an alternative to the conventional approach to learning embedded computing laboratory tasks, it should provide students with a useable AR interface. Evaluating the PVM with AR system in terms of its effectiveness at improving learning and teaching is therefore an important aspect of this thesis, although evaluating the usability of the PVM with the AR interface is also useful for indicating the overall usefulness of the system. Gabbard et al. (1999) stated that the potential usability evaluation should be considered when designing such a system interface because it will serve to reveal any potential weaknesses in the interface.

Further, Dünser et al. (2007) demonstrated eight HCI design principles that could be applied to an AR interface. The first such design principle is affordance, which describes the relationship between the subject and the object. By means of device interfaces, augmented reality can overlay information related to or received from physical objects. Thus, it is important to define the relation between these two (i.e. the subject and the object) in the user study. The second design principle concerns the need to minimise the cognitive workload so that users can concentrate on the task at hand rather than striving to understand the AR interface (Dünser et al., 2007; Kaufmann and Schmalstieg, 2006). In addition, according to the third design principle, the AR system interface should enable users to perform their task while expending less physical effort. The fourth principle concerns the ease of use of the system. Rizzo et al. (2005) stated that any user interface which requires users to expend additional effort in order to master it can result in a negative impact when the use of the system is evaluated. User satisfaction is another design principle that should be considered when involving users in the study, since it can indicate how the users perceive and interact with the AR system. Thus, gathering data concerning users' subjective satisfaction and perceptions is an essential factor that researchers should take into account when evaluating an AR system (Dünser et al., 2007; Norman, 2004). Moreover, Dünser et al. (2007) highlighted another HCI principle when developing AR user interfaces, which is related to the ability to provide an interface for a variety of users. This is based on users' needs and preferences. For instance, researchers could provide different kinds of technology for interacting with the AR system, including gesture, speech and leap motion (Irawati et al., 2006). The final two design principles relevant to AR systems are related to technological issues such as slow tracking, poor performance and environmental conditions (e.g. lights). Thus, researchers should incorporate feedback that informs users about the status of the system (Coelho et al., 2004; Henrysson et al., 2005). This thesis considers the eight aforementioned HCI principles when designing the PVM with AR system. However, in terms of the flexibility of use, the system was deployed on tablets (iPads) that the students used to interact with the system. A tablet-holder desk stand was used to support the PVM with AR system to allow learners to work on the physical object (e.g. robot) without needing to hold a pad. Thus, the PVM with AR system design process did not consider the potential of other technologies that could be used for interaction, since such a consideration falls outside the scope of this thesis.

4.3.4.2 PVM with AR interface design

Following the HCI design principles stated in the previous section, a PVM with an AR user-interface design presents learners with a graphical interface for interacting with the physical object. Learners used a handheld tablet as a display to perform the learning activity, interacting with and overlaying virtual information on the physical object. To perform the activity, learners launched the installed PVM with AR application on an iPad2. Once the application launched, the learners could click on the learning activity designed by instructors and begin interacting. Each learning activity can have different overlaying graphical elements, which depend on the learning objectives and requirements. However, all learning activities have the same menu in the AR user interface, corresponding to the PVM layers (data, aggregation, pedagogical). Learners can inspect each layer using the menu, which displays information related to the chosen layer. It is important to display information one layer at a time to avoid overwhelming the learner's view with content (Henderson and Feiner, 2009). For instance, if the learner clicks on the 'data layer' button, only information regarding the data layer is shown. For each learning activity, learners utilised the appropriate graphical elements to facilitate information flow between the physical object and the learner. Examples of overlaid elements on the UI include 2D buttons, 2D and 3D images, text, and shapes. The PVM with AR interface design provides the following:

- Arrows to indicate action: The use of arrows is one of the heuristic principles for visualising processes (Heiser et al., 2004). It is used to highlight the action needed to perform by learner (Figure 4-7).
- Lines to indicate sensors' measurements: Lines are used to represent the physical objects sensor's data (values) in a line shape such as infra-red (IR), line sensor, or light sensor. The line shape is updated instantly based on the information received from the sensors (**Figure 4-9**).
- Graphical elements: The PVM with AR overlays graphical elements that represent physical objects sensors and actuators and show the users the current state of these sensors and actuators (e.g. motor wheels' graphical element, LED graphical element, BUTTONS graphical element)(**Figure 4-9**).

- 2D text: 2D text is used to overlay physical objects' actions and states and also presents instant feedback during the learning process.
- 2D images: 2D images are used to overlay aggregated actions as well as show description regarding the physical object.
- Interactive 2D images. By clicking on each 2D image, learners are able to inspect the aggregated object, and view its low-level actions (Figure 4-5).

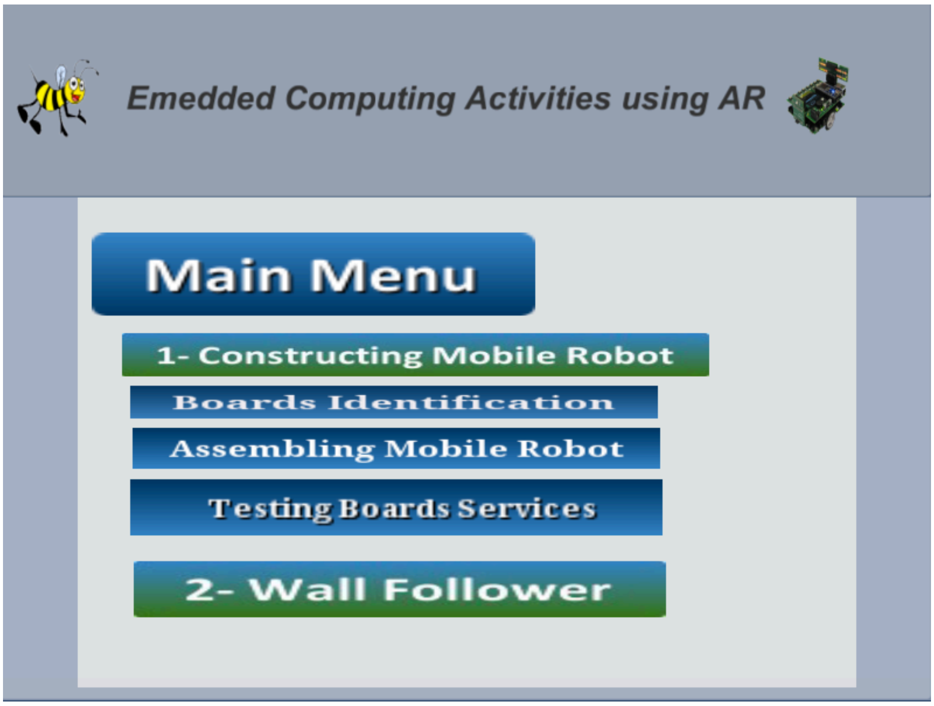


Figure 4-7 PVM with AR Main Menu

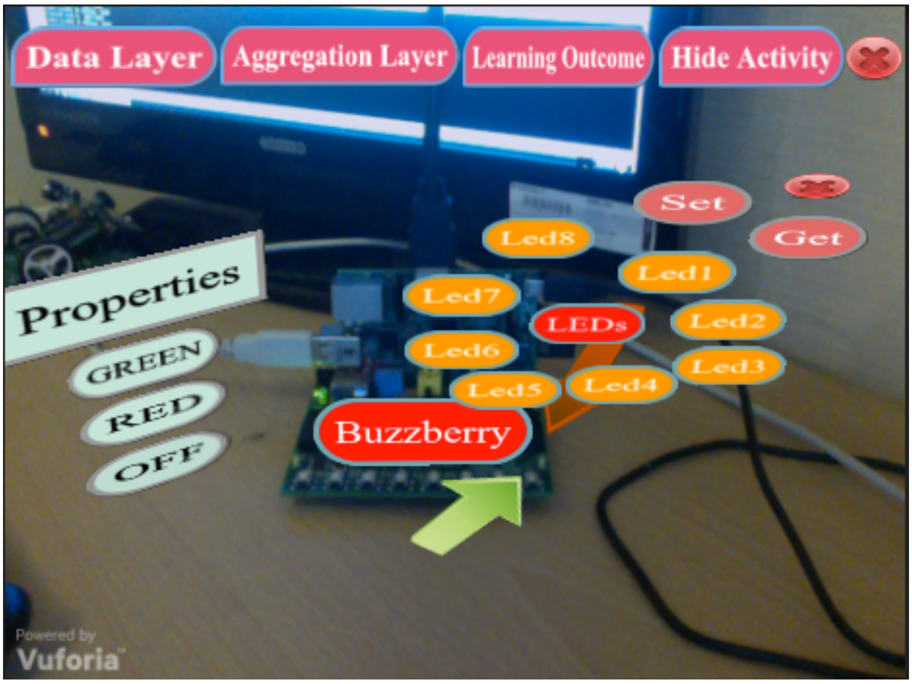


Figure 4-8 Examples of guidance and supplemented information on top of the physical object (Aggregation Layer)

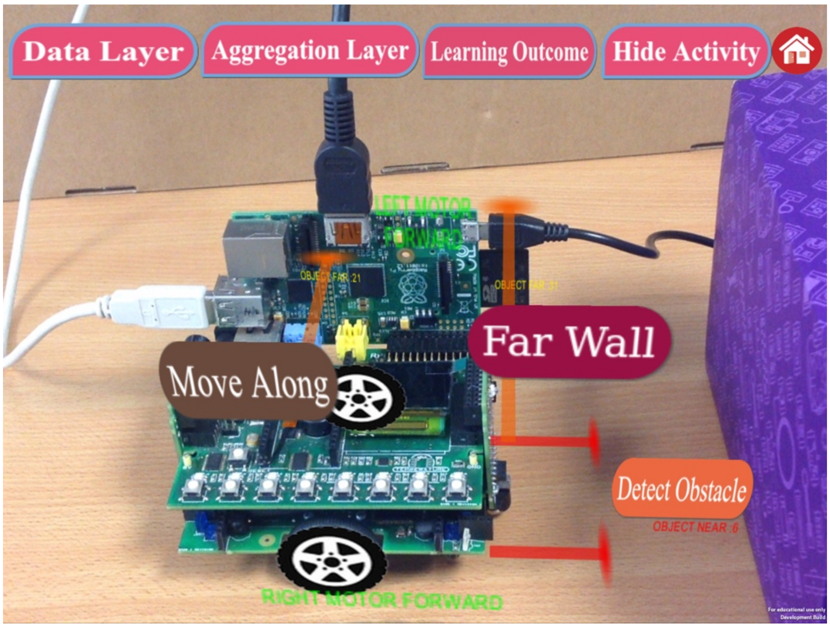


Figure 4-9 Examples of graphical elements used (Aggregation Layer)

The PVM augmented reality learning environment was implemented using Unity3D, a cross-platform game engine for creating interactive 3D content that supports C# and JavaScript routines. Augmented reality implementation was deployed using the Vuforia Augmented Reality Software Development Kit within Unity3D.

4.4 Engineering laboratory learning activities implementation

Two embedded computing learning activities based on BuzzBoard components were designed and employed in the experiment. Both learning activities targeted independent (single) learners rather than collaborative learning activities. Both activities employed the AR 'see-through' approach to superimpose educational components such as processes and communication. The instructional design of the activities followed the procedure of creating laboratory-based engineering activities based on BuzzBoard components, as presented in Chapter three. The developed system did not provide authoring features, as it was out of the scope of this thesis. Thus, two embedded computing learning activities were realised to evaluate the learning benefits of the proposed PVM model.

4.4.1 Student Exercise 1: Assembling and Exploring Embedded Computing

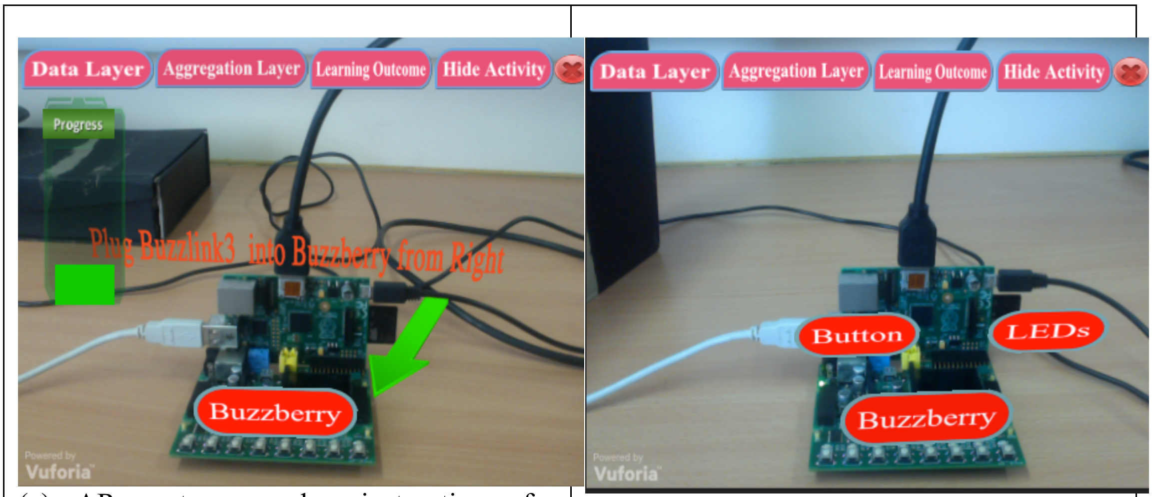
The first learning activity was based on assembling and exploring a modularised educational mobile robot called BuzzBot. The robot was composed of five main modules: BuzzBot, Buzzberry, two BuzzLink3 units, and Raspberry pi. This activity targeted the lower level educational objectives of Bloom's taxonomy: knowledge, comprehension and application. All three levels help students acquire a depth of understanding of robotic components as well as their functionalities and computational processes. In line with the learning objectives paradigm, the learning activity was decomposed into three learning tasks in which the learners had to work on and achieve the desired learning outcomes. The learning objectives of the tasks were introduction, assembly and exploration. These learning objectives required the use of static and live data. The static data were presented as a means of informing the learner about the relevant knowledge and concepts required for each learning step (e.g. such as each attribute of the computing objects), whereas the live data were generated from each physical component (in real time) and provided information about the current state and use of each component. For instance, the modules identified themselves to the system and to each other as they were plugged in. The learning objectives were as follows:

- ***Introduction***: the learning objective of this activity was to identify the BuzzBoard components, which provided students with a full introduction to each of them. Each BuzzBoard component was recognised using augmented reality object recognition presented in Section 4.1.2. Learners pointed their tablet cameras at the module to overlay the relevant information. At this stage, no real live data were produced by the lower layer of the PVM; instead, only pre-defined/static information related to the learning activity were presented as text, images, or 3D visual objects. The AR system provided students with feedback based on their progress.
- ***Assembling***: The learning objective of the next activity was to build/construct/assemble the mobile robot. Through this activity, students could familiarise themselves with the robotic components, their parts, how they should be assembled and their functions and properties. To assemble the robot, students needed to connect five main modules: BuzzBot, Buzzberry, two BuzzLink3 units, and Raspberry pi. AR with PVM provided the students with a sequence of steps, which guided them on building and assembling the

components (Figure 4-10). In the first step, the students plugged the Buzzberry into the Raspberry pi. Then they connected both the Buzz-link 3 units to the Buzzberry. Finally, they plugged the Buzz-link3 units into the BuzzBot. The AR with PVM system informed the learners whether they had connected the modules correctly by listening to the live data produced from the system. The information representation for this learning objective included both static and dynamic data. The static data (information representation) guided learners on achieving each step while the dynamic (live) data gave feedback based on the current state of each physical object. Without feedback from the dynamic data, the learners could not have progressed to the next step until they had plugged the modules correctly. Moreover, dynamic data allowed learners to inspect component data during plugging and explore component functionalities and entities. Therefore, students not only learned how to assemble BuzzBoard components for constructing robots but also how to investigate objects and their functionalities.

- ***Exploring***: After assembling the mobile robot, students were allowed to invoke functionalities using the Python programming language and visualising how certain parts interact and affect the robot. A complete list of robot classes, following the objective-oriented paradigm, was given to the students so that they could interact with and explore the robot. Once students executed a function, the AR with PVM listened to and revealed the data/processes being communicated inside the mobile robot. Robot actions were visualised using the AR view—for instance, 2D text, shapes, images. The AR system provided students feedback on their progress, indicating which object has been manipulated and interacted with (e.g. Button, Left Motor, IR, Line Follower).

- ***Assessment***: The aim of the assessment step was to evaluate learners' understanding of all previous learning objectives, using the hardware and software components.

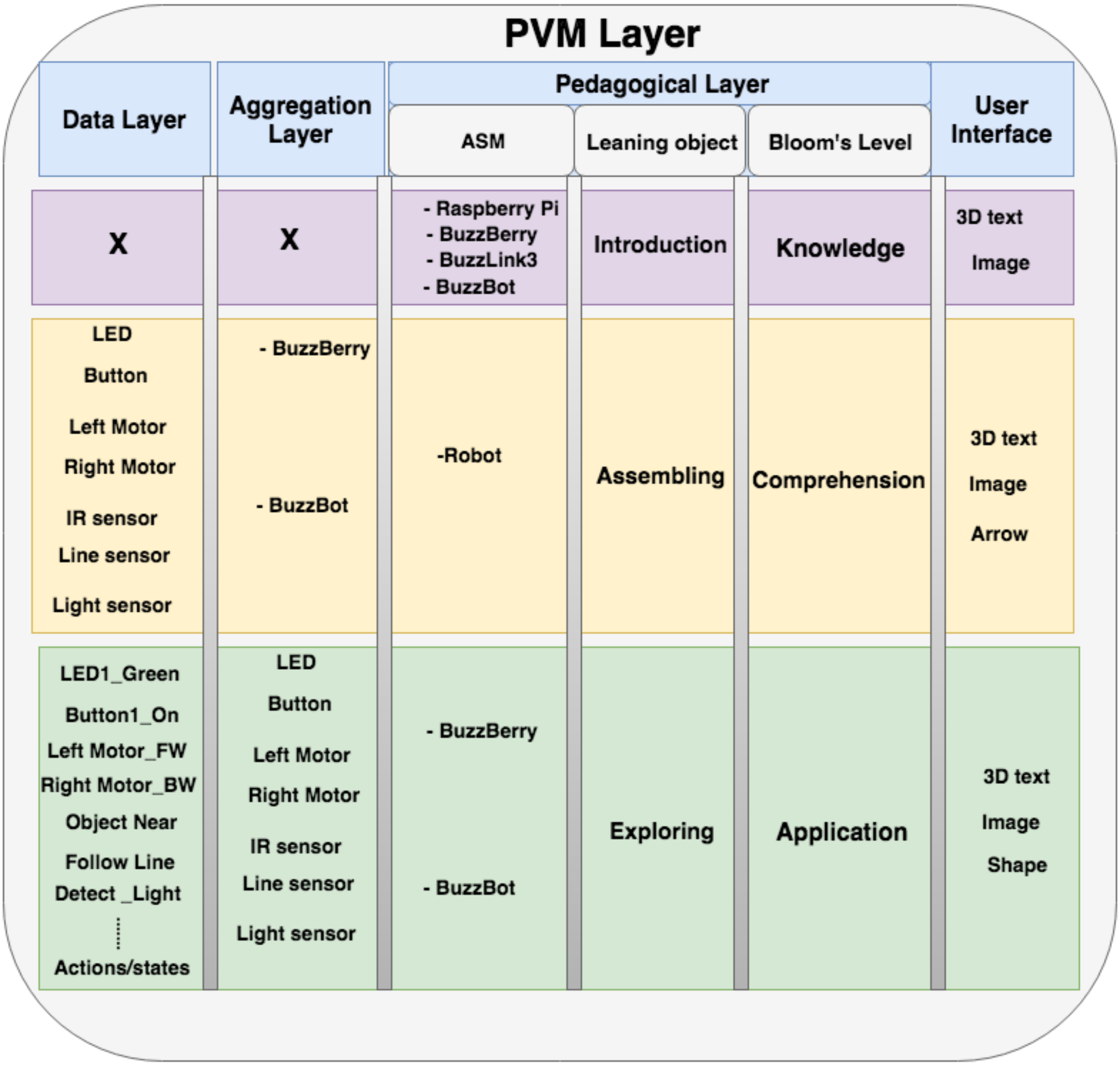


(a) AR system overlay instructions for assembling components with the feedback progress bar.

(b) AR system shows Buzzboard component objects.

Figure 4-10 Example of AR system showing assembling steps

Figure 4-11 shows the workflow of the PVM layers, from left to right. It shows how the learning activities of assembling and exploring require the student to use of both static and dynamic data. In addition, it shows how the computational activities map the learning activities. Moreover, it reveals how both the learning design and computer activities link to Bloom's taxonomy. The X indicates steps/states that did not use any real-time data from the physical object; in other words, the steps were initiated by the learning object (LO) layer itself (rather than a response to a state change from the layer below).



PVM Layer

Data Layer	Aggregation Layer	Pedagogical Layer			User Interface
		ASM	Leaning object	Bloom's Level	
X	X	- Raspberry Pi - BuzzBerry - BuzzLink3 - BuzzBot	Introduction	Knowledge	3D text Image
LED Button Left Motor Right Motor IR sensor Line sensor Light sensor	- BuzzBerry - BuzzBot	-Robot	Assembling	Comprehension	3D text Image Arrow
LED1_Green Button1_On Left Motor_FW Right Motor_BW Object Near Follow Line Detect _Light ⋮ Actions/states	LED Button Left Motor Right Motor IR sensor Line sensor Light sensor	- BuzzBerry - BuzzBot	Exploring	Application	3D text Image Shape

Figure 4-11 Assembling and exploring mobile robot activities workflow

4.4.2 Student Exercise 2: Controlling Mobile Robot Behaviour

Tasks that require students to control a mobile robot to perform a designated activity are commonly used as learning tasks in computer science, electronic engineering, computer vision and robotics. Typically, students are asked to create and design several applications for mobile robots, which include using a line follower, performing obstacle avoidance, goal seeking, maze escape and movement. These learning activities require students to design a behaviour-based robot that follows a wall and avoids obstacles (Figure 4-12). This behaviour is widely known in the field of robotics (Arkin, 1989; Pfeifer and Scheier, 1999).

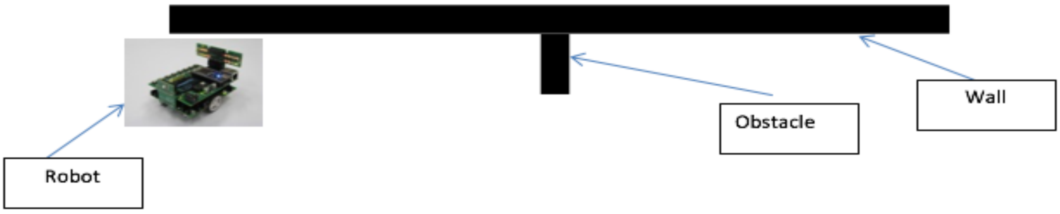


Figure 4-12 Robot test-bed scenario

As part of the learning design, the learning activity was decomposed into reasonable learning tasks. Similarly, the learning activity of designing a behaviour-based robot was decomposed into four learning objectives (

Figure 4-13):

- Get the robot to find the wall
- Get the robot to move along the wall
- Get the robot to avoid collision with the wall

- Get the robot to avoid the obstacle

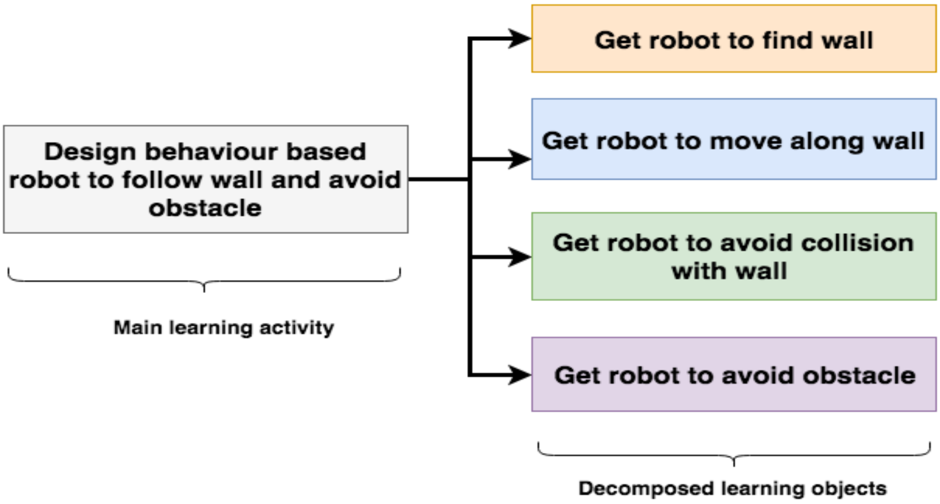


Figure 4-13 learning activity task decomposition

The decomposition of the learning activity into small learning objectives was based on the hierarchical tasks of wall following and avoiding obstacles. The global functionalities, wall following and obstacle avoidance, were decomposed into a number of modules, each responsible for a specific sub-function (**Figure 4-14**). These modules were then treated as individual learning objects. This was because of BuzzBot's modularity feature, which allowed each component to interact with any other module to generate robot behaviour. Modularity is a feature of object-oriented programming, where multiple modules are made first, and then integrated together to create a complete system (Agha, 1990; Booch, 1994; Moon, 1986; Riel, 1996; Szyperski et al., 1999).

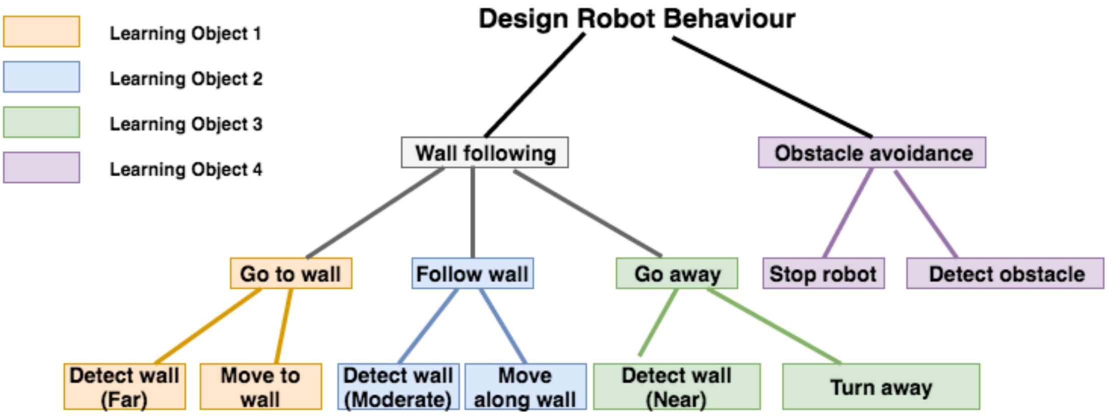
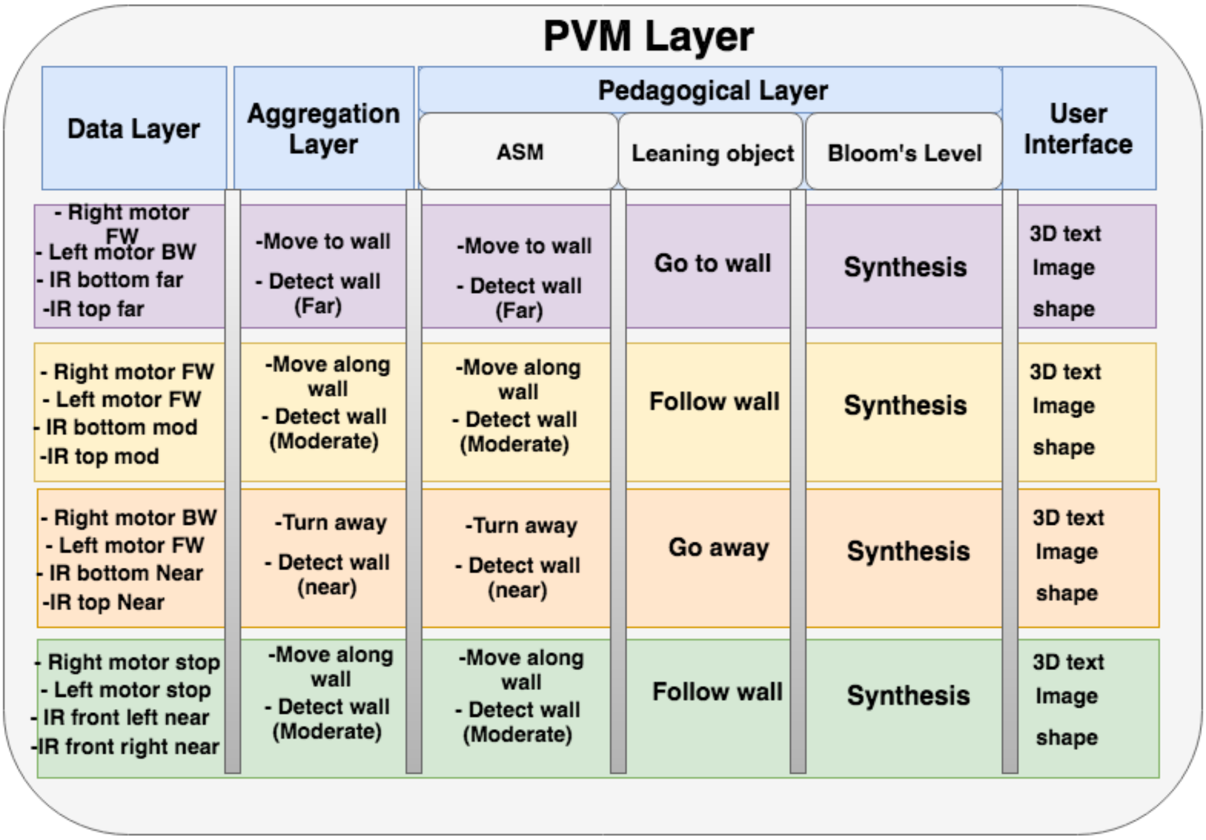


Figure 4-14 Hierarchical decomposition of wall following and obstacle avoidance. (Adapted from Rolf Pfeifer and Christian Scheier. 1999. Understanding Intelligence. MIT Press, Cambridge, MA, USA.)

Moreover, each learning object has its own learning objective and requirement, which all feeds into the main learning activity aim and objective. In regards to Bloom taxonomy, this activity focuses on analysis and synthesis levels, as students are expected to analyse what is needed to construct the behaviour and then use their programming abilities to design a solution that will incorporate creating a robot to follow the wall and avoid the obstacle. As the level of this activity is high in terms of Bloom taxonomy, students should have prior programming skills to solve the learning objects. Figure 4-15 shows the workflow of all learning objects with the PVM model. It shows how low-level data are analysed and correlated with pedagogical processes. Moreover, Figure 4-16 shows an example of the AR view, which supports students during execution time and enriches them with a deep understanding of the robot world.



PVM Layer

Data Layer	Aggregation Layer	Pedagogical Layer			User Interface
		ASM	Leaning object	Bloom's Level	
- Right motor FW - Left motor BW - IR bottom far -IR top far	-Move to wall - Detect wall (Far)	-Move to wall - Detect wall (Far)	Go to wall	Synthesis	3D text Image shape
- Right motor FW - Left motor FW - IR bottom mod -IR top mod	-Move along wall - Detect wall (Moderate)	-Move along wall - Detect wall (Moderate)	Follow wall	Synthesis	3D text Image shape
- Right motor BW - Left motor FW - IR bottom Near -IR top Near	-Turn away - Detect wall (near)	-Turn away - Detect wall (near)	Go away	Synthesis	3D text Image shape
- Right motor stop - Left motor stop - IR front left near -IR front right near	-Move along wall - Detect wall (Moderate)	-Move along wall - Detect wall (Moderate)	Follow wall	Synthesis	3D text Image shape

Figure 4-15 Wall follower workflow within PVM

Actions, behaviours, learning processes and feedback were presented to students via the AR user interface.

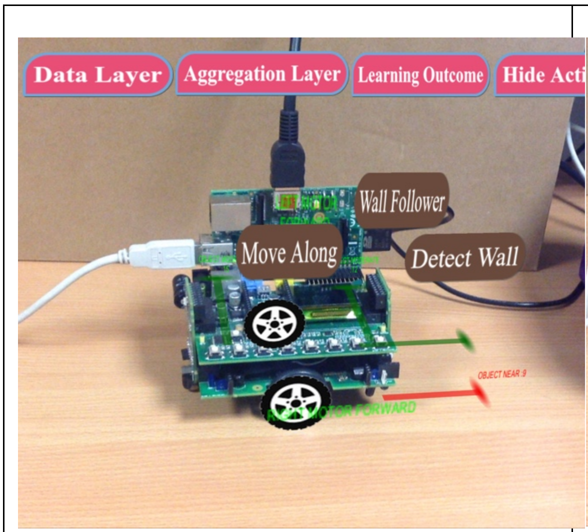
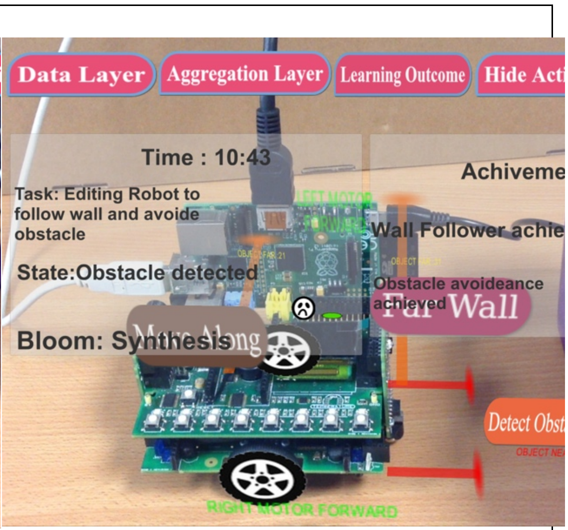
 (a) AR shows learning objects' behaviour with sub-modules in real-time.	 (b) AR shows feedback based on the current state of the robot in real-time.

Figure 4-16 AR shows real-time feedback overlaid on the robot

4.5 SUMMARY

This chapter introduced a proof-of-concept for the implementation of an augmented reality system, which employed the PVM architecture model explained in Chapter three. The developed system is a structured augmented reality learning environment that assists learners and developers in explaining hands-on activities in engineering laboratories. This chapter showed the implementation of the computational objects being learnt by students. This was based on a modularised set of hardware and software components with network capabilities (e.g. BuzzBoards). Additionally, the chapter described the implementation of the distributed architecture that addressed the synchronising problem between computational objects and the AR environment. This involved two related mechanisms: (1) object recognition that allows the AR system to track physical objects and (2) listener connection for computational processes. The PVM framework implementation introduced in this chapter describes how the system manages computational processes with low-level data and no pedagogical value to high-level information that makes pedagogy meaningful. This was aligned with the AR mechanism, which enabled learners to visualise the invisible aspects of learning. Finally, the chapter presented two embedded computing learning activity scenarios, which followed the instructional design discussed in Chapter three. The learning objectives of both the activities were determined by Bloom's taxonomy: the first activity was concerned with lower-level learning goals, whereas the second one focused on higher learning goals.

The next chapter evaluates the learning accomplished under the assembling and exploring activity (first scenario). It compares two educational settings: the AR with PVM system as described in this chapter and traditional hands-on activities in an engineering laboratory. The results of these two test settings are compared and significant differences are discussed.

Chapter Five

5 Experimental Design & Evaluation (Assembly Activities)

"The only source of knowledge is experience."

- *Albert Einstein (1879 – 1955)*

The previous chapter presented the implementation of the PVM architectural models, which resulted in constructing an explanation framework for teaching and learning engineering laboratory hands-on activities within AR learning environments. Two learning activities were designed to be used within PVM with AR learning environment systems to evaluate the learning benefits. This chapter presents the experimental evaluations for the first learning activity, assembling and exploring embedded computing, and reveals the learning effectiveness of using PVM with AR compared to traditional engineering laboratory methods. The chapter starts with an introduction of current AR evaluation techniques and states the evaluation strategy being used in this thesis. Then, the chapter describes the experimental design used to gather evidence of the value of the concepts proposed in this thesis. After that, the chapter presents the experimental results.

5.1 Current AR evaluation techniques

Recently, researchers have focused their attention on evaluating the applications of AR (augmented reality) despite it being studied for more than forty years (Dünser et al., 2008). In fact, Dünser et al. indicated that most of the publications on AR research have been concentrated on applications of experimental prototypes or enabling technologies, such as displays and tracking. In contrast, little evaluations have been conducted on the AR interface user. Dünser et al. stated the reason for that was related to the lack of education especially regarding the mechanisms of evaluating experiences. Additionally, the lack of education may result in having little knowledge in regards to the ways of properly designing experiments, identifying appropriate methods, applying empirical methods, as well as analysing the results. Dünser et al. suggested that the knowledge regarding user evaluations from other disciplines should be collected and brought into AR settings. For instance, the general HCI (human-computer interaction) or psychology has used various methods to study the behaviour of people.

Fundamentally, Bach and Scapin (2004) mentioned three categories of methods that are appropriate for use in mixed-reality systems. They include the inspection methods, questionnaires or interviews, and user-testing methods. On the other hand, the AR user-based experimentation can be classified into three main areas entailing perception, performance, and collaboration (Swan and Gabbard, 2005). Specifically, the perception category involves the experiments studying low-level tasks to understand the operation of human cognition and perception within the context of AR. Consequently, the performance category involves the experiments examining the user task performance for AR application or domains for gaining an understanding of the impact of AR technology tasks. Moreover, the collaboration category involves the experiments examining the generic user communication and interaction among multiple collaborating users. Essentially, Dünser et al. (2008) proposed a new classification scheme for AR user evaluations. As such, it categorises the AR user evaluation research into five major types including objective measurements, subjective measurements, qualitative analysis, usability evaluation techniques, and informal evaluations.

Specifically, the objective measurements are the most popular, involving the task completion times and error rates such as scores, movement, number of actions or position. The studies utilise statistical analysis of the variables that are measured with some having only the descriptive analysis. The subjective measurements include the user evaluation through judgments, subjective evaluation of users, or questionnaires. On the other hand, the qualitative analysis involves the user observations, classification, formal interviews, or user behaviour coding. The usability evaluation techniques require schemes including human-computer interaction (HCI) heuristic evaluation, task analysis, expert-based evaluations, or think-aloud techniques. Furthermore, the informal evaluations entail techniques such as informal user observations or collection of feedback from users. Therefore, the AR evaluations should focus on identifying the functional aspects for attaining the most effective results.

The PVM framework, as described in previous chapters, involves a layered explanation approach that examines computing activities to enrich learners' experience with pedagogical information related to the technology being learned. Therefore, as the proposed framework depends on the completion of the educational activity, user learning and performance studies were utilised to find the learning effectiveness of the PVM model. Learning effectiveness of the PVM framework was measured based on objective and subjective evaluations. Joy and Garcia (2000) indicated that much of the research in the field of instructional technology compare at least two different learning approaches to find out the learning effectiveness (e.g. technology based and conventional delivery media). Thus, the PVM with AR framework was evaluated against traditional learning and teaching methods of the same hands-on-activities. In addition, user perception and usability studies about the use of PVM were involved to reveal user experience of the approach designed to support the task. The evaluation strategy adopted in this thesis will be explained in detail in the experimental design for each experiment. The main reason for this was that there are two experimental evaluations in this thesis, and the constructed factors for each experiment were not totally identical. The next section introduces the experimental design for assembling and exploring an embedded computing learning activity, followed by the results and the discussion section.

5.2 Experimental Design

Chapter three presented the PVM architecture models that embed learning design processes with computational objects in structured augmented reality learning environments followed by the implementation of the key elements that contributed to developing an PVM with AR system as discussed in Chapter Four. The developed PVM with AR system was used in the experimental evaluation for the first scenario, assembling and exploring embedded computing learning activity, to test the hypothesis stated in chapter one, particularly H2, H3, H4, H5, H6 and H8.

Assembling learning objects is already widely used under the category of augmented reality applications for assembly, maintenance, and repair of complex machinery (Billinghurst et al., 2008; Sanna et al., 2015; Tang et al., 2003; Wang et al., 2016). However, it lacks educational values as the current use of assembling tasks is just concerned with teaching learners and users how to build/construct components, and discards teaching learners deep IT technology functionalities. For this reason, introducing and exploring learning objects were added to the proposed scenario. Exploring learning objects was used in the learning activity, as it enabled embedding pedagogical processes within the technology being learnt. For instance, learners were asked to write simple programming code and the PVM provided instant meaningful information to the learners.

Two educational learning applications were developed to be used by learners for solving the proposed embedded computing learning activities. The first application was AR systems that used PVM frameworks, which provides an explanation of the computation objects' processes. The application provided predefined and real-time information related to each learning activity, as explained in Chapter Four. The PVM with AR application was developed with the Unity 3D game engine (https://unity3d.com) using the Vuforia Augmented Reality Software Development Kit (https://developer.vuforia.com). Then, the application was built and run on an IOS apple tablet using the Apple Integrated Development Environment Xcode (https://developer.apple.com). Programming the robot was done with Raspberry Pi (https://raspberrypi.org) using the Geany text editor (www.geany.org).

The second application was based on a more traditional computer science teaching approach. Students were given instructions for performing the learning activity, which offer the same learning activity content and procedure proposed in Chapter Four. Most teachers in computer science, especially in laboratory settings, often distribute handouts or instructions to their students for a given learning activity. For this study, distributed paper-based instruction was designed for all proposed learning objects. The introduction learning objects included paper-based material that explained each Buzzboard component with text and pictures. For assembling learning objects, it contained a picture of each component and instructions that show learners how to assemble the robot. Whereas, exploring the learning object followed the same procedure of PVM with AR application where learners were given a complete list of robot classes and were asked to manipulate and interact with the robot's functionalities. The difference between both approaches was on the visualization and the explanation techniques as PVM with AR allows learners to see the action of the invoked method overlaid on learners' tablets and utilize the PVM framework to deduce pedagogical meaningful semantics, whereas the traditional approach allows learners to see the result of the method/function of the robot without technological and pedagogical help. It just showed the performed action on the programming environment console.

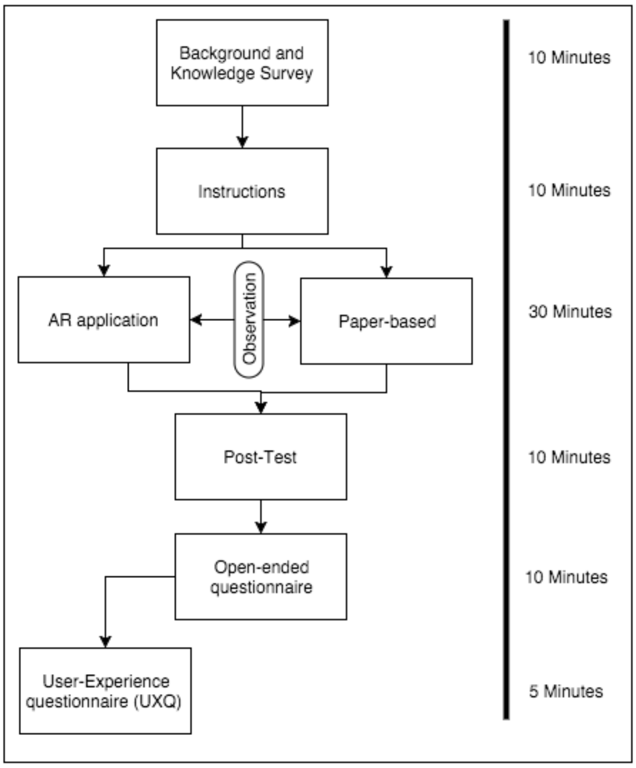


Figure 5-1 Experimental design procedure

A between-subject design approach was chosen and the number of participants was 18 students for each group, which equals 36 participants in total. All participants were undergraduate and postgraduate students at the University of Essex, especially targeting the school of computer science and electronic engineering and were assigned randomly to either PVM with AR or the paper-based approach. The study took place in the iClassroom at Essex University, and participants from the School of Computer Science and Electronic Engineering (CSEE) were invited to take part in the experiment (**Figure 5-1**). The experiment was divided into three phases:

1. Before the experiment. Participants were informed about the study and their phases by the instructor and were provided with an online survey link to gather information about demographic information, computer science experience, the object-oriented paradigm, embedded systems,

and embedded system technology, the Internet-of-Things, and immersive technology. This was used as an indication of the validity of the samples. The survey link was completed before the day of the experiment. A copy of the questionnaire is provided in the Appendix A.

2. At the day of the experiment. Participants signed a consent form, which stated their participation is voluntary and can stop at any time. Then, participants were assigned randomly to the experimental group (PVM with AR) or control group (paper based), groups were given a brief instruction corresponding to each application and their use. As a result of the availability of only one physical object (buzzboard components), participants were invited individually based on their preferred time/day. Each participant completed the three learning objects based on the respective application. During this time, the instructor/researcher observed participants by taking note for further analysis. A tablet-holder desk stand was used to support the PVM with AR group to allow them to work hands-free.
3. After the experiment. Participants/students completed a post-test followed by a user questionnaire to examine the learning effectiveness of each approach. Students who participated in PVM with AR answered an extra section in the questionnaire to obtain feedback about their experience of using the application. The estimated time to finish all learning objects was around 30 minutes, and for both questionnaires (pre-and-post) it was around 20 minutes. The time given to answer the post-test questions was 10 minutes. The time for

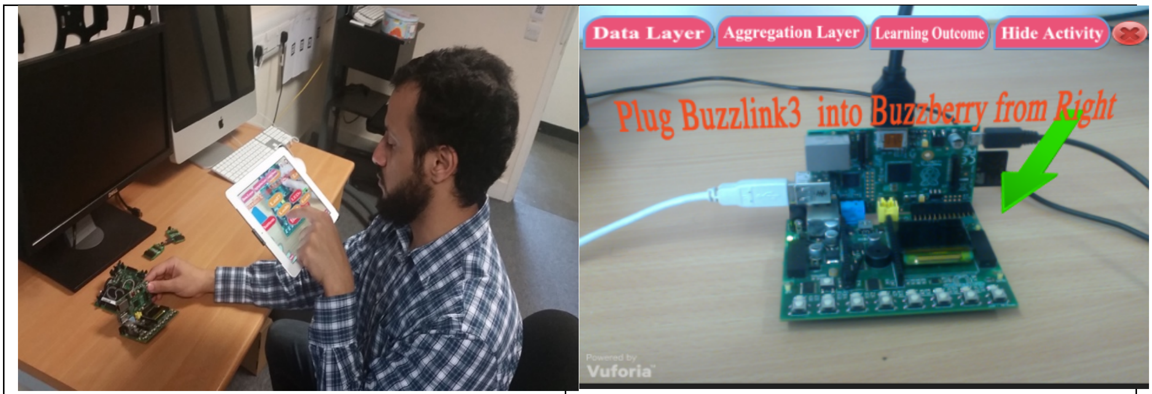
completing the user experience questionnaire was five minutes. A copy of all experimental materials is included in the Appendix A.

Quantitative data were collected when evaluating both learning applications (PVM with AR and paper-based). The collected quantitative data was investigated through descriptive statistics to find correlations with the hypothesis. The research instruments consisted of three questionnaires for participants, a knowledge test, observation (including manipulated objects and action performed), and time recording. The first questionnaire collected general demographic information and preliminary knowledge on computing, embedded systems, and technology to establish participants' background. The second questionnaire measured five factors based on participants' subjective experience. The first factor was a task workload, adopted from NASA-TLX, as a common standard measurement for cognitive overload (Hart and Staveland, 1988). This includes six sub-scales: mental demand ("*How much mental and perceptual activity was required?),* physical demand *("How much physical activity was required?"*), temporal demand ("*How much time pressure did you feel because of the pace at which the tasks or task elements occurred?"),* performance ("*How successful were you in performing the task?"*), effort ("*How hard did you have to work (mentally and physically) to accomplish your level of performance?")* and frustration ("*How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task?").* This factor was used to assess the workload experienced while performing the learning activity. The second factor measured the learning effectiveness of the approach used. The third, fourth, and fifth factors were adopted from the intrinsic motivation inventory (IMI) questionnaire, which aims to assess usefulness, enjoyment, and competence in activities conducted in a laboratory experiment (Ryan, 2006). The third questionnaire measured participants’ experience of the PVM system. The design of the questionnaire was adopted from a user experience questionnaire (UEQ) to assess the strengths and weaknesses of interactive systems (Laugwitz et al., 2008). It includes 26 paired items (semantic differential scale) that form six factors: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. Reliability of the UEQ scales is typically high, i.e., the Cronbach-Alpha coefficient is typically greater than 0.7. After completing all learning objects, participants were provided with a knowledge test that measures their understanding of hardware and software components based on the completed activities. It included ten multiple choice questions. Participants were observed while exploring learning objects and annotations were made with the goal of documenting any performed action and manipulated objects. This measurement was used to assess participants’ curiosity by counting the number of actions and objects manipulated. A stopwatch was used to calculate the time participants took to complete each learning object. Data was anonymised and analysed using the statistical program IMB SPSS (Statistical Package for the Social Sciences).

5.3 Evaluation

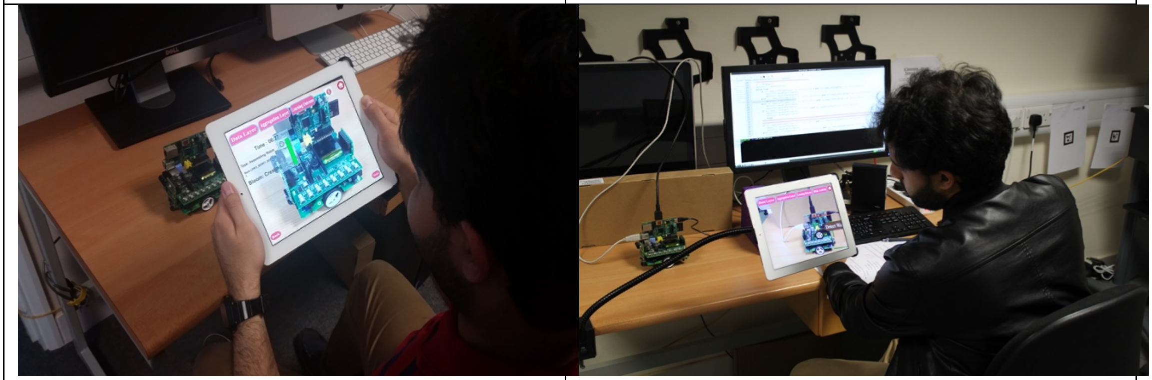
5.3.1 Demographics and preliminary data

The objective of the pre-survey questionnaire was to explore students’ knowledge and background on both groups: experimental (PVM with AR) and paper-based(traditional). A copy of the survey is available in the Appendix A.



(a) Students inspecting robot functionalities using AR

(b) AR shows the next step of assembling the robot



(c) Student looking at the robot using AR

(d) Student looking at the performed action

(e) Student using paper-based instruction

(f) Student exploring robot functionalties using paper-based instruction

Figure 5-2 Students working on AR and paper-based approaches

The total sample of participants was 36, which formed 18 participants (16 males and 2 females) for the experimental group (PVM with AR) and 18 participants for the control group (16 males and 2 females) (Figure 5-2). The experimental group age ranges were: 17-24 years old, which was 33%; 25-30 years old was 17%; 31-35 years old was 39%; and over 35 years old was 11%. The ranges were 56%, 22%, and 22% respectively for the control group with no participants over 35 years old. The level of studies for the experimental group at the time of the experiment was 28% studying an undergraduate course and 72% a postgraduate course, and the courses ranged between computer science (72%) and engineering (28%). In the control group, 50% of the participants were doing an undergraduate course and 50% were doing were postgraduate course, and their courses were computer science (56%) and engineering (44%). In both groups, all participants owned a personal computer and 97% of the participants had a smart device, while only one participant in the control group said that he/she did not have a smart device.

The participants in the experimental group stated their computer expertise as beginners (6%), intermediate (28%), and experts (67%), whereas it was beginner (28%), intermediate (44%), and expert (28%) for the control group. In addition, participants in the experimental group stated their computer programming skills as beginners (6%), intermediate (50%), and experts (44%) while it was beginner (44%), intermediate (39%), and expert (17%) for the control group. About participants' familiarity with object-oriented paradigms, the experimental group had a slightly higher mean value of 3.50 compared to 2.94 for the control group but no statistical difference was found between both groups $t=-1.382$, $p = 0.176$ (Figure 5-3). It is worth noting that for learners to participate in the experiment, the minimum requirements included having basic knowledge in computers, programming languages and object-oriented paradigms.

Caveat: The PVM with AR group was more experienced, and it is possible that this is why the students in this group appear to have assimilated more knowledge.

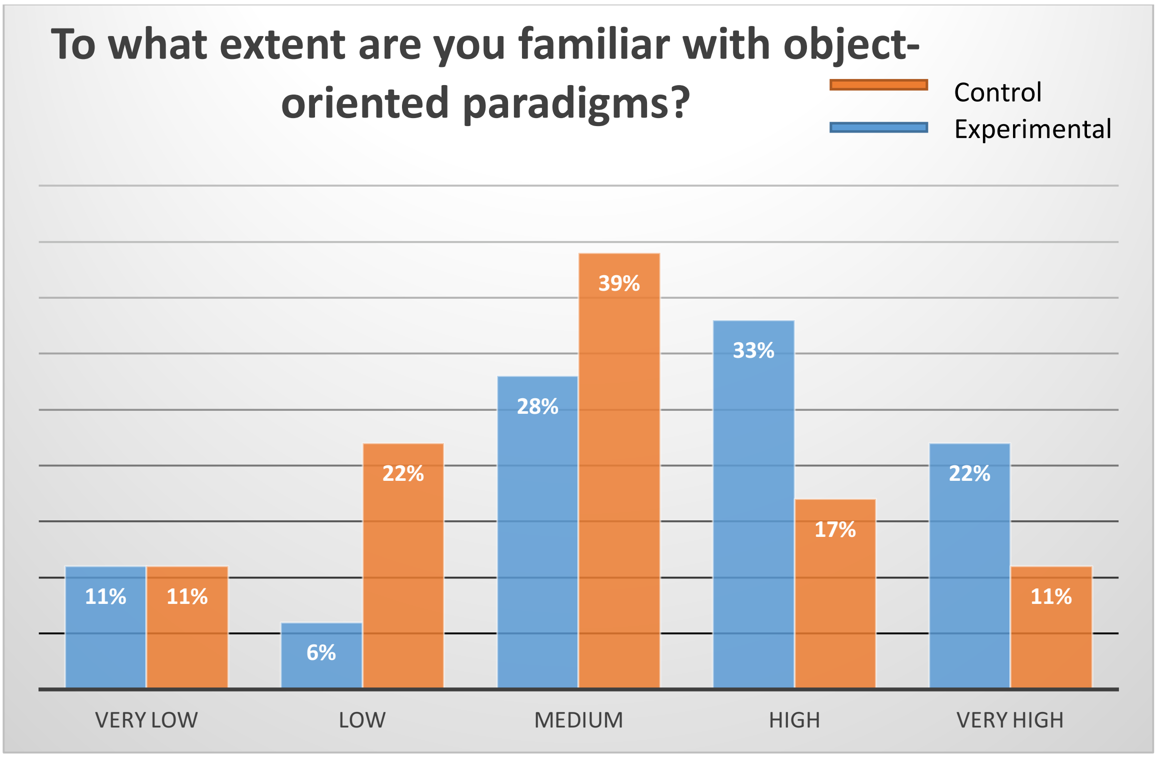


Figure 5-3 Familiarity with the object-oriented paradigm

Moreover, participants declared having experience in a wide range of object-oriented programming languages, where Python in both groups was among the highest with 72% for the control and 44% for the experimental group. Similarly, 50% of participants in the experimental group stated that they had experience with Java and C++, and 39% had familiarity in C, whereas it was Java (39%), C++ (22%) and C (17%) in the control group. LavView and Matlab were stated as other object-oriented programming languages in which participants had skills.

In relation to participants' previous experience in embedded system modules, 44% of the experimental group declared that they studied embedded system modules during their degree compared to 33% in the control group. On the other hand, 56% of the experimental and 67% of the control group said they did not take any embedded system modules (Figure 5-4). For those who studied modules in embedded systems from both groups, they responded to 10 statements, on a 5-point Likert scale, strongly disagree (1) to strongly agree (5), related to their experience with embedded systems (Figure 5-5). 71% of the participants stated ‘neutral’ in terms of easy-to-learn embedded system courses, whereas the remaining disagree (21%) and strongly disagree (7%) for ease of learning. 29% of responders agreed that they would consider embedded systems as an abstract technology, while the same percentage disagreed. Regarding the difficulties of understanding embedded system architecture, 36% of the participants agreed that it is difficult, while 21% of the population disagreed.

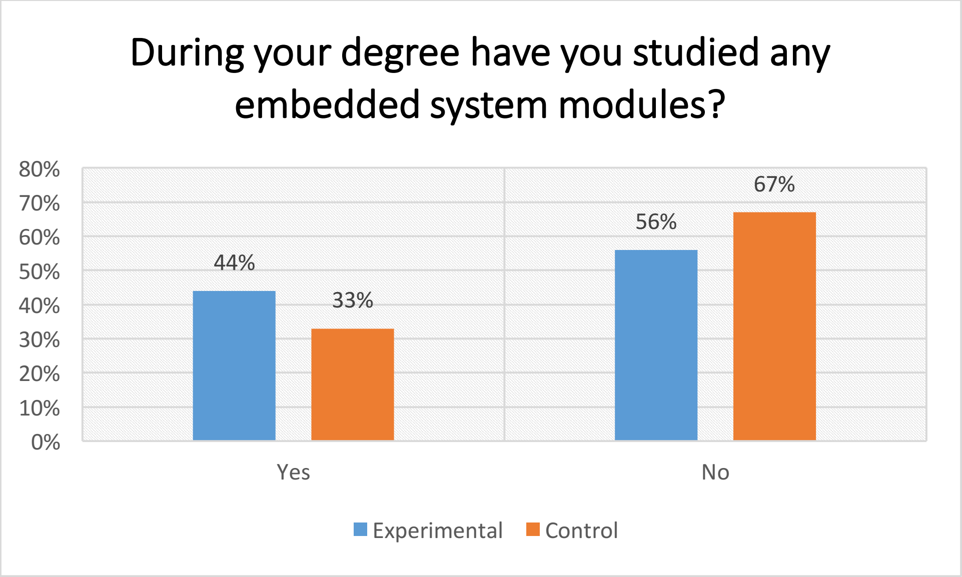


Figure 5-4 Embedded system experience

Half of the participants were neutral regarding the ease of describing the system architecture of embedded systems, whereas 21% either agreed or disagreed. 50 % of students strongly agreed and agreed that they think programming embedded systems is a hard task, while 29% was neutral. In contrast, a solid majority of students (86%) strongly disagreed and disagreed that programming embedded systems does not require a lot of effort. A high percentage of students (64%) were neutral about the ease of explaining how things (hardware and software) work inside embedded systems, while 21% disagreed.

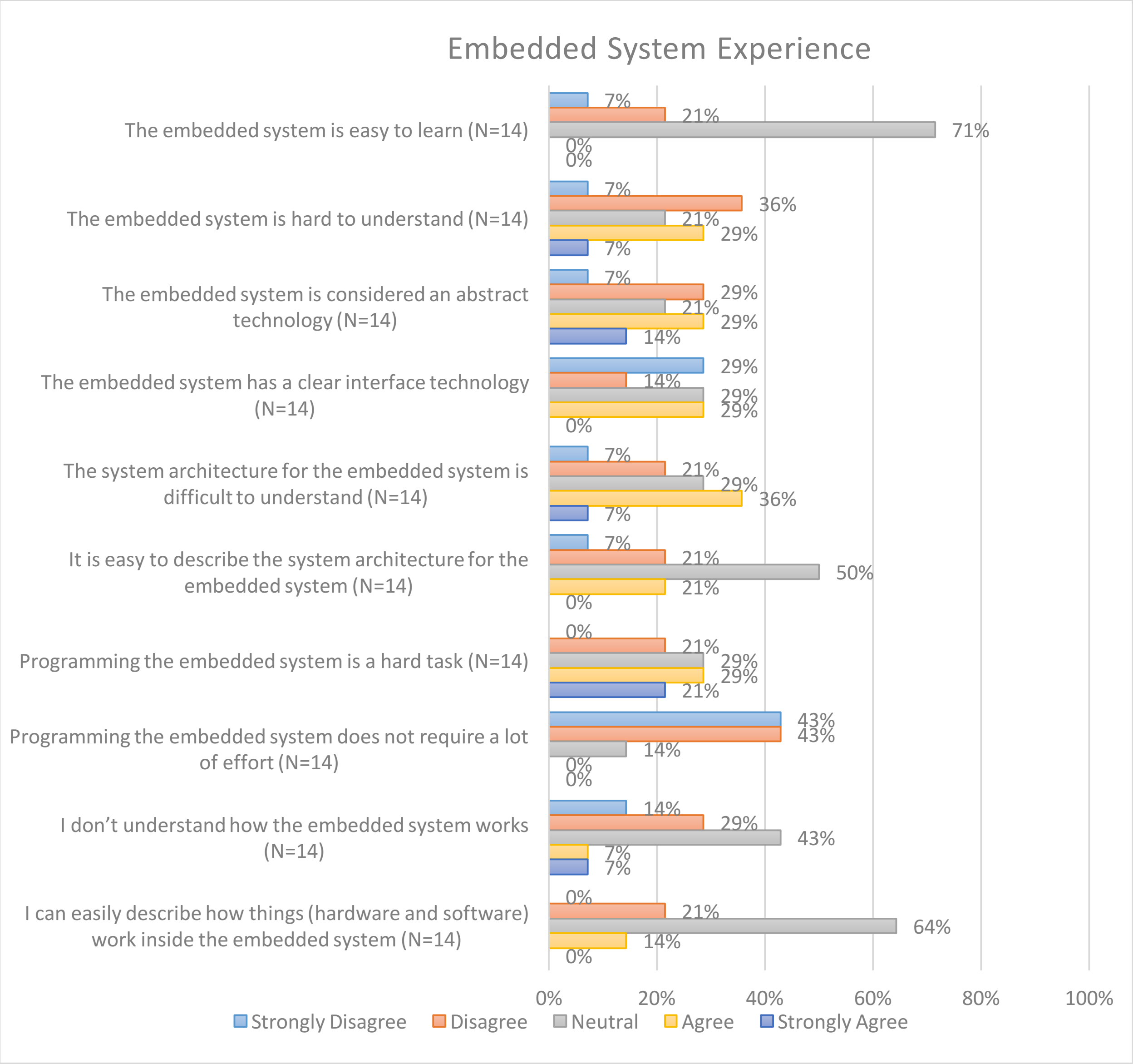


Figure 5-5 Embedded system participants’ experience

In relation to students’ previous experience in assembling or programming an educational mobile robot, the majority of students 89%(control) and 83% (experimental) said that they were not involved in a learning activity that required either assembling or programming a mobile robot during their study (Figure 5-6).

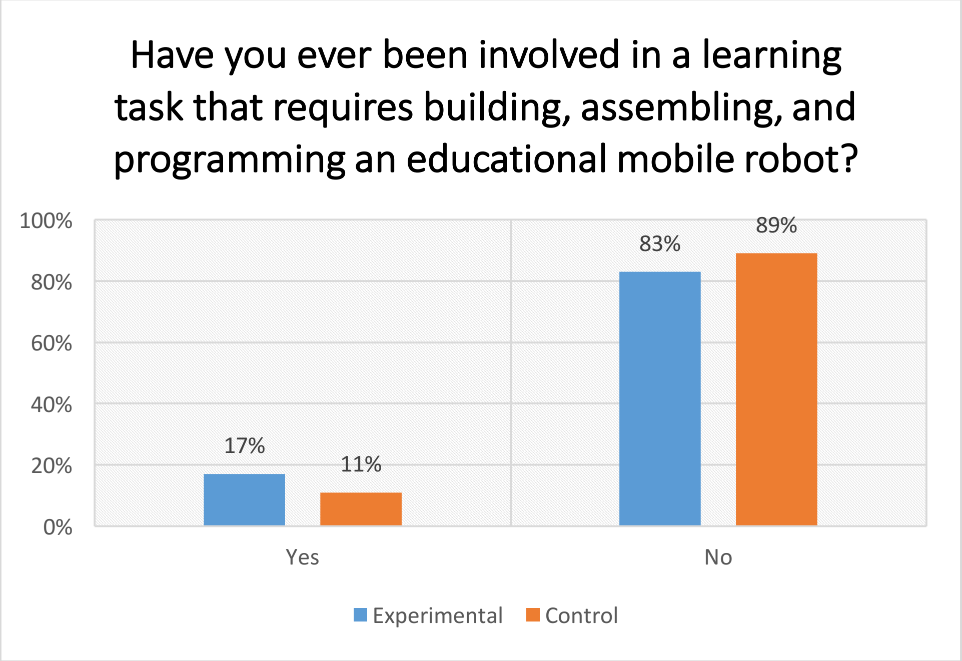


Figure 5-6 Students’ previous experience assembling and programming an educational robot

Regarding technology familiarity, a large number of participants were not at all familiar (44%) and not very familiar (28%) with mixed reality (MR) and augmented reality (AR), whereas a tiny number of participants stated that they were very familiar with MR (6%) and AR (8%). Similarly, 42% of participants said that they were not at all familiar with the Internet-of-Things (IoT), and only 8% was familiar and very familiar. In addition, Raspberry Pi was considered as not at all familiar by 44% of participants and only 8% found it familiar and 6% very familiar. The majority of participants were not at all familiar with mBed (61%), Arduino (61%) and littleBits (75%).

5.3.2 Descriptive Analysis

This section presents the descriptive statistics and tests needed to explore the research variable to test assumptions for inferential analysis and hypothesis testing. Means, medians, standard deviations, and skewness were calculated and reported, and histograms were drawn. Means, medians, and standard deviation describe the central tendency and spread of the data within a variable, and skewness describes the shape of the distribution. The Kolmogorov-Smirnov and Shapiro-Wilk tests of normality were conducted and the results are reported, as well. However, the Shapiro-Wilk test results are preferred, as the test is more appropriate for small samples (< 50) (Shapiro and Wilk, 1965).

The standard error of skewness can be used to determine if the distribution is significantly skewed, or whether the distribution is skewed within the normal range. To find this value, the numerical value for skewness was compared with twice the standard error of skewness. If the value for skewness lies inside the range of between minus twice the skewness standard error and plus twice the skewness standard error, the skewness is considered to be near normality standards; hence, normality is not violated. It is worth considering the sample size when performing such a comparison. In small sample sizes (< 50), skewness and kurtosis values are very sensitive and can vary extensively from negative to positive to perfect normal skews.

5.3.2.1 User objective measurement

Different criteria were used to assess the normality of the experiment variables. Means and standard deviations, broken down by student groups, were reported in Table 5-1. Skewness values show the distributions seem to be skewed in the normal range (-1 to +1). However, for the variable "objects a student manipulated", the skewness value was -1.498 in the PVM with AR, which seems to be beyond the normal limits.

Table 5-1. Descriptive Statistics for Experiment Variables

		M	**Md**	**SD**	**Min**	**Max**	**Sk***
Augmented Reality	Introduction time	03:27	03:15	01:10	01:24	05:30	.075
	Assembling time	07:42	07:12	02:12	04:19	12:08	.461
	Exploring time	14:57	15:01	02:34	10:13	19:30	-.277
	Total time	26:06	26:42	03:11	18:04	30:52	-.872
	Mean time	08:42	08:54	01:03	06:01	10:17	-.872
	Actions a student made	12.56	13.00	1.822	9	15	-.363
	Objects a student manipulated	**6.33**	**7.00**	**1.029**	**4**	**7**	**-1.498**
	Students' post-test result	7.78	8.00	1.396	5	10	-.139
Paper-based Approach	Introduction time	03:19	03:12	01:14	01:16	06:12	.800
	Assembling time	03:13	03:20	00:46	02:06	04:20	-.171
	Exploring time	09:20	07:56	04:12	03:25	18:07	.803
	Total time	15:53	14:41	04:59	09:43	25:17	.739
	Mean time	05:17	04:53	01:39	03:14	08:25	.739
	Actions a student made	6.44	6.00	3.552	2	14	.792
	Objects a student manipulated	3.50	3.00	1.654	1	7	.922
	Students' post-test result	3.67	4.00	1.455	0	6	-.888

Standard Error of Skewness = .536, Normal Skewness Limits: ±1.072

Assessing Normality

Examining the normality test results, the majority of variables had distributions not significantly different from the normal distribution (Table 5-2) . The variable “objects a student manipulated” had a significantly different distribution than normal, $p < .001$, for both tests in the augmented reality group, and $p < .05$ for both tests in the paper-based approach. In this case, an examination of the histograms helps in deciding whether there is a significant violation of normality or not. Histograms, presented in Figure 5-7, suggest non-existence of significant violation of normality. The deviation of normality of the distribution of the variable “objects a student manipulated” is considered normal as it is sensible to consider a higher number of objects manipulated by a student using PVM with AR. Hence, this variable cannot be a violation of normality.

Table 5-2. Kolmogorov-Smirnov/Shapiro-Wilk Tests of Normality for Experiment Variables

	PVM with AR				**Paper-based Approach**			
	K-S		**S-W**		**K-S**		**S-W**	
Experiment Variables	**Statistic**	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**
Introduction time	.163	.200	.955	.504	.228	.014*	.931	.205
Assembling time	.136	.200	.957	.554	.185	.105	.903	.066
Exploring time	.088	.200	.974	.873	.212	.031*	.914	.101
Total time	.124	.200	.942	.313	.179	.130	.904	.068
Mean time	.124	.200	.942	.313	.179	.130	.904	.068
Actions a student made	.152	.200	.942	.309	.164	.200	.902	.063
Objects a student manipulated	.353	.000*	.687	.000*	.230	.013*	.896	.048*
Students’ post-test result	.156	.200	.949	.409	.257	.003*	.908	.079

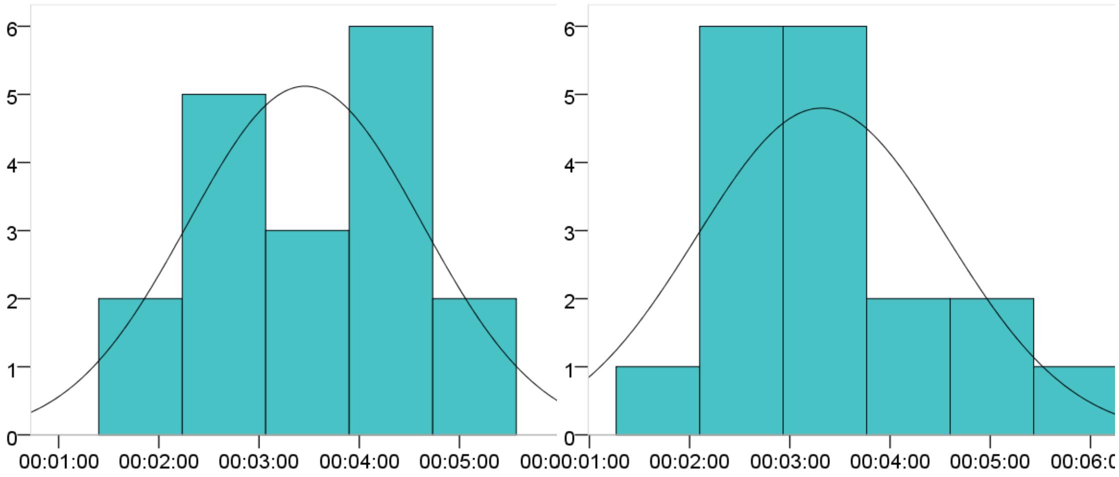
. Significant at α = .05

Figure 5-7. Histograms for User Objective Variables

Introduction Time

PVM with AR

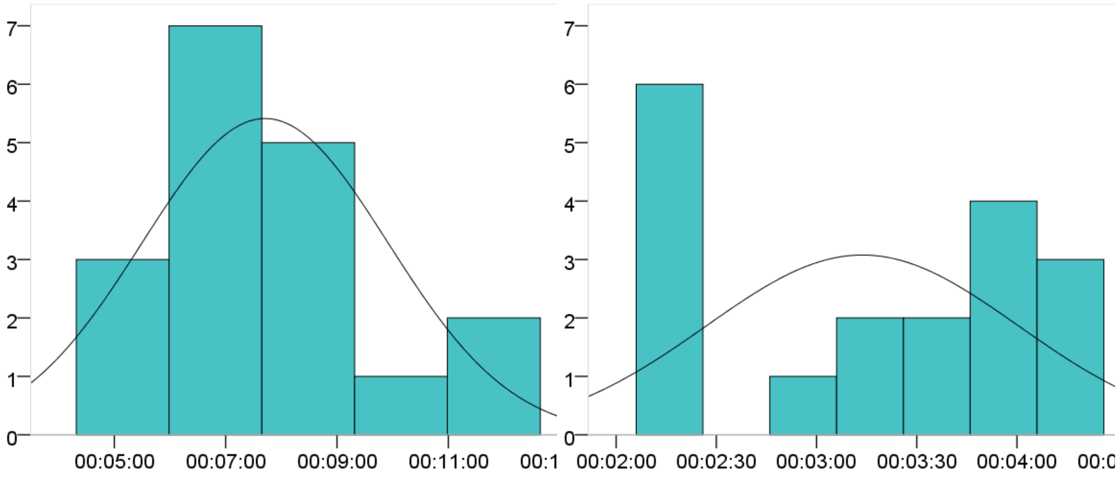
Paper-based Approach



Assembling Time

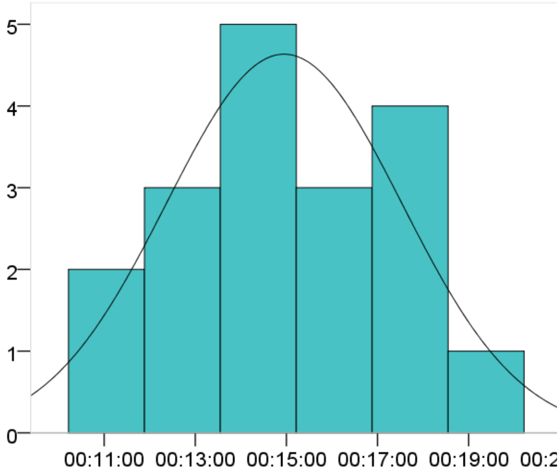
PVM with AR

Paper-based Approach

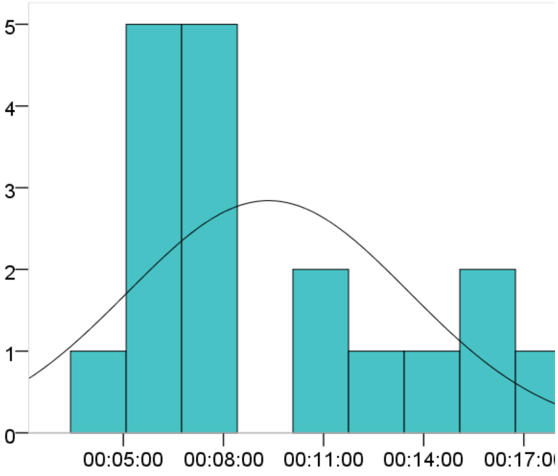


Exploring Time

PVM with AR

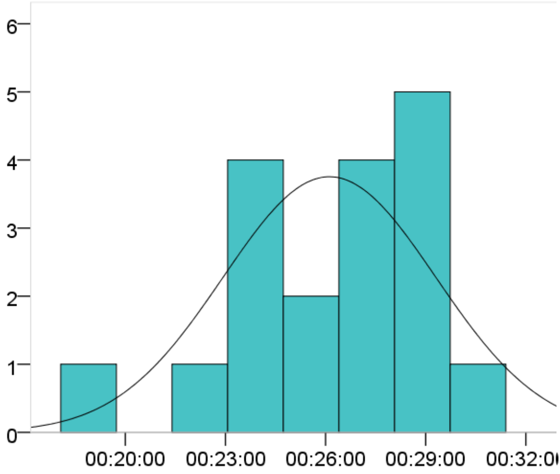


Paper-based Approach

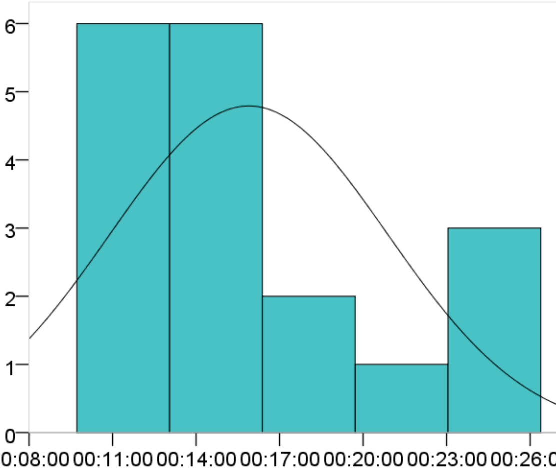


Total Time

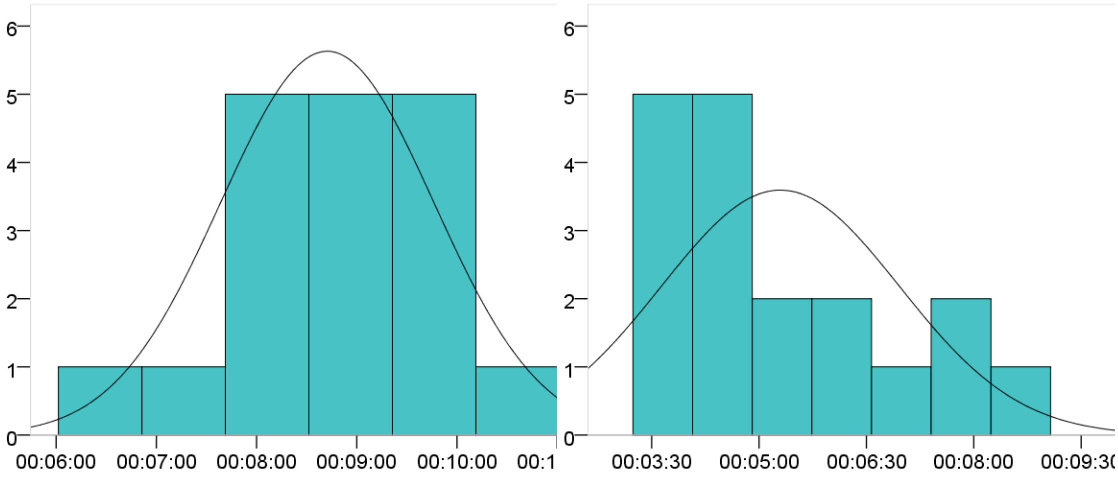
PVM with AR



Paper-based Approach



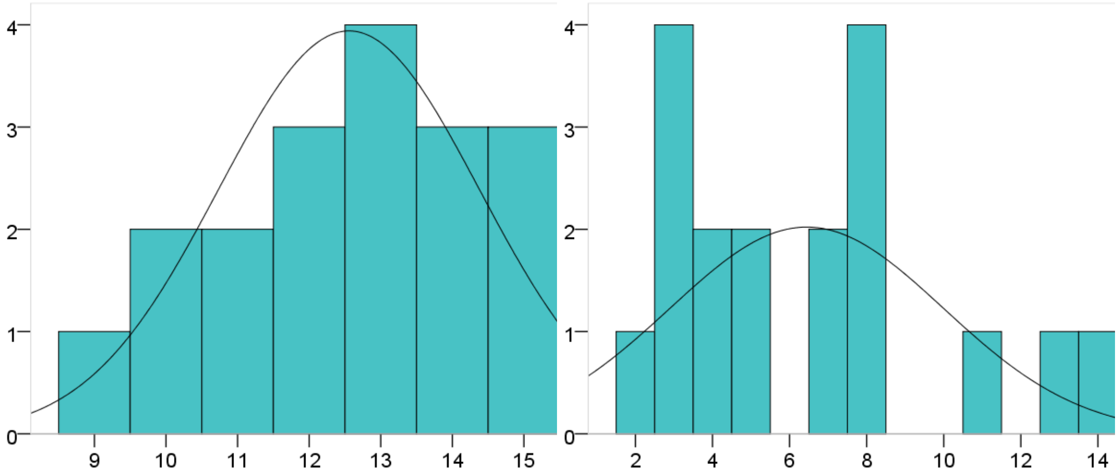
Mean Time



PVM with AR

Paper-based Approach

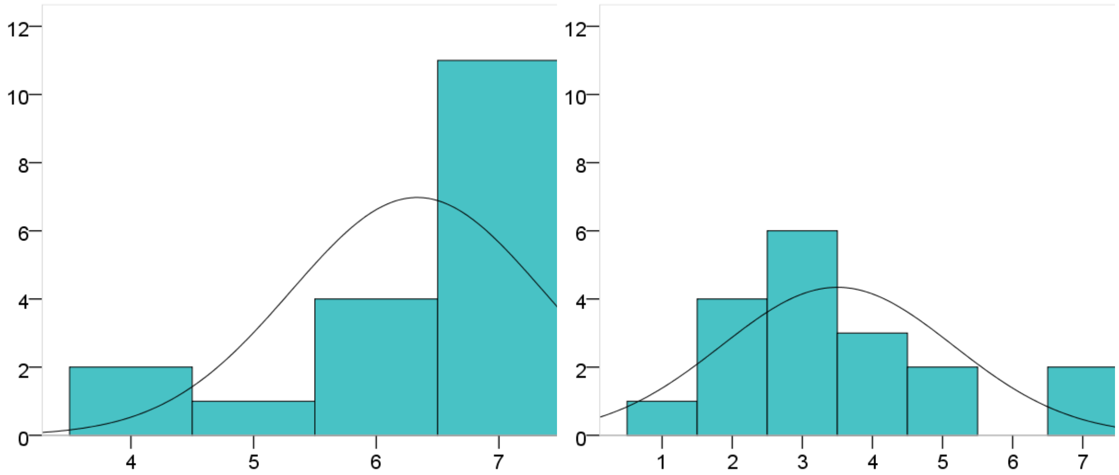
Actions a Student Made



PVM with AR

Paper-based Approach

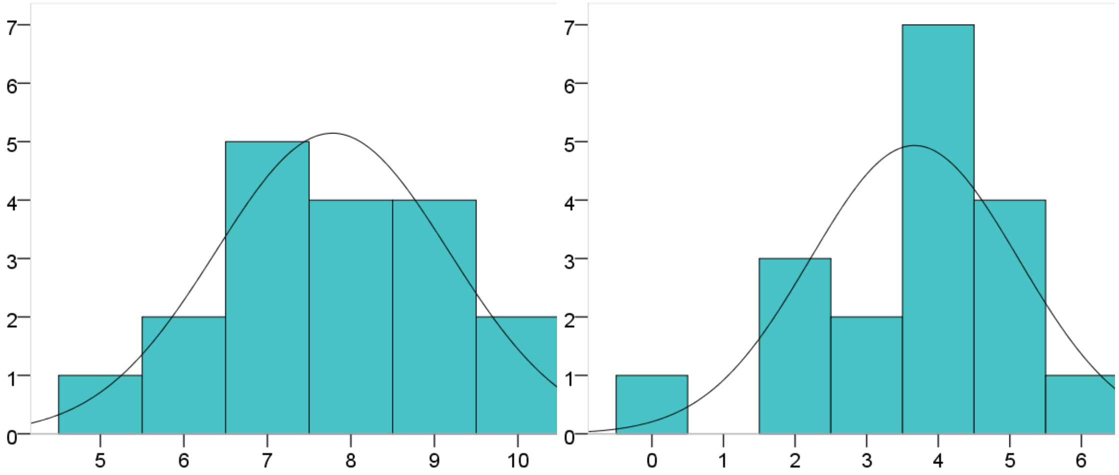
Objects a Student Manipulated



PVM with AR

Paper-based Approach

Students' Post-Test Result



PVM with AR

Paper-based Approach

5.3.2.2 User subjective measurement

The post-questionnaire contained five assessment constructs: cognitive overload, effectiveness, usefulness, enjoyment, and competence. Each construct was measured by a number of items (questions). For further analysis of the questionnaire, a single measure of each factor was calculated by averaging the items in each factor. Before creating the new average variables, the internal consistency of items under each factor was assessed using the reliability coefficient Cronbach's alpha (α). Reliability coefficients are reported in Table 5-3 for the five factors and they all show a good to excellent internal consistency; all reliability coefficients exceeded the accepted threshold of .7. Therefore, using the averages of each factor items is reliable.

Table 5-3. Descriptive Statistics & Reliability Measures for User Subjective Factors

Group	Factors	M	Md	SD	Min	Max	Sk*	α
PVM with AR	Cognitive overload	2.27	2.00	.904	1.17	4.00	.762	.797
	Effectiveness	4.28	4.50	.771	2.50	5.00	-1.388	.680
	Usefulness	6.33	6.50	.625	4.71	7.00	-1.314	.871
	Enjoyment	6.35	6.64	.901	3.43	7.00	-2.419	.961
	Competence	6.03	6.30	.956	3.60	7.00	-1.373	.853
Paper-based Approach	Cognitive overload	3.15	3.25	1.334	1.33	5.50	.212	.845
	Effectiveness	3.78	3.88	.947	2.00	5.00	-.484	.875
	Usefulness	4.84	5.14	1.592	2.00	6.86	-.454	.935
	Enjoyment	5.14	5.64	1.515	2.71	6.86	-.399	.926
	Competence	5.47	5.50	.965	3.80	7.00	.010	.706

Standard Error of Skewness = .536, Normal Skewness Limits: ±1.072

Assessing Normality

The normality of factors was assessed using different criteria including Kolmogorov-Smirnov and Shapiro-Wilk's normality test, skewness measures, and histograms. Skewness normal limit is ±1.072. The skewness measures reported in Table 5-3 reveals that, for the PVM with AR group, the effectiveness, usefulness, enjoyment, and competence factors reported significant deviations from the normal skewness limits. Moreover, the results of normality tests, reported in Table 5-4 , show that—in augmented reality—effectiveness and enjoyment were not close to the normal distributed, $p < .05$, and in Paper-based Approach Usefulness also was not close to the normal distributed, $p < .05$. However, as mentioned earlier, these results are only a guide to decide whether to consider a violation of normality or not. Examining the histograms in Figure 5-8 is necessary for making a decision. According to the Shapiro-Wilk test results and examination of the histograms, the distributions of effectiveness, usefulness, enjoyment, and competence looked far from normal distribution. Histograms suggested the existence of outliers on the lower tails of the distributions. This was a violation of normality and a transformation was applied to these factors to make them closer to normal distribution. The most appropriate transformation was performed by applying the following equation:

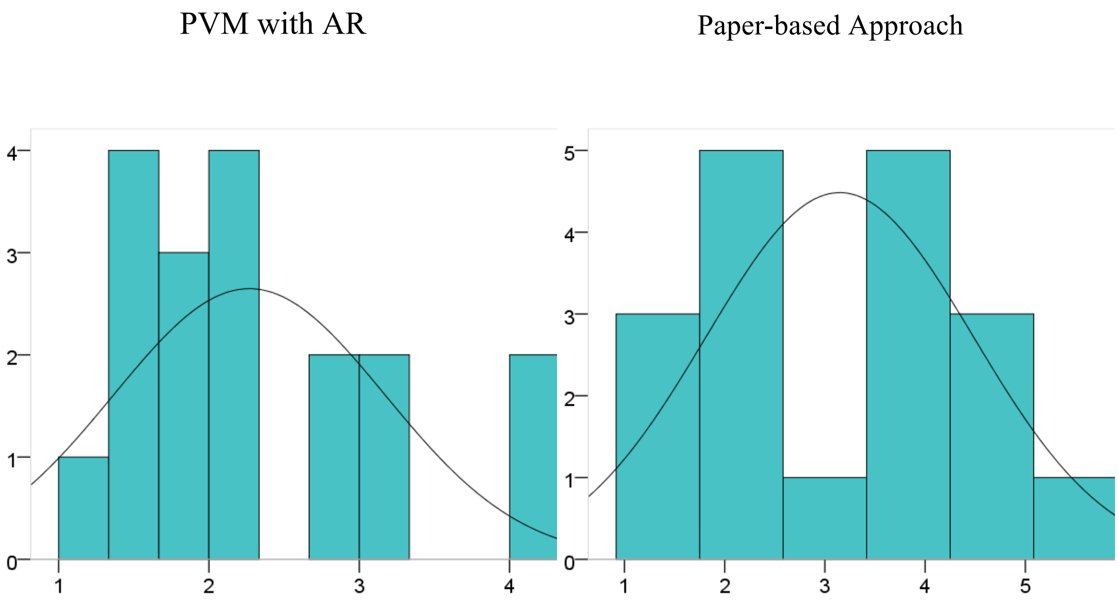
NewVar = SQRT(C – OldVar), where C is a constant determined as the maximum value + 1.

Table 5-4. Kolmogorov-Smirnov/Shapiro-Wilk Tests of Normality for Factors

	PVM with AR				**Paper-based Approach**			
	K-S		**S-W**		**K-S**		**S-W**	
	Statistic	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**
Cognitive overload	.172	.167	.900	.058	.174	.158	.937	.258
Effectiveness	.224	.017*	.803	.002*	.135	.200	.938	.264
Usefulness	.177	.143	.868	.017*	.211	.033*	.908	.078
Enjoyment	.305	.000*	.685	.000*	.190	.086	.868	.016*
Competence	.181	.121	.858	.012*	.088	.200	.962	.641

Figure 5-8. Histograms of User Subjective Experience Factors

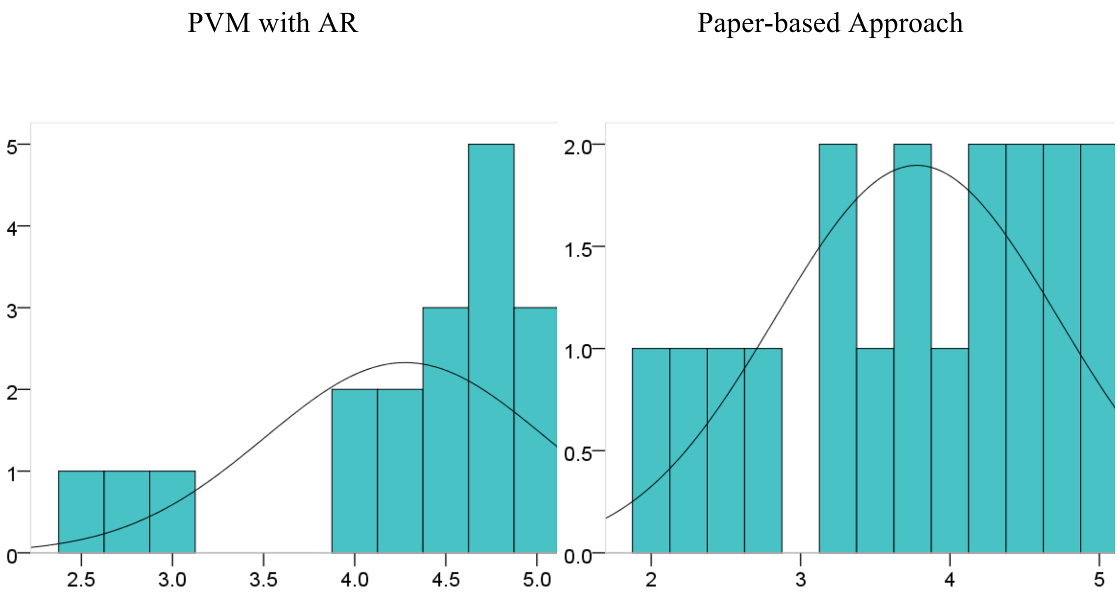
Cognitive Overload



PVM with AR

Paper-based Approach

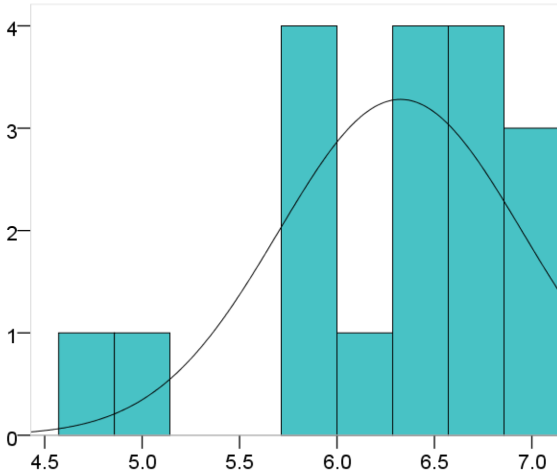
Effectiveness

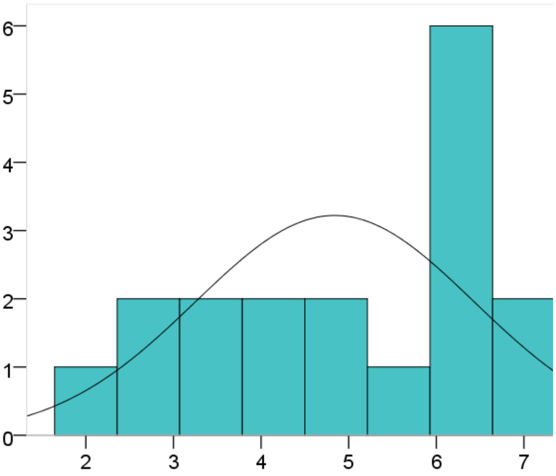


PVM with AR

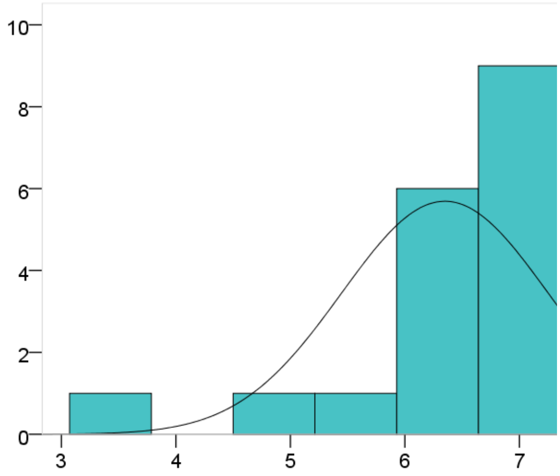
Paper-based Approach

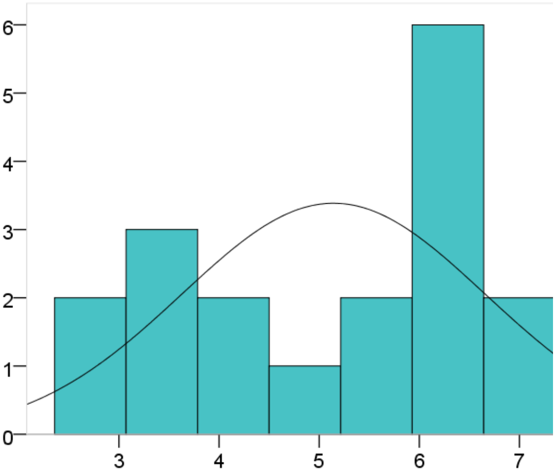
Usefulness


PVM with AR

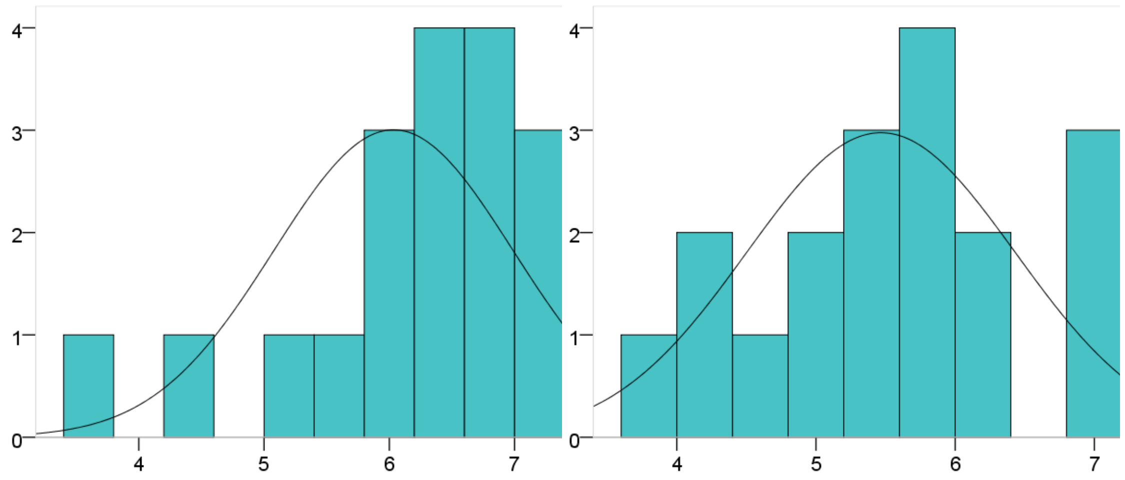

Paper-based Approach

Enjoyment


PVM with AR


Paper-based Approach

Competence



PVM with AR

Paper-based Approach

Transformed Factors

Applying the transformation to the negatively skewed factors, they were normalised and the Kolmogorov-Smirnov/Shapiro-Wilk tests of normality were rerun and the results are reported in Table 5-5. The results revealed that the distributions of the transformed variables were not significantly far from normality. Therefore, the transformed variables can be used in further statistical analysis, paying attention to the final interpretation of the results.

Table 5-5. Kolmogorov-Smirnov/Shapiro-Wilk Tests of Normality for Transformed Factors

	PVM with AR				**Paper-based Approach**			
	K-S		**S-W**		**K-S**		**S-W**	
	Statistic	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**
Effectiveness	.159	.200	.930	.192	.102	.200	.952	.454
Usefulness	.176	.147	.941	.296	.177	.142	.942	.317
Enjoyment	.179	.131	.913	.097	.165	.200	.901	.059
Competence	.128	.200	.957	.542	.198	.061	.886	.033

5.3.3 Learning Improvement

After completing the learning objects (activities), the results of the knowledge test for the students in both groups were analysed. A two- samples independent t-test was conducted to compare the mean test scores between students who used PVM with AR and those who used the paper-based approach. Results are reported in Table 5-6. The test revealed a significant difference between the mean test scores of students from both groups, $t = 8.651$, $p < .001$. Students who used PVM with AR ($M = 7.78$, $SD = 1.396$) had statistically significant higher mean test scores than students who used the paper-based approach ($M = 3.67$, $SD = 1.455$). The results show that students who use PVM with AR improve their learning outcome compared to students who use the paper-based approach. These results support hypothesis 2 which states that "*Using the PVM framework within structured AR learning environments (specified in 1) would make the technical activities/information from the embedded devices visible and meaningful to the student, which would lead to improved learning outcomes for complex learning activities and improve the learners' awareness of the activities inside structured PVM with AR learning environments.*"

Table 5-6. Independent Samples T-Test Results for “Students’ post-test result”

Group Statistics				Independent Samples Test	
PVM with AR		Paper-based approach			
M	SD	M	SD	t	Sig.
7.78	1.396	3.67	1.455	8.651	< .001

5.3.4 Learning Performance

Throughout the experiment, the completion time for each learning object was measured to define the amount of time required for participants to solve the task. Two samples independent t-tests were conducted to compare the mean time for each task, total, and average time of the tasks between students who used PVM with AR and those who used the paper-based approach. The results were reported in Table 5-7. The test revealed a significant difference between the mean time for the assembling task and exploring task, *p* < .001. That is, students who used PVM with AR had statistically significant higher mean time for the assembling activity (M = 7 minutes and 42 seconds) than students who used the paper-based approach (*M* = 3 minutes and 13 seconds). Students who used PVM with AR had statistically significant higher mean time for the exploring activity (*M* = 14 minutes and 57 seconds) than students who used the paper-based approach (*M* = 9 minutes and 20 seconds). No significant difference was found in the mean time of the introduction activity for students from both groups, *p* = .727.

The test revealed a significant difference between both groups of students in the mean total time, *t* = 7.319, *p* < .001. The mean total time taken by students who used augmented reality with PVM (*M* = 26 minutes and 6 seconds) was significantly higher than the mean total time taken by students who used the paper-based approach (*M* = 15 minutes and 53 seconds). Similarly, the mean time taken by students who used PVM with AR for all tasks (*M* = 8 minutes and 42 seconds) was significantly higher than the mean time taken by students who used the paper-based approach (*M* = 5 minutes and 17 seconds).

The results showed that students who used PVM with AR do not acquire new knowledge more quickly compared to students who use the paper-based approach. Rather, students who used PVM with AR spend more time than students who use the paper-based approach to acquire knowledge. As can be seen, these results do not support hypothesis 3, which states that "*such structured PVM with AR learning environments (specified in 1) would enable learners to acquire new knowledge more quickly and with fewer misunderstandings.*

Table 5-7. Independent Samples T-Test Results for Tasks and Total Time

	Group Statistics				**Independent Samples Test**	
	PVM with AR		**Paper-based approach**			
Time*	**M**	**SD**	**M**	**SD**	**t**	**Sig.**
Introduction time	0:03:27	0:01:10	0:03:19	0:01:14	.352	.727
Assembling time	0:07:42	0:02:12	0:03:13	0:00:46	8.107	< .001
Exploring time	0:14:57	0:02:34	0:09:20	0:04:12	4.812	< .001
Total time	0:26:06	0:03:11	0:15:53	0:04:59	7.319	< .001
Mean time	0:08:42	0:01:03	0:05:17	0:01:39	7.319	< .001

. Time Stamp: h:mm:ss

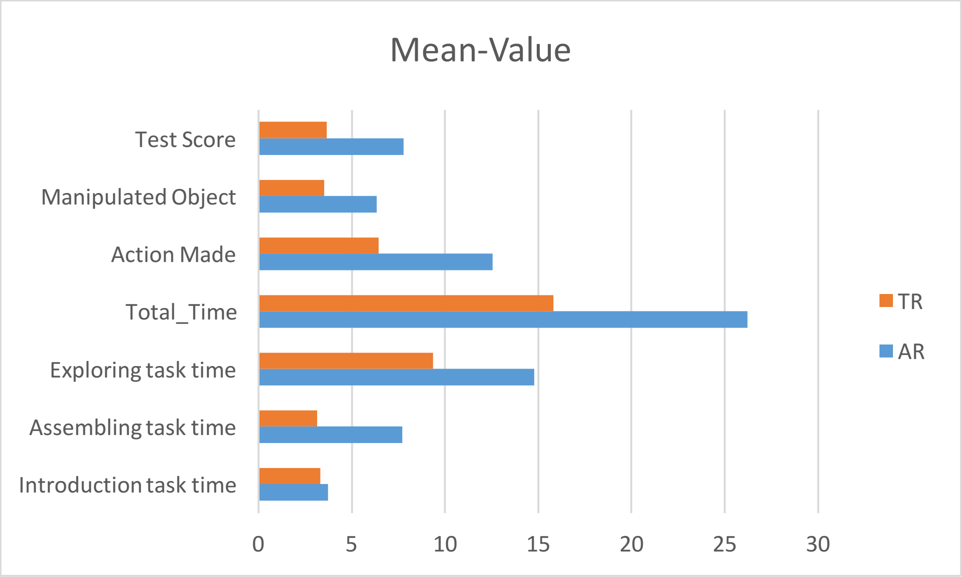


Figure 5-9 Learning effectiveness assessment results of the experimental and the control group

5.3.5 Students' Curiosity

Students' curiosity was measured by two observed factors: the number of actions student made and the number of objects manipulated while performing the learning object task (Exploring task). Similarly to previous constructs, a t-test was conducted to compare the mean *number of actions a student made* and the mean *number of objects students manipulated* between students who used PVM with AR and those who used the paper-based approach. The results were reported in Table 5-8. The test revealed a significant difference between both groups of students in the mean number of actions students made and the mean number of objects students manipulated, $p < .001$. That is, students who used PVM with AR had significantly higher mean number of actions ($M = 12.56$, $SD = 1.822$) and higher mean number of objects manipulated ($M = 6.33$, $SD = 1.029$) than students who used the paper-based approach who had lower mean number of actions made ($M = 6.44$, $SD = 3.552$) and lower mean number of objects manipulated (*M* = 3.50, *SD* = 1.654). In addition, results from the number of objects students manipulated revealed that PVM with AR showed high-frequency use of sensors that interact with the environment and their sensed value is invisible, such as line sensors (70%), light sensors (80%), and infra-red (IR) sensors (85%) compared to the paper-based approach, which had 15%, 25%, and 40%, respectively (Figure 5-10). In contrast, the use of sensors and actuators that have a physical appearance or action were slightly similar in both groups.

Therefore, it can be concluded that PVM with AR increased the curiosity of the learners compared to the paper-based approach, which supports hypothesis 6, which states that "*using structured PVM with AR learning environments (specified in 1) would increase the curiosity of the learners over traditional laboratory learning environments.*"

Table 5-8. Independent Samples T-Test Results for Students Curiosity

	Group Statistics					
	PVM with AR		**Paper-based approach**		**Independent Samples Test**	
	M	**SD**	**M**	**SD**	**t**	**Sig.**
Actions a student made	12.56	1.822	6.44	3.552	6.495	< .001
Objects a student manipulated	6.33	1.029	3.50	1.654	6.171	< .001

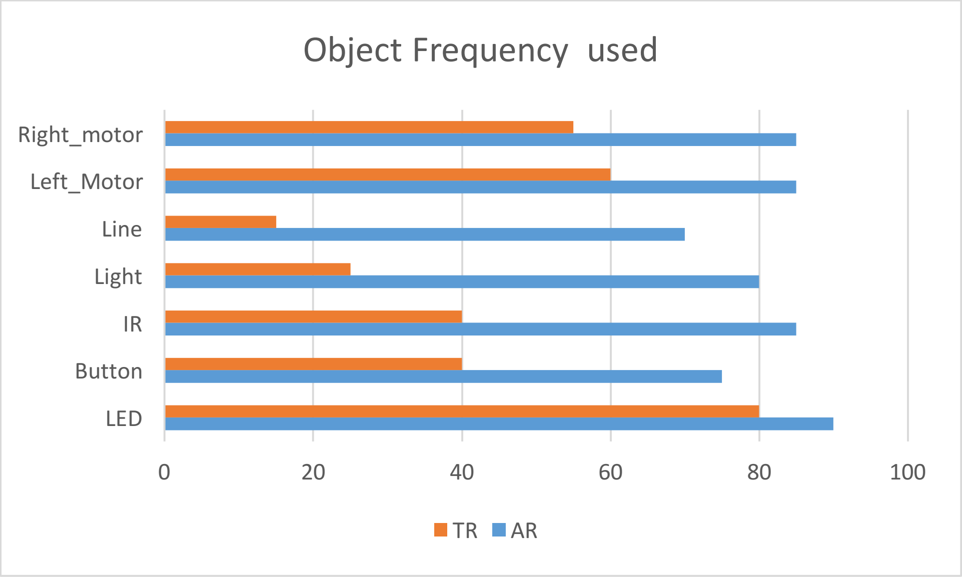


Figure 5-10 Objects-frequency used in exploring the learning object

5.3.6 Correlation Analysis

Previous constructs were examined to reveal any correlated factors that improve students' learning outcomes based on the approach used. Thus, the test score construct correlated with the completion time and curiosity constructs.

5.3.6.1 Relationship between Task Time and Test Scores

Pearson correlation coefficients were computed and reported in Table 5-9 for the time of the three tasks and total time versus students' post-test scores. The coefficients reported were calculated for both groups of students and for total participants as well. The analysis revealed that there was a significant association between total task time and students' post-test scores for PVM with AR students, $r = .509$ and $p < .05$. Another significant correlation was found between exploring time and students' post-test scores for students using the paper-based approach, $r = .568$ and $p < .05$.

That is, for students PVM with AR, the more total time of tasks they take, the higher their post-test scores. On the other hand, for students using the paper-based approach, the more time they take exploring, the higher their post-test scores.

Table 5-9. Pearson Correlation Coefficients of Task Time vs. Test Scores

	Group	**Introduction Time**	**Assembling Time**	**Exploring Time**	**Total Time**
Assembling time	PVM with AR	.080			
	TR	.193			
Exploring time	PVM with AR	.282	-.427		
	TR	.306	.225		
Total time	PVM with AR	.650**	.377	.617**	
	TR	.537*	.393	.954**	
Students' post-test result	PVM with AR	.426	-.025	.457	.509*
	TR	.091	-.416	.568*	.437

. Correlation is significant at the 0.05 level (2-tailed).
****. Correlation is significant at the 0.01 level (2-tailed).***

5.3.6.2 Relationship between Learners Curiosity and Test Scores

Similarly, Pearson correlation coefficients were computed and reported in Table 5-10 for the number of actions a student made and the number of objects a student manipulated versus students' post-test scores. The analysis revealed a significant correlation between the number of actions a student made and students' post-test scores, for both groups of students, $p < .05$.

Table 5-10. Pearson Correlation Coefficients of Learners' Curiosity vs. Test Scores

	Group	**Actions a student made**	**Objects a student manipulated**
Objects a student manipulated	PVM with AR	.429	
	TR	.751**	
Students' post-test result	PVM with AR	.560*	.341
	TR	.565*	.464

. Correlation is significant at the 0.05 level (2-tailed).
****. Correlation is significant at the 0.01 level (2-tailed).***

5.3.7 Students Workload

After finishing the experiment, students were asked to rate their experience related to cognitive overload factors, which assess task load. A two samples -independent t-test was conducted to compare the mean of cognitive overload and their sub-factors between students who used PVM with AR and those who used the paper-based approach; the results are reported in Table 5-11. The test revealed a significant difference between the mean cognitive overload of students from both groups, *t* = -2.315, *p* = .028. That is, students who used PVM with AR had statistically significant lower mean cognitive overload than students who used the paper-based approach. In addition, the result shows that the PVM with AR approach students had statistically significant difference between the mean of physical demand (*t* = -2.666, *p* = .012) and frustration (*t* = -2.172, *p* = .040) compared to the paper-based approach. On the other hand, mental demand, temporal demand, performance and effort had slightly higher workload for those who using the paper-based approach, but no statistically significant difference was found between both groups. Therefore, it can be concluded that PVM with AR reduces cognitive overload compared to paper-based approach. These results support hypothesis 4 which states that "*using structured PVM with AR learning environments (specified in 1) would provide assistance that reduces load in learning*".

Table 5-11. Independent Samples T-Test Results for Cognitive Overload

	Group Statistics					
	PVM with AR		**Paper-based approach**		**Independent Samples Test**	
	M	**SD**	**M**	**SD**	**t**	**Sig.**
Cognitive overload	**2.27**	.904	3.15	1.334	-2.315	.028

Mental demand	2.56	1.504	3.56	1.688	-1.877	.069
Physical demand	1.89	1.132	3.17	1.689	-2.666	.012
Temporal demand	2.61	1.335	3.33	1.782	-1.376	.178
Performance	2.22	1.060	2.94	1.731	-1.509	.140
Frustration	1.44	.784	2.44	1.790	-2.172	.037
Effort	2.89	1.676	3.44	1.977	-.909	.370

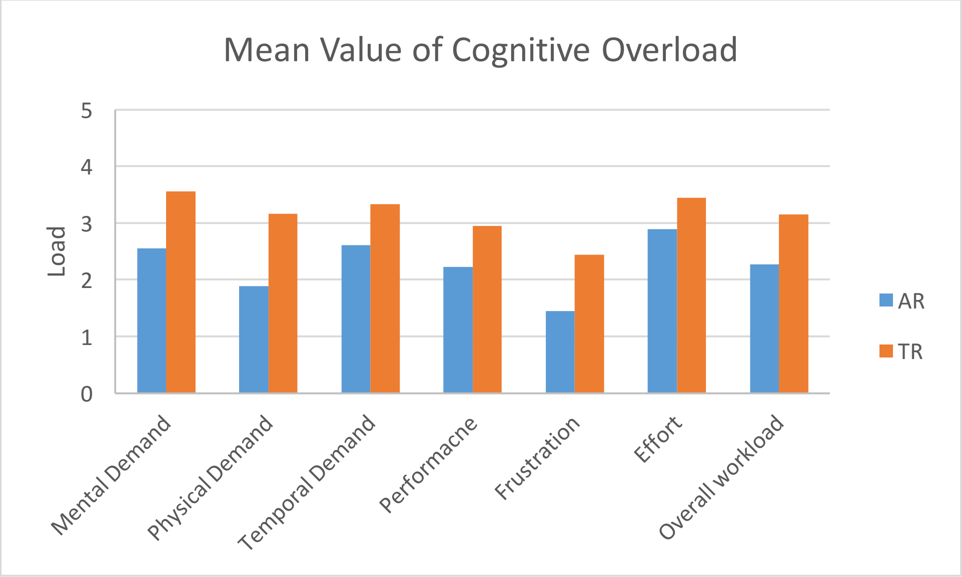


Figure 5-11 NASA TLX mean score for each item for evaluating cognitive overload

5.3.8 Learning Activity Evaluation

Subjects were asked to evaluate the available tools used for doing laboratory hands-on learning activities in terms of enjoyment, competence, usefulness and effectiveness. Therefore, a two-samples independent t-test was conducted to compare the mean score of learning activity enjoyment, perceived competence, usefulness, and effectiveness between students who used PVM with AR and those who used the paper-based approach; results were presented in Table 5-12.

The tests revealed that the two groups of students were statistically significant with respect to the mean score of usefulness and enjoyment, $p < .01$. That is, students who used PVM with AR had significantly higher mean score of usefulness (*M* = 6.33, *SD* = .625) and higher mean score of enjoyment (*M* = 6.35, *SD* = .901) than students who used the paper-based approach who had lower mean score of usefulness (*M* = 4.84, *SD* = 1.592) and lower mean score of enjoyment (*M* = 5.14, *SD* = 1.515). No significant differences were found between both groups of students with respect to the mean score of learning activity effectiveness and perceived competence, $p > .05$.

The test results show that PVM with AR significantly increases learning activity enjoyment and usefulness compared to paper-based. Clearly, these results support two factors in hypothesis 5, which states that "*using structured PVM with AR learning environments (specified in 1) would increase students' enjoyment, perceived competence, and usefulness while performing laboratory hands-on-activities*".

Table 5-12. Independent Samples T-Test Results for Learning Activity Enjoyment, Perceived Competence, Usefulness, and Effectiveness

	Group Statistics				**Independent Samples Test**			
	PVM with AR		**Paper-based approach**		**Original**		**Transformed**	
	M	**SD**	**M**	**SD**	**t**	**Sig.**	**t**	**Sig.**
Effectiveness	4.28	.771	3.78	.947	1.737	.091	-1.637	.111
Usefulness	6.33	.625	4.84	1.592	3.681	.001	-3.835	.001
Enjoyment	6.35	.901	5.14	1.515	2.903	.007	-3.176	.003
Competence	6.03	.956	5.47	.965	1.770	.086	-1.650	.108

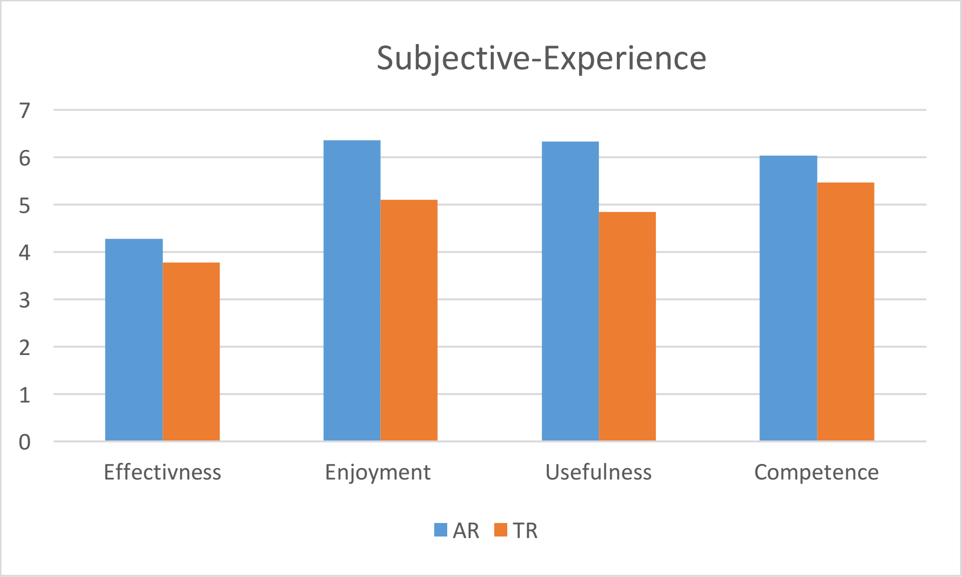


Figure 5-12 Subject experience based on each method

5.3.9 PVM with AR user experience

On completing the experiment, the PVM with the AR group was asked to provide their perspective on the system by completing a user-experience questionnaire (UEQ). The study conducted an analysis of the UEQ questionnaire by calculating the means of the six scales (Table 5-13). The UEQ produced no overall score for the user's experiences. As a result of the UEQ's development through factor analysis, it was insensible to build an overall score by calculating the mean overall scales because it was not possible to interpret this value appropriately(Laugwitz et al., 2008). The scales ranged from +3, which was the positive extreme, to -3, which was the negative extreme. Common answer tendencies in those questionnaires showed that people typically avoided the extremes, with values in the range of 1.5 and 2 indicating a significantly desirable quality.

Table 5-13 UEQ scale measurement

Scale	Intended measurement

Attractiveness	What is the overall impression of the PVM with AR system? Do students like or dislike it?
Perspicuity	Do students become familiar with the PVM with AR system easily?
Efficiency	Can students solve the learning activity tasks with the PVM and the AR system without unnecessary effort.
Dependability	Do students feel in control of the interaction while using the PVM with AR system?
Stimulation	Do students feel excited and motivated to use the PVM with AR system?
Novelty	Is the PVM with AR system innovative and creative and holds students' attention?

The UEQ's findings illustrated that the scores of all scales that described attractiveness, stimulation, novelty, dependability, perspicuity and efficiency tended to be positive, with each scale scoring beyond the value of +1.5. The limited range of the confidence intervals implied that the measured scales were fairly accurate. (bars in Figure 5-13) indicate that the measured scale means were quite accurate. It can be seen that these results support hypothesis 8, which states that "*learners' user experience while using structured PVM with AR learning environments (specified in 1) would show system acceptance*".

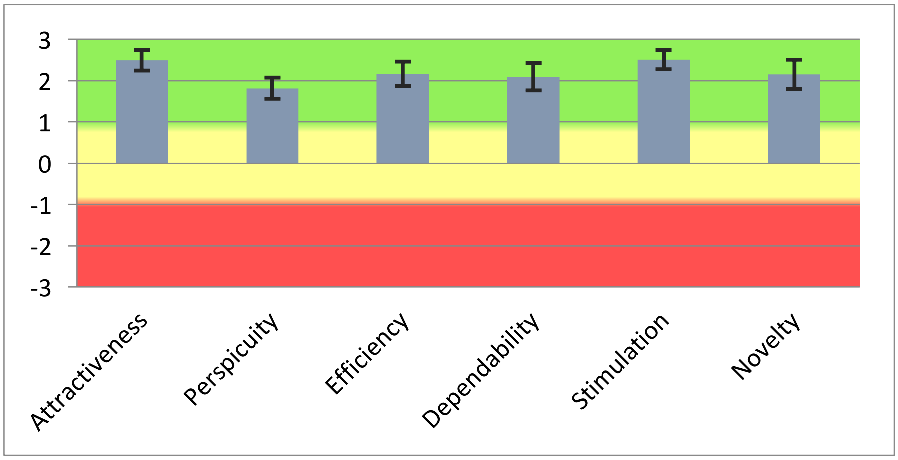


Figure 5-13 Results of UEQ factors

5.4 Summary

This chapter started by reviewing the current evaluation strategy used in the augmented reality field. Then, it introduced the experimental design of the study, which relates to the assembling and exploring of embedding computing learning activities. The experimental design begun by stating the hypotheses (H2, H3, H4, H5, H6 and H8) presented in Chapter one, which needs to be evaluated within the user study and mapped the results to the proposed hypotheses. The user study compared two educational learning approaches. The first approach was the PVM with AR system, whereas the paper-based was the second approach. A group of 36 students were divided randomly in both approaches. Then, the chapter presented the results and findings of the evaluation based on a depth-statistical analysis and mapped the results to the proposed hypotheses to evaluate the effectiveness of the proposed PVM model against alternative traditional methods. The PVM with AR system displayed several benefits when compared to traditional approaches. Learners' outcomes improved; task cognitive overload was reduced; learners' curiosity, usefulness and enjoyment increased; learners showed positive impressions toward the system and learners spent more time assembling and exploring learning objects. The discussion of the result will be presented in Chapter Seven.

The next chapter introduces the second experiment evaluation of designing behaviour robot-based learning activity. It aims to examine the PVM model with more complex learning activity based on the high-level of Bloom's Taxonomy.

Chapter Six

6 Real-Time Feedback for Controlling Embedded Computing Learning Activity

> *"Intelligence is the source of technology. If we can use technology to improve intelligence, that closes the loop and potentially creates a positive feedback cycle."*
>
> - Eliezer Yudkowsky (1979 – present)

The previous chapter presented an experimental evaluation for assembling and exploring embedded computing activity where learners participated in three learning objects, namely introduction, assembling and exploring. The learning outcomes these corresponded to three levels of Bloom's Taxonomy: knowledge, comprehension and application. These levels are the lowest on the pyramid of Bloom's Taxonomy. However, to examine the PVM model at a higher level of Bloom's Taxonomy, a behaviour-based mobile robot learning activity experiment was designed to extract the pedagogical value of PVM with AR in a real-time system. The learning outcomes of the experiment corresponded to analysis and synthesis on Bloom's Taxonomy. Thus, learners were expected to have prior programming language skills as the level of learning for this experiment is associated with a high level of Bloom's Taxonomy.

This chapter presents the experimental evaluations for the second learning activity presented in Chapter four, that is, controlling and designing a behaviour-based robot based on modularised embedded computing components. Two educational systems were employed, to be compared in the user evaluation: PVM with AR, and a traditional learning environment. This revealed the learning effectiveness of using PVM with AR, compared to the traditional engineering laboratory method. The chapter starts by describing the experimental design used to gather evidence of the value of the concepts proposed in this thesis, and concludes by presenting the experimental results.

6.1 Experimental Design

The developed PVM with AR system was used in this experimental evaluation for the second scenario, that is, to design a behaviour-based robot that follows a wall and avoids obstacles, to prove the hypothesis stated in chapter one, particularly H1, H2, H3, H4, H7 and H8.

The first hypothesis is related to creating computational architecture that maps the technical activities (hardware and software) with the user-interface (using AR) within the context of the learning activity. It aims to provide learners and developers with a pedagogical explanation of the work being undertaken in the activity. Therefore, the PVM with AR system designed and developed based on the PVM framework corresponds with the first hypothesis (H1). Thus, to examine the validity of this hypothesis, this study investigates the learning effectiveness of the PVM framework based on the remaining hypothesis by comparing compared two educational approaches (PVM with AR application and TR of programming debugging) to solving the mobile robot learning task to explore the effects of each approach on students' learning outcomes. In doing so, the results will indicate whether the PVM framework supports learning and teaching (H1).

The experiment followed an experimental and group design using the type of approach (PVM with AR, traditional(TR)) as independent variables. The experimental evaluation employs a learning activity, which required students to design a behaviour-based robot that follows a wall and avoids obstacles. Instead of making students create robot behaviour from scratch, they were given a programming source code. The given source code was designed to make the robot misbehave at runtime, and students were asked to solve the problem based on the assigned approach (PVM with AR or TR). This approach was chosen to reduce students' workload, as the aim of the experiment was to compare which approach (PVM with AR or TR) assists students to solve the learning task by analysing robot actions. The programming source code was written using object-oriented style, which makes code easy to read, understand and modify, especial for small program (Wiedenbeck & Ramalingam, 1999). A complete list of robot classes was given and made available to use for modifying or replacing functions in the code. Both approaches (PVM with AR and TR) required students to use python programming language to edit and debug the given programming code for the mobile robot. This was based on Geany text editor that supports an integrated development environment (IDE).

An AR application was developed to visualise and analyse robot actions and states, and produce pedagogical feedback based on the learning activity. This approach followed the PVM model for structuring a learning activity and analysing computational objects in term of pedagogy. Once students debugged or executed the source code, they could point their tablet's camera at the robot to overlay robot actions and behaviours in real time. Robot actions and behaviours, and sensors' values were updated in real time on students' tablets (Figure 6-1). This gave students the ability to explore robot actions from low to high-level views that correspond to learning objects. The application provided students with four features that allowed them to explore the construction of robot actions and behaviours, as well as obtain instant feedback regarding the learning activity.

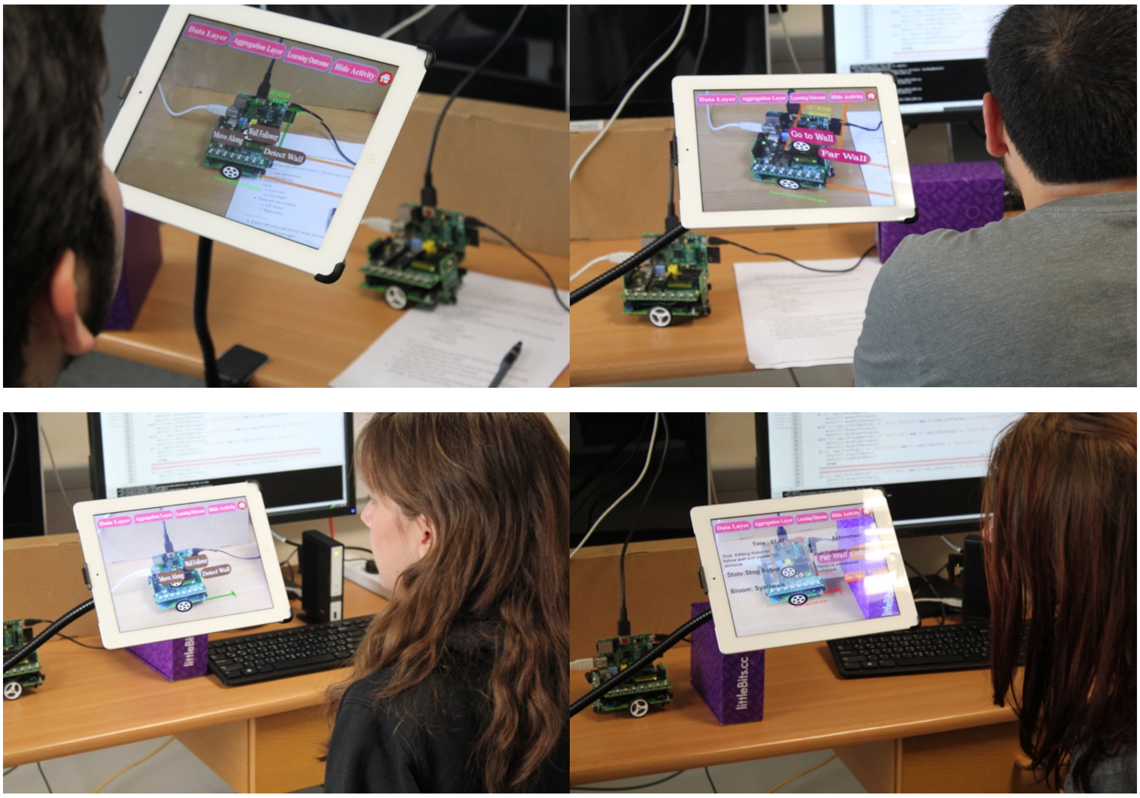


Figure 6-1 Students experimenting with AR

Figure 6-2 Students experimenting with TR

The second approach was the conventional approach for debugging and executing programming code in computer and engineering science. This approach allowed students to execute code and examine the robot without visualisation help; instead, robot outputs (actions) were represented on the tool's console (log file) in real time. A print statement was added to robot classes in order to print robot states and actions on the tool's console. This was added to assist student understanding of robot programme behaviour. It was noted by Ahmadzadeh et al. (2005) that a print statement is used in programming code as a debugging strategy to find errors. In addition, it helps in examining particular programme behaviours and revealing values of interest (Li & Flatt, 2015). Thus, during runtimes, students were able to see robot actions in abstract meanings such as ***Left Motor Move Forward, Object Near :09.*** In both approaches, the low-level data representations were the same but the appearance was different, being visualised in AR, and log file text in traditional. In both approaches, students were allowed to debug the programming code many times until the task learning objectives were achieved.

Between-subjects design was used to examine the PVM with AR and TR systems. This was used to avoid knowledge transfer which might result in making it more difficult to identify the effectiveness of the approach used. Nonetheless, within-subjects design was used only to gather participants' opinions regarding both approaches. Twenty students from the University of Essex were invited to take part in the experiment. The students were divided into two groups, PVM with AR and TR, with 10 students in each group. The procedure of the experiment was divided into three phases: before, during and after (Figure 6-3):

1. Before the experiment. Participants were invited to take part in the study. They were then provided with a link to an online questionnaire to gather demographic information and their familiarity with computing, programming and embedded system learning activity and assessment. The link was sent to participants a few days before the main experiment was run. A copy of the questionnaire is included in the appendices. Due to the availability of only one buzzbot robot, each participant was asked to choose one slot that suited him/her. On the day of the experiment, participants were informed about the experiment aim, phases and procedure. They were then given a pre-computational thinking test, in which they had to decompose a behaviour-based robot that follows a wall and avoids obstacles. The test was similar to the hierarchical decomposition in **Figure 4-14**. A copy of the pre-computational thinking test is included in the appendices B.
2. During the experiment. Participants were provided with the learning activity instruction and robot material, such as classes and functions. They then logged into Raspberry pi and used Geany IDE to start working on the assigned programming task. A tablet-holder desk stand was used to support the PVM with AR group to allow them to work hands-free. Time and debugging times were observed during the experiment by the instructor.

3. After the experiment. Participants were provided with a post-computational thinking test, followed by a post-test knowledge about the learning activity. In addition, a link to an online questionnaire was sent to participants to gather information about their experience on the approach used (PVM with AR or TR). Lastly, participants were asked to experience the other approach (swapping from PVM with AR to TR or vice versa). Another survey link was provided to gather user opinion on both approaches and to evaluate the AR system. Participants were paid £10.00 on completion of the experiment. A copy of the post-test knowledge and both questionnaires are included in the appendices B.

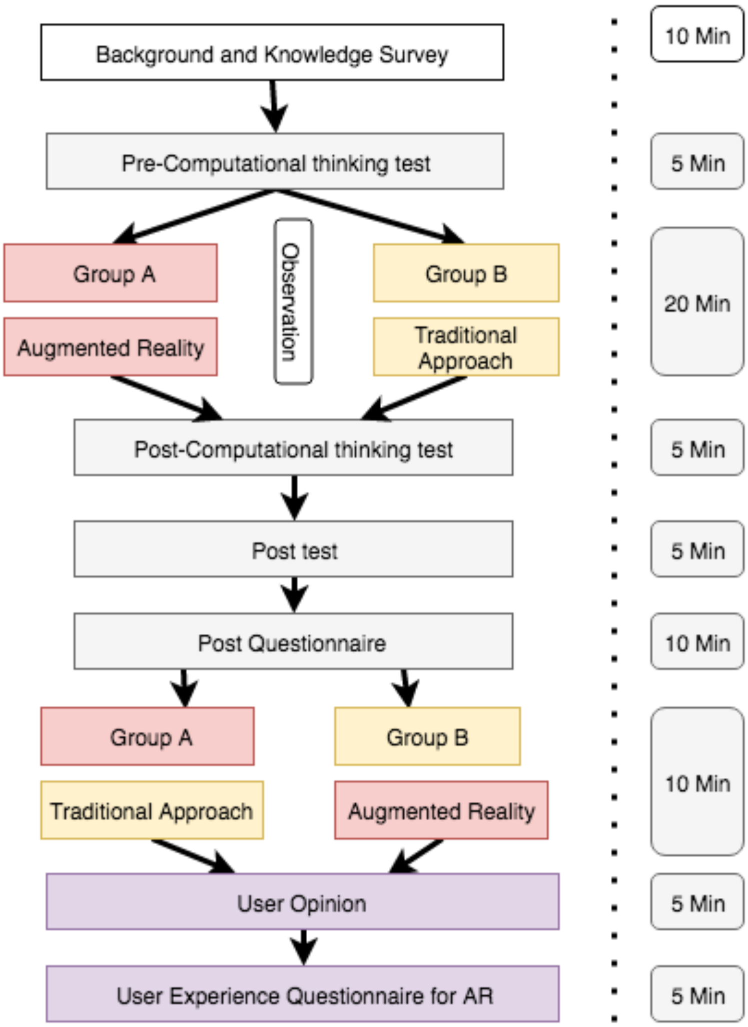


Figure 6-3 Mobile robot experiment design procedure

Qualitative and quantitative data were collected to evaluate the learning effectiveness and pedagogical value of the PVM with AR, as compared to TR. Qualitative data were analysed, identifying students' opinions when evaluating both approaches (PVM with AR, TR). Quantitative data were investigated through descriptive statistics to find correlations with the research questions and hypotheses. The data were studied, focusing on the learning effectiveness of the PVM model for analysing and exploring embedded computing learning tasks, especially on a real-time system.

The research instruments consisted of four questionnaires for participants and a short questionnaire for teachers/instructors, pre-and post-computational thinking tests, a post-learning activity test and participants' observations. The first questionnaire collected general demographic information and preliminary knowledge on evaluated topics, such as computing and programming experience, and a computing system assessment to establish the participant's background. The second questionnaire was given to participants after they finished the learning activity, and divided into two parts. The first part measured the effectiveness of the approach used for the learning activity task. This part was designed using 5-Likert scales ranging from "strongly disagree" at one end to "strongly agree" at the other, with "Neutral" in the middle. The second part measured cognitive workload when doing the learning activity. The cognitive workload was based on NASA-TLX (Task Load Index) that was used to assess the overall workload of the learning activity task (Hart & Staveland, 1988). To measure the workload experience, six rating scales was used: mental demand, physical demand, temporal demand, performance, effort and frustration. Participants rated each of these scales on a Likert scale, ranging from low to high. The third questionnaire gathered participants' opinions after experiencing both approaches (AR and TR). This questionnaire was short and designed using a combination of open and closed questions. Closed questions were utilised to gather the participants' feedback using Likert scales, whilst open questions were used to give participants the opportunity to state their views about the preferred approach. The fourth questionnaire was employed to assess the user experience (UX) of the PVM with AR system. This utilised Laugwitz et al.'s (2008) user experience questionnaire (UXQ). The UXQ was constructed using six scales covering attractiveness, efficiency, perspicuity, dependability, stimulation and novelty, with 26 items in total to support immediate feedback about participants' feelings, impressions and attitudes. Each item was presented by two terms with opposite meanings. The questionnaire for instructors/teachers asked about their impressions of the PVM with AR system. All questionnaires are included in the Appendix B. In addition, computational thinking was measured using pre-and post-test knowledge which assessed participants' ability to decompose the problem into smaller parts. One aspect of computational thinking is to use abstraction and decomposition for solving complex tasks (Wing, 2006). Therefore, the pre- and post-test were designed according to the hierarchical decomposition of a behaviour-based robot that follows a wall and avoids obstacles (see **Figure 4-14**). Participants were given hints and some parts of the hierarchy were given. Participants were provided with a short post-test knowledge to assess their learning outcomes based on the approach used (PVM with AR or TR). The test consisted of six questions, four of which required participants to indicate two things—what wrong action the robot made, and what the correct action was—whereas the other two examined participants' ability to decompose behaviour into actions. The test used an open-ended question design, where participants write short answers. Copies of all tests are provided in the Appendix. Participants were observed while performing the activity and two constructs were recorded. The first construct was the time taken to complete the task, and the second was the amount of debugging time it took to solve the task. These were observed to measure the effectiveness of each approach (PVM with AR and TR) in terms of time and number of trials. Data were anonymised and analysed using the statistical program SPSS (Statistical Package for the Social Sciences), except the user experience questionnaire (UXQ) which was analysed using Laugwitz et al., (2008) data analysis tool based on Excel.

6.2 Evaluation

6.2.1 Preliminary Data

The objective of the preliminary survey was to explore students' background knowledge and ensure that the subjects were to some extent alike, especially in programming and object-oriented paradigm experience. A complete version of the questionnaire is in Appendix B.

The number of participants was 20, and the male formed 75% compared to 20% female and 5% preferred not to say. 45% of the participants were aged between 21 and 25 years old, 25% between 26 and 30, and a further 25% between 31 and 35. Only one participant was between 36 and 40 years old. At the time of the experiment, the level of study for all participants was 100% postgraduate (master and PhD), with 85% doing computer science and 15% doing engineering. All participants (100%) owned a personal computer; furthermore, 95% of them owned a smartphone against 5% who said they did not have a smartphone. 55% of participants stated their computer expertise level as expert, against 45% who stated their level as intermediate. In addition, 50% of participants expressed their computer programming skills as intermediate, compared to expert (45%) and beginner (5%). 40% of participants said that their familiarity with object-oriented paradigms was high, in contrast to very high (25%), medium (30%) and low (5%). Participants' familiarity with object oriented programming languages were Java (75%), C++ (65%), Python (40%), C# (35%) and C (30%).

65% of participants declared that they had studied during their degree at least one of the following modules: embedded systems, robotics, artificial intelligence, digital electronics and communications engineering, compared to 35% who had not studied any of these. 90% of participants stated that they had been involved in practical activities or assignments in science or engineering labs, against 10% who had not participated in lab activities. After completing lab activities or assignments, 55% of participants stated that they felt good in relation to achieving the activity/assignment learning objective, compared to very good (10%), average (30%) and bad (5%). In relation to the assessment, 55% of participants depended on the teacher or instructor to assess their lab activity/assignment work, as compared to those who relied on an educational software tool (10%) or themselves (35%).

All participants (100%) had completed a programming assignment as part of their course. When participants discovered that the programme was not behaving as expected, 40% of them said they would look at the source code, 30% said they would add a print statement inside the programme and 30% stated they would use a debugging tool (Figure 6-4).

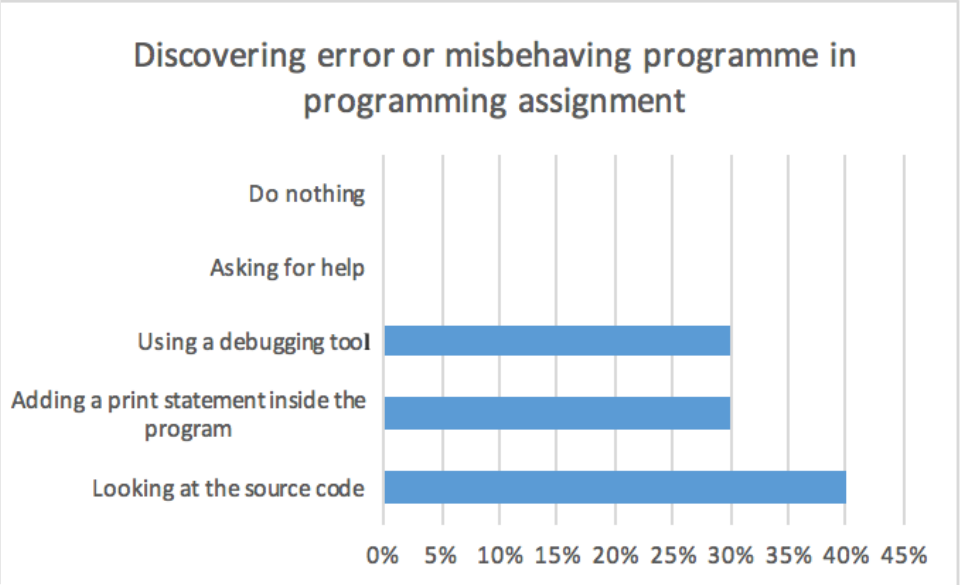


Figure 6-4 Discovering error or misbehaving programme in programming assignment

6.2.2 Exploratory Data Analysis

The first step in analysing the data from an experiment was to perform an Exploratory Data Analysis in order to check assumptions, detect outliers and properly select statistical techniques. As research variables were scale (continuous) data, means, medians, standard deviations and skewness are the best methods of summarising. Table 6-1 presents descriptive statistics of the research dependent variables. The descriptive analysis is split by the “Approach Used”. Table 6-2 presents the findings of the normality tests: Kolmogorov-Smirnov and Shapiro-Wilk. Hence, the number of students involved in this experiment was small; thus, the statistical analysis was treated with some caution.

Time: the mean Time does not differ significantly from the median Time in both approaches, suggesting the symmetry of the distribution. This can be confirmed by the very weak Skewness values. Examining the normality test results, both tests are significant, $p > .05$, suggesting a normal distribution in both data samples PVM with AR and TR. Therefore, Time is eligible for parametric tests.

Debugging Times: the mean Debugging Time is slightly higher than the median Debugging Time in both groups of students. The Skewness values suggest that the distribution is positively skewed but within the normal Skewness limits. For both groups of students, the normality tests suggest that the Debugging Time is normally distributed, $p > .05$, and hence Debugging Time is eligible for parametric tests.

Pre-Thinking: the mean score of Pre-Thinking is higher than the median score, suggesting a positively skewed distribution in the PVM with AR group. However, the Skewness value is still within the normal limits. In the TR group, the mean is relatively higher than the median, suggesting a highly positively skewed distribution. This is confirmed by the high positive Skewness value exceeding the normal limits. This may affect the normality of the distribution. The normality tests show that Pre-Thinking is normally distributed in PVM with AR group, $p > .05$. However, in TR group, the normality tests show that Pre-Thinking is deviant from normality, $P < .01$. Examining the frequency histograms, it can be noticed that there is an outlier on the upper tail of the distribution, which pulls the distribution upward to be highly positively skewed. Therefore, nonparametric tests should be used for Pre-Thinking as they are robust to outliers.

Post-Thinking: the mean score is relatively lower than the median, suggesting an asymmetric distribution of the PVM with AR sample. This is confirmed by the high negative Skewness value, exceeding the normal limits. This may suggest a non-normal distribution. In the TR sample, the mean score is largely higher than the median, suggesting a positively skewed distribution; Skewness exceeds the normal limits. The normality tests show that Post-Thinking test is deviant from normal distribution in the PVM with AR sample, p < .01. In the TR sample, the Kolmogorov-Smirnov test shows a distribution close to normal as p > .05 but the Shapiro-Wilk test shows a different result: the Post-Thinking test distribution is found to be deviant from normal, p < .05. This suggests that nonparametric tests should be used for Post-Thinking test.

Post Test: the mean score is exactly equal to the median, suggesting a symmetric distribution of data in the PVM with AR sample. However, the negative Skewness statistic may have been because of the lower outlier that can be noticed when the histogram is examined. However, the distribution is not deviant from normality. In the TR sample, the mean is slightly higher than the median and there is positive Skewness statistic but within the normal limits. Therefore, Post Test is eligible for parametric tests.

From the findings of the Exploratory Data Analysis and Normality tests, it is recommended to use parametric tests with Time, Debugging Time and Post Test. On the other hand, it is more appropriate to use nonparametric tests with Pre-Thinking and Post-Thinking.

Table 6-1 Descriptive Statistics

	PVM with AR						**TR**					
	M	**Mdn**	**SD**	**Min**	**Max**	**Sk**	**M**	**Mdn**	**SD**	**Min**	**Max**	**Sk**
Time	08:5	08:48	01:25	06:20	11:20	-.030	14:13	14:57	03:44	09:30	19:34	-.079
Debugging Times	4.30	4.00	.949	3	6	.234	8.10	8.00	1.663	6	11	.348
Pre-Thinking	32	30	16.193	10	70	1.36	21.00	10.00	19.120	10	70	2.208
Post-Thinking	73.00	80.00	10.593	50	80	-1.44	27.00	20.00	20.575	10	70	1.173
Post Test	10.50	10.50	1.650	7	12	-1.02	6.40	6.00	2.119	4	11	1.356

Skewness Standard Error = .687, Normal Range of Skewness = ±1.374

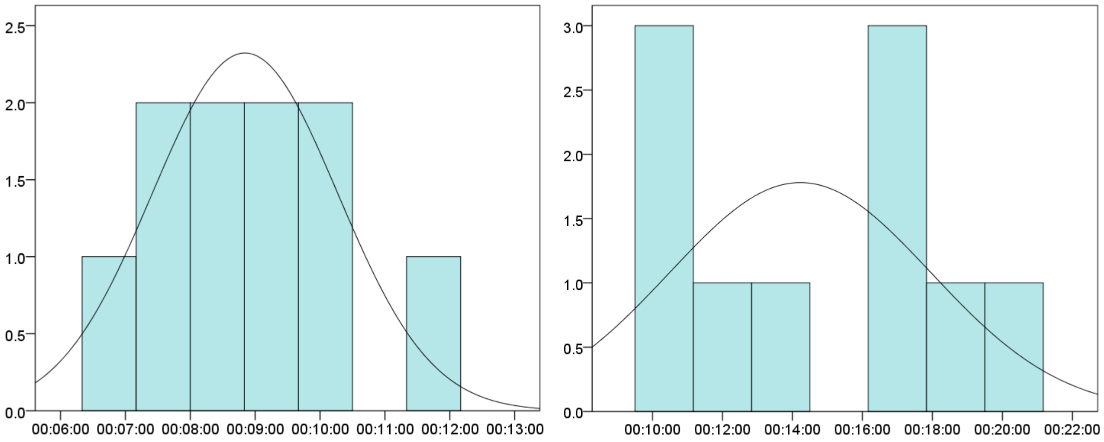
Table 6-2 Tests of Normality

	Kolmogorov-Smirnov[a]				**Shapiro-Wilk**			
	PVM with AR		**TR**		**PVM with AR**		**TR**	
	Statistic	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**	**Statistic**	**Sig.**
Time	.101	.200*	.202	.200*	.997	1.000	.910	.279
Debugging Times	.224	.168	.146	.200	.911	.287	.948	.646
Pre-Thinking	.249	.079	.317	**.005**	.869	.098	.658	**.000**
Post-Thinking	.346	**.001**	.233	.132	.730	**.002**	.837	**.041**
Post Test	.218	.194	.275	**.031**	.852	.062	.857	.070

. This is a lower bound of the true significance.
a. Lilliefors Significance Correction

Frequency Histograms

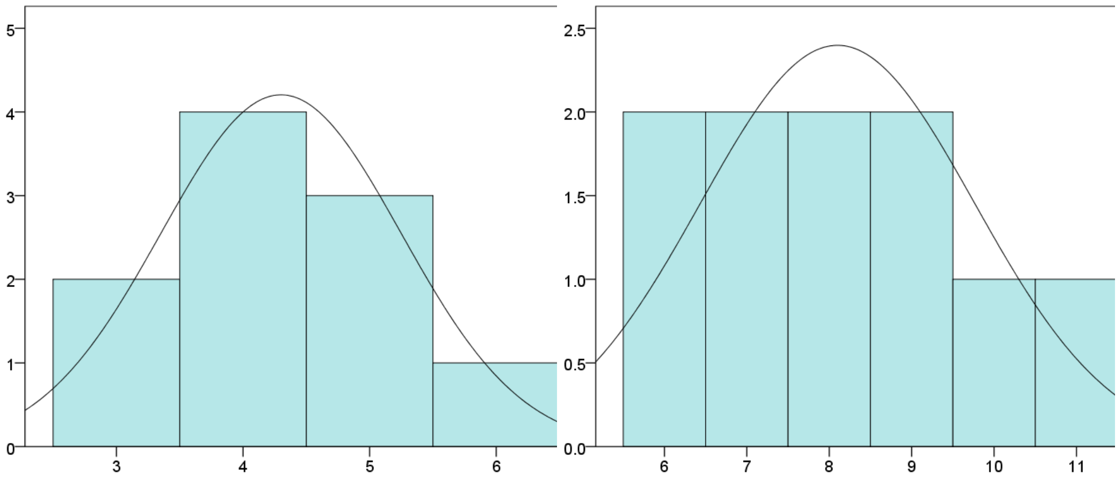
Time



PVM with AR

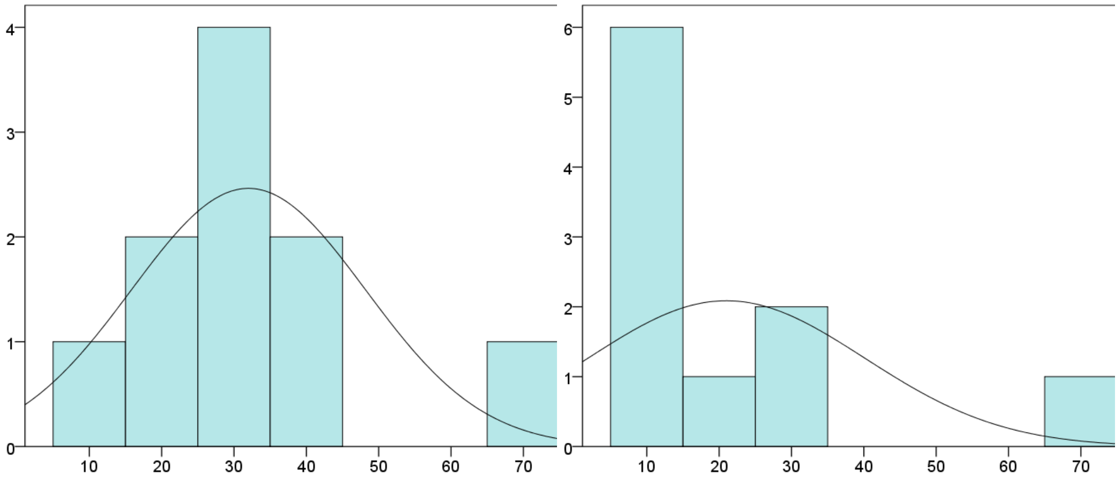
TR

Debugging Times



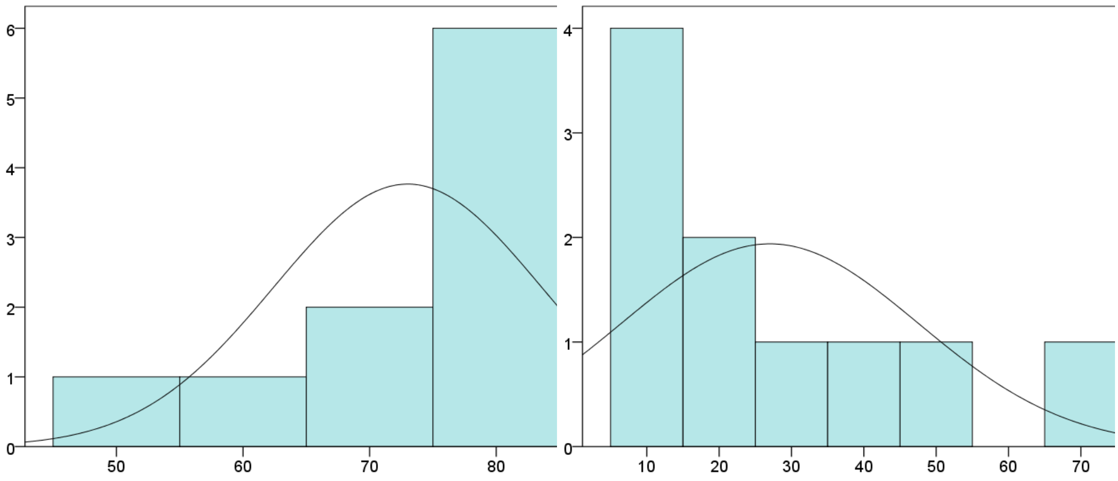
PVM with AR TR

Pre-Thinking



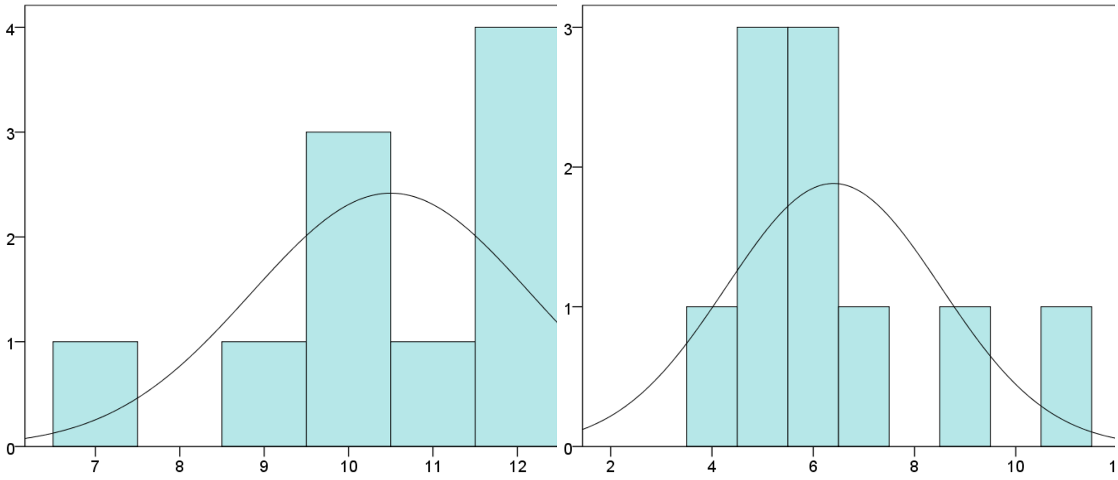
PVM with AR TR

Post-Thinking



PVM with AR TR

Post Test



PVM with AR TR

6.2.3 Learning Achievement

6.2.3.1 Knowledge test

After completing the learning objects (activities), the results of the knowledge test for the students in both groups were analysed. A two-independent sample t-test was performed to examine the significant difference in mean post-test scores among students who used PVM with AR and students who used the TR. The test revealed that students who used a PVM with AR had statistically significantly higher post-test scores (10.50 ± 1.650) than students who used the TR (6.40 ± 2.119), $t(18) = 4.828$, $p < .001$. Therefore, students who used the PVM with AR gained a better learning achievement than those who used a conventional (traditional) approach.

6.2.3.2 Computational Thinking

This construct can be tested by performing the Sign test, which is a nonparametric or distribution-free test. The Sign test was used to decide if there was a median difference between paired or matched observations (Baguley, 2012). This test is a nonparametric alternative to the parametric paired-samples t-test, when the distribution of differences between paired observations is not normal. It can also be used as an alternative to the Wilcoxon signed-rank test when the distribution of differences between paired observations is not symmetrical. Participants are usually tested at either two points of time or under two different conditions on the same continuous dependent variables of ‘pre-thinking’ and ‘post-thinking’. The assumptions of this test were examined and met:

Assumption #1: The dependent variables should be measured at either the ordinal or continuous level. ‘Pre-thinking’ and ‘post-thinking’ are measured at the continuous level.

Assumption #2: The independent variable should consist of two categorically ‘related groups’ or ‘matched pairs’. The utilised approach included two categories: AR and TR.

Assumption #3: The paired observations for each participant need to be independent, i.e. one student's scores should not influence another student's scores.

Assumption #4: The different scores (i.e. differences between the paired observations) are from a continuous distribution.Table 6-3 presents the descriptive statistics for the difference between "Pre-Thinking" and "Post-Thinking" scores, indicating asymmetric continuous distribution.

Table 6-3. Descriptive Statistics of the Difference between "Pre-Thinking" and "Post-Thinking" Scores

Approach	**M**	**5% Trimmed M**	**Mdn**	**SD**	**Min**	**Maxi**	**Sk**
PVM with AR	41.00	41.67	45.00	14.491	10.00	60.00	-1.035
TR	6.00	5.00	.00	10.750	.00	30.00	1.691

Skewness Standard Error = .687, Normal Range of Skewness = ±1.374

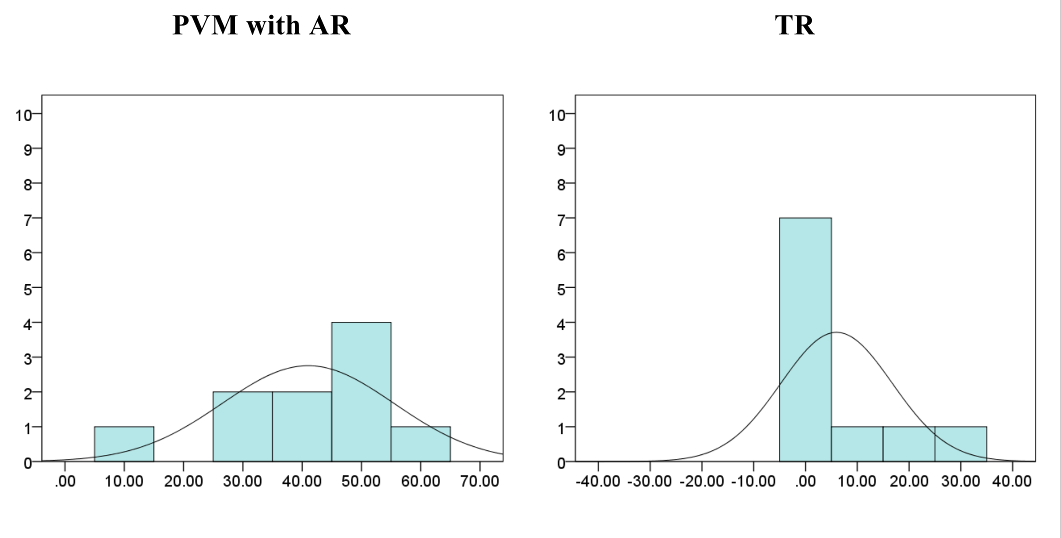


Figure 6-5. Histograms of Difference between "Pre-Thinking" and "Post-Thinking" Scores

Sign Test

Twenty students were tested to understand their computational thinking on a problem-solving task, as measured by the computational thinking scores before and after solving the problem. An exact sign test was used to compare the differences in computational thinking scores in the two trials, “Pre-Thinking” and “Post-Thinking”. In the PVM with AR student group, the “Post-Thinking” elicited a statistically significant median increase in computational thinking scores (50) compared to “Pre-Thinking”, *p* = .002. However, in the TR student group, there was no statistically significant change in the median of computational thinking scores between the two trials, “Pre-Thinking” and “Post-Thinking”.

Table 6-4 evaluates the number of positive, negative and tied paired differences to understand each student’s (relative) response to the two trials. For the PVM with AR group, the table shows that no student had decreased scores (the "Negative Differences" row), 10 students (total sample) had improved scores (the "Positive Differences" row), and no students witnessed no change (the "Ties" row) in their performance. In the TR group, the table shows that no student had decreased scores, three students (total sample) had improved scores, and seven students witnessed no change (the "Ties" row) in their performance.

Table 6-4. Sign Test Frequencies

		PVM with AR	**TR**
Post-Thinking – Pre-Thinking	Negative Differences[a]	0	0
	Positive Differences[b]	10	3
	Ties[c]	0	7
	Total	10	10

a. PostThinking < PreThinking
b. PostThinking > PreThinking
c. PostThinking = PreThinking

Table 6-5. Test Statistics[a]

	"Post-Thinking" – "Pre-Thinking"	
	PVM with AR	**TR**
Exact Sig. (2-tailed)	.002[b]	.250[b]

a. Sign Test
b. Binomial distribution used.

Based on the results of the knowledge and computational thinking tests, students who used the PVM with AR had better learning achievements than those in the traditional group. These results support hypothesis 2, which states that "*using the PVM framework within structured AR learning environments (specified in 1) will make the technical activities/information from the embedded devices visible and meaningful to the student, which will lead to improved learning outcomes for complex learning activities and improve the learners' awareness of the activities inside structured PVM with AR learning environments.*"

6.2.4 Task Performance

Throughout the experiment, the completion time for the learning activity was measured to define the amount of time required for participants to solve the task. In addition, the number of trials (debugging) was counted to indicate the number of debugging participants required to solve the task. A t-test was performed to test whether there was a statistically significant difference in mean Time and mean Debugging Times between the PVM with AR and TR groups. The test revealed a significant difference in mean Time and mean Debugging Times between students using PVM with AR and students using TR, p < .01. The test results are reported in Table 6-6. From the table, the mean Time and mean Debugging Times for the PVM with AR students were significantly lower than those for the TR students, indicating that the PVM with AR reduced the time for solving a learning activity task, and with fewer trials.

Table 6-6 Independent Samples T Test Findings for Time and Debugging Times, grouped by Approach

	Group Statistics				**Independent Samples Test**	
	PVM withAR		**TR**			
	M	**SD**	**M**	**SD**	**T**	**Sig.**
Time	08:50	01:25	14:13	03:44	-4.252	.001
Debugging Times	4.30	.949	8.10	1.663	-6.275	.000

This study found that students who used a PVM with AR had statistically significantly lower Time (08:50 ± 01:25 minutes:seconds) than students who used the TR (14:13 ± 03.44 minutes:seconds), *t*(18) = -4.252, *p* = .001. Similarly, this study found that students who used a PVM with AR had statistically significantly lower Debugging Times (4.30 ± .949 trials) than students who used the TR (8.10 ± 1.663 trials), *t*(18) = -6.275, *p* < .001. This supported hypothesis 3, which states that it *"will enable learners to acquire new knowledge more quickly and with fewer misunderstandings."*

6.2.5 Cognitive Overload

The cognitive overload questionnaire was used to assess students' workload given the methods used to achieve the learning activity. There are six subscales that calculate the overall workload. Results from two independent t-tests revealed that students who used AR with a PVM for solving programming activity had a lower mean in the overall cognitive workload than those students who used the conventional approach. This indicated a significant difference between both groups, *t* = -5.052, *p* = .001 (Table 6-7). In relation to the subscales, the test revealed that the AR group found the task did not require mental effort, as compared to the hands-on group, which was statically different (*t*= -3.398, *p* = .007). In terms of physical effort, there was no significant difference between the groups (*t*= -1.573, *p* = .143). In addition, the group which used the PVM with AR application had a lower time pressure than the control group (*t*= -2.939, *p* = .014). Control group students were not satisfied with their task performance as the approach used influenced them significantly (*t*= -4.034, *p* = .002). During the task, the PVM with AR group was less frustrated than control one (*t*= -4.572, *p* = .001). After completing the task, control group students revealed they had found it hard to accomplish, which was significantly different from the PVM with AR group (*t*= -5.903, *p* = .001). These results supported hypothesis 4, which states "*PVM with AR learning environments will provide assistances that reduce the load in learning.*"

Table 6-7 Cognitive Workload for Controlling Mobile Robot

	Group Statistics				**Independent Samples Test**	
	PVM with AR		**TR**			
	M	**SD**	**M**	**SD**	**t**	**Sig.**
Mental demand	3.40	1.430	9.90	5.877	-3.398	**.007**
Physical demand	2.90	1.969	5.70	5.272	-1.573	.143
Temporal demand	3.50	1.509	8.20	4.826	-2.939	.014
Performance	2.30	1.767	8.80	4.789	-4.034	**.002**
Effort	2.80	1.549	11.60	4.452	-5.903	**.001**
Frustration	1.50	1.080	8.50	4.720	-4.572	**.001**

Cognitive Workload	2.73	.778	8.78	3.70	-5.052	**.001**

6.2.6 Post-Questionnaire

After completing the activity, participants were asked to rate the approach used in a subjective questionnaire. In total, the frequency analysis shows that in student groups which used TR, 49% of students agreed with the items on the questionnaire while 31% disagreed. On the other hand, in student groups that used PVM with AR, the percentage of students who disagreed, 56%, was higher than the percentage of students who agreed, 43%. Table 6-8 reports the frequencies and percentages associated with each scale score for both groups of students.

In detail, all students who used PVM with AR disagreed that "discovering bugs and errors took a lot of effort", while 70% of students who used TR agreed. All students who used PVM with AR agreed that "this approach helped them to discover and correct the bugs and errors very quickly", while no student who used TR agreed. All students who used PVM with AR agreed that "by using the available tools, they were able to understand the robot behaviour", while 60% of students who used the TR disagreed. None of the students who used PVM with AR agreed that "by using the available tools, they were not able to deconstruct and examine the robot behaviour in more detail", while 50% of students who used TR agreed. All students using the PVM with AR agreed that "by using the available tools, they were able to know if they had accomplished the learning objective", while only 40% of students using the TR agreed. None of the students using the PVM with AR agreed that "when they debugged the robot, they found it difficult to keep track of the robot and to look at the robot output at the same time", while 90% of students using the TR agreed. None of the students using the PVM with AR agreed that “knowing why the robot was misbehaving while the robot was running in real time was very challenging”, while 70% of students who used the TR agreed.

Table 6-8. Frequency Distribution of Post Questionnaire Items – n (%)

	PVM with AR					**TR**				
Questionnaire	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**
1. Discovering bugs and errors took a lot of effort	7 (70)	3 (30)	0 (0)	0 (0)	0 (0)	0 (0)	2 (20)	1 (10)	7 (70)	0 (0)
2. This approach helped me to discover and correct the bugs and errors very quickly	0 (0)	0 (0)	0 (0)	2 (20)	8 (80)	0 (0)	3 (30)	7 (70)	0 (0)	0 (0)
3. By using the available tools, I was able to understand the robot behaviour	0 (0)	0 (0)	0 (0)	2 (20)	8 (80)	1 (10)	5 (50)	2 (20)	2 (20)	0 (0)
4. By using the available tools, I was not able to deconstruct and examine the robot behaviour in more detail	6 (60)	3 (30)	1 (10)	0 (0)	0 (0)	0 (0)	2 (20)	3 (30)	4 (40)	1 (10)
5. By using the available tools, I was able to know if I had accomplished the learning objective	0 (0)	0 (0)	0 (0)	2 (20)	8 (80)	0 (0)	5 (50)	1 (10)	3 (30)	1 (10)
6. When I debugged the robot, I found it difficult to keep track of the robot and to look at the robot output at the same time	6 (60)	4 (40)	0 (0)	0 (0)	0 (0)	0 (0)	1 (10)	0 (0)	6 (60)	3 (30)
7. Knowing why the robot was misbehaving while the robot was running in real time was very challenging	7 (70)	3 (30)	0 (0)	0 (0)	0 (0)	0 (0)	3 (30)	0 (0)	6 (60)	1 (10)

Scale Scores: (1) Strongly Disagree, (2) Disagree, (3) Neutral, (4) Agree, (5) Strongly Agree

6.2.6.1 Significant Differences in Post Questionnaire Items between PVM with AR Students and TR Students

Additionally, a t-test was performed to study the differences between the two groups of students with respect to their responses to the Post Questionnaire items. The test revealed that there was statistically significant differences in the mean scores of both groups' students, p <.001. The findings are reported in Table 6-9. From the table, it can be stated that:

1. Students who used PVM with AR had a significantly lower mean score of "discovering bugs and errors took lot of effort" than students who used TR, indicating that students who used PVM with AR were more likely to disagree that "discovering bugs and errors took lot of effort" than students using TR.
2. Students who used PVM with AR had a significantly higher mean score of "this approach helped me to discover and correct the bugs and errors very quickly" than students who used TR, indicating that students who used PVM with AR were more likely to agree that "this approach helped me to discover and correct the bugs and errors very quickly" than students using TR.
3. Students who used PVM with AR had a significantly higher mean score of "by using the available tools, I was able to understand the robot behaviour" than students who used TR, indicating that students who used PVM with AR are more likely to agree that "by using the available tools, I was able to understand the robot behaviour" than students using TR.
4. Students who used PVM with AR had a significantly lower mean score of "by using the available tools, I was not able to deconstruct and examine the robot behaviour in more detail" than students who used TR, indicating that students who used PVM with AR were more likely to disagree that "by using the available tools, I was not able to deconstruct and examine the robot behaviour in more detail" than students using TR.
5. Students who used PVM with AR had a significantly higher mean score of "by using the available tools, I was able to know if I had accomplished the learning objective" than students who used TR, indicating that students who used PVM

with AR were more likely to agree that "by using the available tools, I was able to know if I had accomplished the learning objective" than students using TR.

6. Students who used PVM with AR had a significantly lower mean score of "when I debugged the robot, I found it difficult to keep track of the robot and to look at the robot output at the same time" than students who used TR, indicating that students who used PVM with AR were more likely to disagree that "when I debugged the robot, I found it difficult to keep track of the robot and to look at the robot output at the same time" than students using TR.
7. Students who used PVM with AR had a significantly lower mean score of "knowing why the robot was misbehaving while the robot was running in real time was very challenging" than students who used TR, indicating that students who used PVM with AR were more likely to disagree that "knowing why the robot was misbehaving while the robot was running in real time was very challenging" than students using TR.

Table 6-9. Independent-Samples T-Test Findings for Experiment II Post Questionnaire Items

	Group Statistics				**Independent Samples Test**	
	PVM with AR		**TR**			
Questionnaire	**M**	**SD**	**M**	**SD**	**T**	**Sig.**
1. Discovering bugs and errors took a lot of effort	1.30	.483	3.50	.850	-7.117	< .001
2. This approach helped me to discover and correct the bugs and errors very quickly	4.80	.422	2.70	.483	10.357	< .001
3. By using the available tools, I was able to understand the robot behaviour	4.80	.422	2.50	.972	6.866	< .001
4. By using the available tools, I was not able to deconstruct and examine the robot behaviour in more detail	1.50	.707	3.40	.966	-5.019	< .001
5. By using the available tools, I was able to know if I had accomplished the learning objective	4.80	.422	3.00	1.155	4.630	.001

6. When I debugged the robot, I found it difficult to keep track of the robot and to look at the robot output at the same time	1.40	.516	4.10	.876	-8.399	< .001
7. Knowing why the robot was misbehaving while the robot was running in real time was very challenging	1.30	.483	3.50	1.080	-5.880	< .001

6.2.7 User Post Opinion

After experiencing both approaches, participants were asked to state their view in a short questionnaire. Frequency analysis revealed that the distribution of opinion responses by students in both groups seems similar, as 47.50% of students in both groups disagreed with the four opinions and 50% agreed (Table 6-10).

Table 6-10. Frequency Distribution of User Post Opinion

	PVM with AR					**TR**				
Opinion	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**
1. I think the use of AR has a significant advantage over traditional methods for discovering and revealing errors and bugs in embedded computing activities	0(0)	0(0)	0 (0)	1 (10)	9 (90)	0 (0)	0 (0)	1 (10)	3 (30)	6 (60)
2. I think that the use of AR is not suitable for assisting students' learning activities	5 (50)	4 (40)	1 (10)	0 (0)	0 (0)	4 (40)	5 (50)	0 (0)	0 (0)	1 (10)
3. I think the use of AR allows me to get a deeper understanding of how things work and communicate more than TR for computer science and engineering activities / assignments	0 (0)	0 (0)	0 (0)	6 (60)	4 (40)	0 (0)	0 (0)	0 (0)	6 (60)	4 (40)
4. I don't see that AR application makes any difference	7 (70)	3 (30)	0 (0)	0 (0)	0 (0)	6 (60)	4 (40)	0 (0)	0 (0)	0 (0)

Scale Scores: (1) Strongly Disagree, (2) Disagree, (3) Neutral, (4) Agree, (5) Strongly Agree

6.2.7.1 Significant Differences in the Opinions of Students in the PVM with AR and TR groups

Additionally, a t-test was performed to study differences between both groups of students with respect to their opinions (Table 6-11). The test revealed no significant differences between students of both groups, p > .05. That is, students who used PVM with AR and those who used TR thought that the use of PVM with AR had a significant advantage over traditional methods for discovering and revealing errors and bugs in embedded computing activities, and that the use of PVM with AR allowed them to get a deeper understanding of how things work and communicate more than TR for computer science and engineering activities / assignments. On the other hand, students from both groups did not think that the use of PVM with AR was not suitable for assisting students' learning activities and they disagreed that PVM with AR application made no difference.

Table 6-11. Independent-Samples T-Test Findings for User Post Opinions

	Group Statistics				**Independent Samples Test**	
	PVM with AR		**TR**			
Opinion	**M**	**SD**	**M**	**SD**	**t**	**Sig.**
1. I think the use of AR has a significant advantage over traditional methods for discovering and revealing errors and bugs in embedded computing activities	4.90	.316	4.50	.707	1.633	.127
2. I think that the use of AR is not suitable for assisting students' learning activities	1.60	.699	1.90	1.197	-.684	.503
3. I think the use of AR allows me to get a deeper understanding of how things work and communicate more than TR for computer science and engineering activities / assignments	4.40	.516	4.40	.516	.000	1.000
4. I don't see that AR application makes any	1.30	.483	1.40	.516	-.447	.660

difference		

6.2.8 Groups’ approach preferences

Moreover, participants were then asked which approach they preferred for doing the activity. 95% of participants stated their preferences for the use of a PVM with AR system for practising and carrying out similar activities, whereas only one participant preferred TR. Additionally, participants were asked to state the reason for their choice, and some of the participants’ statements were:

> *“PVM with AR makes a real difference in the way code is debugged, changing stressful and laborious work into a fun, engaging and enjoyable experience. Using this approach in assignments would definitely simplify things and have a great impact on the quality of the work submitted”* (Participant 3 in the PVM with AR group).

> “*Seeing in real time how my action translates to the robot behaviour was beneficial. I was able to quickly detect what went wrong when the code is running and didn't have to spend much time debugging*” (Participant 9 in the PVM with AR group).

> *“I am not experienced with programming robots, so I found this approach really useful as it clearly explains how everything is connected” (*Participant 4 in the PVM with AR group*).*

> *“I would rather know what the robot is doing at a higher level when I am first learning to use it. And I believe PVM with AR will allow students to progress in their understanding of the low-level instructions of the robot” (Participant*

2 in the TR group).

"As the number of rules gets larger, it becomes increasingly difficult to track the robot's responses to sensory different input. The PVM with AR method provides pictures and high-level linguistic descriptions for monitoring the robot. This would be useful for debugging the code and varying the rules" (Participant 4 in the TR group).

"I can see the use of this technology in robotics but its use in other domains will need further study. It had a lot of expressiveness which is not possible on a whiteboard" (Participant 6 in the TR group).

"I'm more used to TR and for such a simple task, it is easier to stick to already learned behaviour" (Participant 8 in the TR group).

From these comments, it is possible to see that although a few participants did not think of the use of a PVM with AR as necessary when they started to learn robot programming, also for learners who have expertise in programming though doing such a simple task in the environments that are used to use is better. The majority of participants thought the PVM with AR gave extra value to the activity, helping them to obtain a clear picture of the things that were happening in hidden worlds. This supported hypothesis 7, which states that "*using structured AR with PVM learning environments will be preferred to traditional learning environments*."

6.2.9 User Experience of the PVM with AR system

After experiencing the PVM with AR system, both PVM with AR and TR groups were given a questionnaire to rate their experience while using the system in terms of attractiveness, perspicacity, efficiency, dependability, stimulation and novelty. A t-test was conducted to indicate any difference between the groups with respect to their user experience (Figure 6-6). The test revealed that no significant difference was found in all factors. Both groups agreed on the system's attractiveness and ease of use. In addition, they rated the system as helpful in assisting them to solve the task without unnecessary effort. Moreover, both groups rated the system as reliable, supportive, excited, creative and motivated.

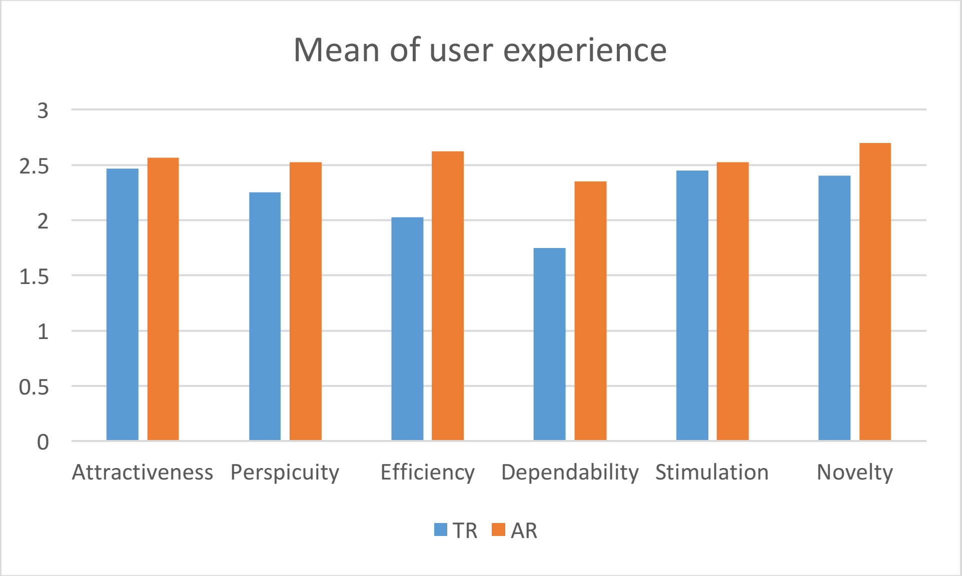


Figure 6-6 User experience mean score

6.2.10 Teachers' evaluation

Three teachers who had experience in teaching computer science were invited to experience the PVM with AR system, and were then given short open questions to state their opinion (Table 6-12). The teachers' view was gathered based on the learning activity performed (designing a behaviour-based robot) using the PVM with AR. It was important to include them in this evaluation as they have experience in computer science teaching which could offer different perspectives on the PVM with AR system.

Table 6-12 Teachers' evaluation

No.	**Question**
1	Could you give us your views on the PVM with AR system?
2	What aspects do you think that the PVM with AR could help you with when used in lab-based computer engineering activities?
3	Can you mention any other activity/class/lab that could benefit from using a PVM with AR system?
4	What would be your suggestions for improving the PVM with AR system?

In general, the views of the PVM with AR were positive, with teachers stating that it was an engaging and interactive system with attractive multimedia components. In addition, they indicated the pedagogical value of AR in enhancing the learning experience. Some of the teachers’ comments were:

> *“Students usually have difficulties in some learning tasks, such as constructing robots. Thus, I used to follow a method to ease these learning tasks by giving students instructions that have more description and images. However, I found the PVM is more attractive and timesaving than the method I used”* (Teacher 1).

> *“Sometimes, I can see students are confused when dealing with electronics (robotics as an example) due to the invisibility of the transmitted and processed data. Luckily, a PVM and AR can help in revealing this hidden data”* (Teacher 2).

> *"I think it develops the skills of problem-solving and tracing the root of problems. It could also allow real-time debugging and provides a kind of simulation which I see as very informative"* (Teacher 3).

Additionally, teachers stated that the PVM with AR could be applied in other learning activities, such as in networking modules, chemistry labs or a smart intelligent environment (e.g., transmitting data between "things"). However, they stated clearly that the PVM with AR lacks a teacher's interface where they can track students' progress, overlay students' actions and examine their strategy to accomplish the task. In addition, providing an authoring tool for constructing and customising the learning activity would be beneficial for teachers. These were suggested as improvements for the PVM with AR system, which might inform the future design of the system.

6.3 Summary

This chapter presented the experimental design used to evaluate the PVM framework based on the second learning activity scenario (designing a behaviour-based robot activity as proposed in Chapter four). The chapter explained the methodology used to validate the hypotheses (H1, H2, H3, H4, H7 and H8) introduced in Chapter one. The evaluation compared two educational applications: PVM with AR, and a traditional programming environment based on a set of research instruments used in the experiment. A group of 20 students was divided randomly between the approaches. In addition, the chapter presented the results of the learning effectiveness of the PVM model in term of computational thinking and knowledge tests, completion time, debugging times and subjective experience. Learners who used PVM with AR had better learning outcomes, faster performance and lower cognitive overload than traditional approaches. Students stated that PVM with AR helped them find errors faster and understand robot-embedded-computing processes. The chapter also presented results with an in-depth analysis regarding the usability of the system. Both groups preferred to use the PVM with AR for similar activities. They found the PVM with AR to be attractive, novel, efficient and helpful in learning embedded-computing activities. In addition, teachers stated the usefulness of using the PVM with AR when teaching.

The next chapter presents the discussion of the results of both experiments presented in this chapter and chapter five, and their wider significances for the research area.

Chapter Seven

7 Discussion

"The aim of argument, or of discussion, should not be victory, but progress."

- Joseph Joubert (1754 - 1824)

Chapters five and six presented the experimental evaluations for an augmented reality (AR) learning application that utilises a PVM explanation framework for constructing pedagogically meaningful information about computational objects. Both experiments compared the PVM with AR-based application with its equivalent traditional approach with real-time systems. Both applications were designed to provide the same information and workflow capabilities. This chapter discusses the results of the experiments and their wider consequence for the research area, as well as noting implications of the study. The chapter starts by stating the research aim and hypotheses, and then discusses the findings regarding the pedagogical effectiveness of the PVM, and system usability.

7.1 Aim and hypotheses

The aim of this research, as explained in Chapter one, was to create a computational framework that integrates learning design processes within the technology and, in particular, maps computational objects processes with learning objects in a pedagogically meaningful way to improve learning and teaching. In doing so, the thesis explored an explanation of a PVM-layered framework (presented in Chapter three) with the use of AR to show educationally-related entities. The proposed PVM framework aimed to support students and developers in understanding hidden worlds (e.g., making the invisible visible). A PVM with AR system was implemented (as presented in Chapter four) as a proof-of-concept and was used in the empirical experimentations to evaluate the effectiveness of the proposed model in improving teaching and learning, focusing on the context of embedded computing systems. Two learning activity scenarios were presented to evaluate the PVM framework, and to examine the hypotheses stated in Chapter one. The strategy to validate the hypotheses was to compare the implemented PVM with AR learning system with a traditional approach in two learning activity contexts. The reason for demonstrating the PVM in two learning contexts was to examine it against multiple learning pedagogical goals and objectives, based on a pedagogical framework (e.g., Bloom’s Taxonomy). The primary objective of any learning environment is to improve learning effectiveness; thus, PVM with AR aimed to *embeds pedagogical processes into the technology being learnt, so as to reveal the hidden computational and learning processes to the students & developers* that improves learning and teaching. Table 7-1 shows the hypothesis and states where it is evaluated, which constructs are used, and which category it belongs to.

Table 7-1 Hypothesis, evaluated experiment, constructs and type of evaluation

Hypothesis	Evaluated in	Constructs	Hypothesis categories
1	Experiment 2	Improve learning and teaching	Pedagogical and user experience
2	Experiment 1 & 2	Post knowledge test	Pedagogical
	Experiment 2	Pre-and-post computational test	Pedagogical
3	Experiment 1 & 2	Task completion time	Pedagogical
	Experiment 2	Debugging Time	Pedagogical
4	Experiment 1 & 2	Cognitive workload	Pedagogical
5	Experiment 1	Enjoyment, perceived competence and usefulness	Pedagogical
6	Experiment 1	Learner curiosity	Pedagogical
7	Experiment 2	Learning approach	user experience

		preferences	
8	Experiment 1 & 2	System evaluation	user experience

7.2 The pedagogical benefits of the PVM framework

A fundamental aspect of this research was to propose a computational model that supports learning and teaching. One of the assessment components used to assess this aspect was measuring learners' achievement by means of a knowledge test to indicate if any possible improvement had been achieved. After each experiment, the post-test knowledge was conducted, with regards to the students' learning achievement in both experiments, the results of both studies revealed that students who used the PVM with AR application had gained a higher level of learning outcome than those using the traditional approach, regardless of the learning context. The reason for this could be related to the PVM framework, as it presents abstract concepts in a pedagogically meaningful way, and gives learners enriched details of technology entities (e.g., hardware and software objects). As Tufte (1991) suggests, "to clarify, add detail", thus the PVM provided the semantic meaning of data gathered from computational objects in such a way that it blended with the pedagogical meaning, and allowed learners to learn while performing laboratory activities. This could provide clarity while studying and may result in a better learning achievement. These findings support the claim that embedding pedagogical agents into learning environments improved learners' performance and outcomes (Kim et al., 2006; Kim and Wei, 2011).Furthermore, an important aspect of a PVM is the use of AR to visualise physical details, by making the invisible visible. As noted by Magnenat et al.(2015), many learners tend to learn better in an environment that allows them to see abstract concepts by means of visualisation techniques. Additionally, AR directs learners' attention to the relevant content by highlighting important educational information about physical objects (Radu, 2014). Therefore, these findings support Hypothesis 2, which aligns with other research reporting that the use of AR technology in education enhances learning achievements (Andujar et al., 2011; Bacca et al., 2014; Ibáñez et al., 2014; Kaufmann & Schmalstieg, 2003; Magnenat et al., 2015; Shirazi & Behzadan, 2013).

Another assessment component to measure learning achievement was a pre-and-post computational thinking test. This was assessed in the second experiment to determine whether students are able to break down a complex problem into smaller parts and, in particular, to decompose a behaviour-based robot into sub-modules. Students were given the test before performing the activity and then after completing it. Students' ability to understand computational objects' behaviour and processes was found to be better in the PVM with AR approach than in the traditional approach. A possible explanation for this is that the PVM with AR system had a feature that allows learners to visualise the workflow of the data, from low- to high-level. This enabled students to interact with the behaviour by decomposing it into parts to see where the data belonged. The students do not have to imagine what is happening within abstract concepts; instead, unlike in traditional applications, they can actually see (Furió et al., 2013). Increased interactivity within the PVM with AR system is an aspect that cannot be applied in traditional methods and, therefore, may influence learners to acquire knowledge through the manipulation of the content (Dünser et al., 2012). One further explanation is that PVM provides a deeper understanding of behaviour processes, reinforcing the statement by Veletsianos and Russell (2014) that pedagogical agents enrich learners with comprehensive learning information. Another possible reason for this finding might relate to knowledge retention acquired during the use of PVM with AR application (Diegmann et al., 2015). However, further studies are required to provide evidence of the effectiveness of knowledge retention in a PVM model, especially in short- and long-term memory. This finding from the second experiment also supported Hypothesis 2 and is consistent with the study that found that AR systems improve learners' understanding of computer science concepts, such as event-handling (Magnenat et al., 2015).

An important aspect of the PVM computational model proposed was assisting learners to acquire new knowledge more quickly and with fewer misunderstandings (Hypothesis 3). This aspect was examined in both experiments. In the first experiment (an assembling and exploring activity), the completion time was calculated to see how long it took participants to complete the activity. The result showed that students who used a traditional approach assembled the mobile robot faster, and spent less time in exploring robot functionalities than those who used the PVM with AR system approach. The results showed that students using a traditional approach assembled the mobile robot faster and spent less time exploring robot functionalities than those using the PVM with AR system approach. This may relate to the PVM as it enriches learners with pedagogical information regarding the physical objects being learnt. For instance, in the assembling learning activity, each time students constructed a physical component, relevant information regarding the sensors and actuators of the components was revealed. Thus, the PVM encouraged inspection of these components during the learning process, which can enhance learner knowledge. Another explanation may be that the students found the task to be more fun and engaging with the use of AR. This finding was correlated with students' learning achievements, which revealed that learners who spent less time assembling and exploring learning objects gained a lower level of learning outcome than those who spent more time learning. This result may be explained by the fact that learners pay more attention and concentrate more when using AR technology and, therefore, to the learning content (Diegmann et al., 2015).

In contrast, in the second experiment, the same factor was conducted with debugging time and the result was that the PVM with AR system enabled learners to find misbehaving actions faster, and with less debugging time, than the traditional approach. It reduced the time taken to complete a complex task, such as solving behaviour-based robotics, and reduced the number of trails that learners needed to fix the problem. Comparison of this finding with those of other studies (Collett & MacDonald, 2010; Lalonde et al., 2006; Magnenat et al., 2015) confirms the benefits of using a visual interface for debugging, as this assists learners and developers to identify errors faster and minimises the time between runs. However, the inconsistency in findings from the two experiments may be due to the difference in learning contexts. For instance, if the learning context is at the lower level of Bloom's Taxonomy (knowledge, comprehension and application), learners might spend more time obtaining a deep understanding of the content being learnt. If, however, the learning context is at the higher level of Bloom's Taxonomy, students might prefer to solve the learning activity faster, as the knowledge is already acquired. Another possible explanation might relate to the PVM with AR side as one of affordance of AR is superimpose information on physical objects which allows learners to not shift between real environments and physical objects compared to the traditional approach that isolated both worlds. This enabled PVM with AR learners to focus on the learning context and enrich their learning experience, whether at the lower or higher levels of Bloom's Taxonomy. Further explanation may support the idea that the longer time of the first experiment was related to the higher levels of fun and engagement experienced by the students that used AR.

An important angle for the PVM model proposed was the requirement of providing learners with assistance during learning in order to reduce the task workload, as proposed in Hypothesis 4. The reason for this measurement was to ensure that the PVM model, as a learning environment, does not add more complexity for learners. The cognitive workload was evaluated, in both experiments, by obtaining learners' views on the learning activity workload in terms of mental demand, physical demand, temporal demand, performance, effort and frustration. Based on feedback from learners, the analysis showed that cognitive overload was lower in the PVM with AR than with the traditional approach in both experiments. These results agree with the findings of other studies, in which users in the AR condition had a lower task workload (Medenica et al., 2011; Tang et al., 2002, 2003). In a more in-depth analysis of cognitive workload subscales, learners considered the assembling and exploring learning task made high physical demands and felt more frustrated when using the traditional approach than when using a PVM with AR. This could be related to the fact that the traditional group shifted between paper-based instruction and the technology being learnt (e.g., BuzzBot), whereas learners who used a PVM with AR used tablets as a single medium to accomplish the task. Another explanation for this may relate to the quantity of materials that learners need to work on (Cheng & Tsai, 2012). Additionally, learners in both groups found the task simple and not hard to accomplish, and they felt relaxed during the task, under no time pressure and satisfied with their performance. In contrast, the result from the editing programming learning activity (the second experiment) was that the use of a traditional programming environment to debug the robot and find misbehaving actions was complex and required more effort than the use of a PVM with AR system. Moreover, the traditional programming method increased learners' frustration during the task, and traditional learners rated their performance lower than those in the PVM with AR group. It is worth noting that the primarily demographic results revealed that all learners, in both groups, have programming skills; thus, these results did not relate to learners' programming skills ability. It seems possible that these results are due to learners who use a traditional programming environment seeing both worlds (robot and environment) and trying to figure out the cause of the problem, whereas those who were using a PVM with AR were immersed in both worlds. This could be additional evidence in support of Hypothesis 3, that states that AR with PVM learners achieve new knowledge faster and with fewer misunderstandings. Both groups agreed that an editing embedded computing programming task did not require physical activity. It may be that these learners are used to practical tasks that involve programming in their studies.

Hypothesis 5 pointed to the importance of assessing learners' enjoyment, competence and usefulness while performing a learning activity. The effectiveness of the available tools for assembling tasks was also measured. These constructs were evaluated in assembling and exploring an embedded computing activity experiment. The data analysed from learners' subjective experiences showed that learners using the PVM with AR tool found the learning activity more enjoyable and useful than those in the traditional group. This may support the idea that AR increases learners' motivation and enjoyment levels compared to traditional approaches (Radu, 2014). Learners' ability to do the task did not differ between groups, as they could both accomplish the learning objectives of the task using the available tools. In both groups, the results showed no difference between the tools for aiding learners in assembling mobile robot components (e.g., Buzzbot). Thus, the present results support Hypothesis 5 in two factors, namely, increased learning activity enjoyment and usefulness for learners; and do not support competence and assembling teaching approach effectiveness of using PVM with AR. It is possible that these results are due to users feeling that AR technology is more fun than the non-AR application, and being more keen to repeat it, even if the AR system is not easy to use (Juan et al., 2010; Radu, 2014). The reason that learners' competence and effectiveness may relate to the simplicity of the learning activity is its link to the low level of Bloom's Taxonomy. These factors have not been investigated in the second experiment, which is situated at a high level of Bloom's Taxonomy; therefore, it is worth considering these factors as applied to a complex learning context.

Another aspect that was addressed in the evaluation of the assembling and exploring experiment relates to learners' curiosity when using a PVM framework with AR, as compared to the traditional approach. This construct was analysed by observing learners when interacting with and manipulating robot functionalities. Both manipulated objects and the number of actions performed by learners were recorded. The study found that learners who use a PVM with AR were more motivated in exploring robot functionalities than those in the traditional group (which supports Hypothesis 6). An in-depth examination of this shows that the number of occurrences of robot objects being manipulated by learners was higher in the PVM with AR approach than in the traditional one. Similarly, the number of actions which students performed on robots was higher in the PVM with AR group. When exploring robot functionalities, the traditional group tended to inspect objects that had a tangible physical appearance (e.g., motors, LED, buttons), whereas objects that had hidden outcomes were less interacted with and manipulated (e.g., line sensors, light sensors, IR sensors). On the other hand, the PVM with AR group showed great curiosity in inspecting all robot functionalities by making use of AR visualisation (which supports Hypothesis 6). There are several possible explanations for this result. The first might be that the PVM provided learners with pedagogical explanations and interactivity regarding the interacted objects, and they got instant feedback about their progress. Another possible explanation is that AR technology enables learners to experience phenomena in a way that is impossible using the traditional approach, and that these phenomena are dynamic and interactive, allowing learners to have control over the learning content (Chen, 2006). Lastly, this result may be explained by the fact that student-centred learning is increased by the use of AR technology, which improves students' ability to explore knowledge and solve problems . However, an interesting finding is that the learning achievements of those learners who manipulated and interacted more with robot objects and functionalities increased. This finding was observed in both approaches (PVM with AR and traditional). The fact that the PVM with AR approach increased learners' curiosity for learning and discovering is in agreement with our earlier finding, which showed that the use of a PVM with AR improves learners' learning outcomes.

7.3 Approach preferences and system evaluation

Hypothesis 7 specified that learners would prefer the use of PVM with AR learning environments over traditional environments to perform embedded computing learning activities. The strategy to evaluate learners' preference was to ask participants to experience both learning environments. Based on learner feedback, the analysis showed that a large majority of participants preferred using a PVM with AR to do embedded computing laboratory practical work. Learners thought a PVM with AR had a significant advantage over traditional methods for discovering and revealing errors and bugs in embedded computing activities, and that it enabled them to acquire a deeper understanding of how things work and communicate than the traditional approach. Additionally, they considered that a PVM with AR was a suitable tool to aid them during the activity and guide them in achieving the learning objectives. This result is consistent with the comments provided by participants who stated that a PVM with AR keeps the focus on the learning context, making it easy to observe and visualise phenomena. Another comment stated the usefulness of a PVM with AR in regard to the deconstruction and construction mechanism in explaining the components of physical objects (e.g., software and hardware). Holding attention, usefulness, being exciting, offering greater learning speed and saving time are among other reported comments that might provide evidence for participants' choice of a PVM with AR. One interesting comment made by one of the participants who preferred the traditional approach to using a PVM with AR was that he/she would do a simple learning activity in the environment already learnt. A possible explanation for this claim may be related to the computing and programming skills the participant has. Radu (2014) indicates that AR technology may not be an effective learning approach for some students. It could also be related to the design of the learning activity, as this was not proposed for solving complicated embedded computing problems. However, it can be concluded that this finding supports Hypothesis 7, that states that a PVM with AR would be preferred to traditional methods, which is also supported by the research outcome of Sayed et al. (2011).

Finally, usability is a fundamental aspect that needs to be considered when designing a new learning environment. Therefore, a PVM with AR application was assessed to gather information regarding learners' experience of the system. The results of the user experience evaluation showed that the overall impression of the PVM with AR was positive and that participants liked the system. Similarly, participants positively rated the PVM system as clear, and easy to use and become familiar with. Moreover, learners indicated the PVM with AR was an effective tool in solving the learning activity without extra effort, whereas losing tracking issue was minor. Furthermore, the ability to interact with the content and control the level of information representation meant the learners regarded the system as highly dependable. The use of a PVM with AR was viewed as exciting and motivating by the learners. Likewise, the way that the PVM presented information regarding the physical objects being studied was considered to be innovative and creative. These results supported Hypothesis 8 and give an indication as to why participants prefer using a PVM with AR to the traditional approach (Hypothesis 7).

7.4 Summary

This chapter discussed the results of two experiments conducted from two angles, namely, the pedagogical effectiveness and usability of a PVM. It examined the results in relation to existing research and provided further explanations. Overall, the PVM with AR improved learning and teaching, as compared to traditional environments. Learners who used the PVM with AR had higher learning achievements in both experiments. There was a positive impact on learners' achievement in terms of decomposing and understanding embedded-computing behaviours. Learners also more quickly pinpointed misbehaviours in embedded-computing systems. The PVM with AR also minimised the task workload, as it kept learners focused on the learning context by combining the embedded-computing world with the learning environment. Furthermore, users' impressions of the PVM with AR were positive.

The next chapter presents a summary and final thoughts on this thesis, describing challenges for future work.

Chapter Eight

8 Concluding Remarks

"The clarification of visual forms and their organization in integrated patterns as well as the attribution of such forms to suitable objects is one of the most effective training grounds of the young mind."

- Rudolf Arnheim (1904, 2007)

The motivation of this thesis was to create a platform that would be able to extract pedagogical meaningful information from the technology being learnt to improve learning and teaching abstract concepts within structured augmented reality environments. Examples of such activities include learning embedded-computing activities that are hidden from a learner's view, which makes the ideation of its processes and communications harder. To achieve this challenge, this thesis presented an in-depth background and literature review concerning mixed reality and augmented reality, their use to enhance learning and teaching hidden things, and the capability to incorporate them with smart technologies and educational paradigms (Chapter Two). Informed by the literature findings, the thesis proposed a novel computational architecture model called the pedagogical virtual machine (PVM) (Chapter Three) that offered solutions for integrating learning processes with computational objects in a single learning environment. The PVM was implemented as a proof-of-concept system (Chapter Four), which employed two embedded computing activities to evaluate the learning effectiveness of the PVM model, the evaluations reported in chapters five and six, and discussed their outcome in Chapter

Seven. This final chapter (Chapter Eight) presents the conclusions from this work and suggests further research.

8.1 Summary of Achievements

This thesis presented two empirical experiments to validate the eight hypothesis, as stated in Chapter one, based on a proof-of-concept prototype (PVM with AR) conducted with undergraduate and postgraduate students from the School of Computer Science and Electronic Engineering (CSEE) at the University of Essex. The implemented proof-of-concept prototype employed the principle of the computational architecture (PVM) proposed in Chapter Three, which provides synchronised real-time layered explanation components that process the computing activities obtained from technology in terms of pedagogy and construct a meaningful view of the activities using augmented reality making invisible things visible. Two embedded computing educational learning activities that utilise modularised educational mobile robot components (e.g., Buzzboards) were illustrated following the design specifications of learning and computational objects presented in Chapter Three. The context of learning activities was assessed by classifying the activity learning objectives into levels of complexity and specificity on Bloom's cognitive hierarchical model. For example, assembling and exploring the learning activity was based on the lower level of Bloom's Taxonomy, knowledge, comprehension, and application (the first learning scenario as presented in Chapter Four), where learners learn, construct, and discover mobile robots (e.g., Buzzboard components). On the other hand, the learning objectives of designing behaviour-based robot learning activities focused on analysis and synthesis on Bloom's Taxonomy, which is considered a high-level learning objectives (the second learning scenario as proposed in Chapter Four), where learners are required to have prior knowledge in programming to work on the activity. Both activities measured the pedagogical effectiveness and the user experience of the PVM with AR learning environments compared to the traditional educational approach.

Results of the pedagogical effectiveness of both learning activity experiments (presented in chapters Five and Six) indicated that the learning achievement of the students using the PVM with AR significantly outperformed the other group who used traditional environments. Moreover, students' computational thinking ability for formulating the problem was regarded as high when using the PVM with AR, supporting Hypothesis 2. In terms of learning performance, results from the first experiment showed that PVM with AR students spent more time discovering the functionality of the mobile robot while performing the activity compared to the traditional approach, which also found an increase in students' curiosity, supporting Hypothesis 6. In contrast, the PVM with AR students performed faster and with fewer trails when designing behaviour-based robots compared to the traditional group. Thus, students' acquisition of new knowledge faster with few misunderstandings (Hypothesis 3) depending on the type of the learning context indicated students would spend more time in the lower level of Bloom's Taxonomy and less time in the high level when using a PVM with AR system. In addition, the longer time of the first task may be related to the fact that students found AR to be more fun and engaging. Students reports of cognitive overload of the learning activity tasks was reduced when using the PVM with AR system compared to the traditional approach. Moreover, they indicated the PVM with AR would increase enjoyment and usefulness while performing hands-on activities. However, perceptions of students' competence with respect to learning activities did not differ from both learning approaches, as both facilitate their goal attainment, particularly in the first scenario.

In general, the PVM with AR improved learning and teaching over traditional learning environments, supporting Hypothesis 1.

Results of the user experience after using both approaches revealed that students would use the PVM with AR system over traditional learning environments when conducting lab-based computer-engineering activities. Moreover, students indicated the value of having pedagogical virtual information overlaid on top of the technology being learnt while doing the activity as a supportive for learning, particularly in providing an explanation about deeply hidden technology (e.g., software and hardware). They stated that their learning experience and understanding of the activity increased, recognising how things work and understanding their structure from low-level to high-level explanation of learning and computational processes. However, few participants stated the use of PVM with AR would be beneficial in complicated learning activities, but for such a simple learning task they would prefer to use the environments they are already familiar with. In terms of PVM with AR system evaluation, students stated the system is attractive and it is easy to become familiar with its functionalities. Furthermore, they indicated the PVM with AR is supportive while conducting hands-on activities that involved working on computational objects, and it has helped them solve tasks without unnecessary effort, as the PVM system combines the learning within the technology in a single learning environment. Last, students evaluated the PVM with AR system as exciting and motivating to use, as well as viewed it as innovative and creative for doing lab activities.

8.2 Contributions

This thesis proposed in Chapter Three the PVM model, a platform-independent interface for students to access information that is pertinent to learning. It interprets and communicates the hidden (deep) computational processes for the purpose of helping students or developers visualise functions in a computer. Its acts as an interpreter for managing educational learning-related functions on the computer. This model addressed the limitation of isolating learning and technology activities and allowing the coexistence of both activities within single learning environments. The model provided a layered explanation framework of learning and computing activities to improve learning and teaching. This model was complemented with augmented reality technology to provide students and developers with a meaningful view of the activities, making the invisible visible. This thesis also proposed an instructional design strategy for constructing learning activity tasks that incorporate learning objects with physical objects; this was based on learning objects paradigms (Chapter Three).

To implement the proposed framework, this research presented a proof-of-concept prototype in Chapter Four, the PVM with AR learning environments, which scaled up activities into a level of complexity and specificity based on a pedagogical framework. The activity involved the use of software and hardware objects, thereby enabling the integration of both learning and computing activities in a single learning environment.

Finally, this thesis presented an evaluation the PVM with AR prototypes through two user studies comparing them to equivalent traditional approaches, including an analysis of the pedagogical effectiveness of the proposed model and the user experience. The first study involved constructing and exploring embedded computing learning tasks targeting the lower level of the pedagogical framework (Chapter five), whereas the second study addressed the challenge of controlling real-time systems focusing on the high level of the pedagogical framework (Chapter six).

Additionally, a secondary contribution was included as follows:

- Designing AR user interfaces driven by the structured information architecture to support complex learning tasks.

8.3 Limitations

One aspect that was not addressed in this study, due to time limitations, is related to comparing the PVM with AR application to other forms of learning technology that utilise the principle of the PVM framework (e.g., web-based, virtual environments). It is, therefore, worth considering this aspect when employing a PVM framework in other studies. This could help in identifying whether the information provided by the PVM influences learners to gain knowledge regardless of the technology being used. Hence, Ibáñez et al. (2014) compared AR learning to web-based applications and found that the AR group performed better than the web-based group.

The study took into consideration learners' prior knowledge to ensure the validity of the sample, whereas other learner characteristics (e.g., gender, age difference, level of computer expertise, level of study) were not examined. Squire and Jan (2007) found older students are different to younger students, in terms of making arguments and integrating pieces of evidence in AR-related science-learning. In addition, O′Shea et al. (2011) found that a male group had better conversations during the process of the activity than a female group, and related this to male gaming experience. Additionally, Cheng and Tsai (2012) suggested considering presence in

AR environments as an important learner characteristic which indicates to what degree learners feel immersed within the AR learning environments. Another aspect not addressed relates to AR displays. A tablet was used as a visual interface for the AR learning environment, whereas a head-mounted display and glasses were not employed. In addition, a technical evaluation for measuring network and system latency between the AR display and physical objects was not considered; it would be beneficial if this were applied to indicate the accuracy of synchronisation when overlaying information from a real-time system. Lastly, another important aspect which was not examined relates to content creation for a PVM with AR. The PVM model proposed provides the workflow for constructing learning activity embedded within technology but, as mentioned earlier, creating an authoring tool for the teacher is beyond the scope of this thesis. This evaluation, therefore, excluded the evaluation of teachers of the usability of a PVM authoring tool.

8.4 Future work

This thesis uncovered a number of additional research challenges that provide a general outline for future research. These can be listed as:

- The proposed PVM model was focused on a single learner in formal education environments, which has not gone through all 4D-learning activities classification proposed in Chapter three, specifically collaboration activities. Thus, employing collaborative activities within the PVM model would pose a different challenge that could be worth investigating by researchers. For example, the second learning activity scenario proposed in Chapter four could be modified to allow a team of students to collaborate on writing software to build two desktop robots (e.g. based on BuzzBoards technology) to play a simple game of football by pushing a large ball into a goal. This is a common assignment is computer science education. The way these assignments work is that each student assumes responsibility for one function of the robot control system (e.g., obstacle avoidance, ball follow, ball pushing, goal finding, and inter-robot communication). When combined into a robot, none of these functions are visible and the students would each use the augmented reality displays (e.g., tablets, smartphones, glasses, HMD) to point at the robot and see the invisible aspects of the robot (sensor data, process interaction) from various perspectives while discussing the problem. Then, by using constructionist pedagogy, they would combine as a team to learn how to develop the robots together. Synchronisation interaction in real time between viewers and the physical world and between the collaborating team remains a challenge when employing a PVM with AR model.

- The PVM with AR prototype (presented in Chapter four) did not enable teachers to construct and customise the learning activity that involves the integration of learning processes within technologies because of the time limit of this PhD. This limitation was also reported by teachers when they tried the PVM with AR system. Thus, providing an interface that allows teachers to configure single or collaborative learning activities is still possible research (e.g., defining learning and computational objects, activities relationships, assessments, linking AR virtual elements to learning and computational processes, feedback).
- In this thesis, the PVM framework was deployed within one dimension (AR) in the virtuality continuum (Milgram and Kishino, 1994). Thus, it is important to consider other dimensions in the scales, by employing PVM

within these dimensions in an educational setting, researchers can examine the pedagogical effectiveness and the feasibility of the framework in other domains to see if the model adds extra value.

- The evaluation of PVM was demonstrated within a short period. Thus, considering undertaking a large scale of learners and longitudinal evaluation could give an important indication of the suitability of the model for improving learning and teaching (Bacca et al., 2014).
- The PVM was examined in a specific learning context (embedded-computing learning activities), and modularised educational components called 'Buzzboards' were utilised to prove the vision of the PVM. However, generalising the PVM model to work with diverse physical components and other learning contexts remains open. Thus, further challenges could be related to creating a communication layer that works to reveal communication processes within physical components.
- In the AR user interface, providing a feature that allows learning to control the time-series, and describes the events and process during learning activities would be beneficial (e.g. timestamps). As the system provides real-time analysis of technology processes related to learning, it is worth considering capturing the whole computing process and store it in students' profile as a video or story image sequences to enable them to investigate it in their own pace.

Finally, I would like to think that the ideas of the PVM presented in this thesis would be the first step toward making an independent platform that anyone can use for learning within technology as stated in the first chapter, "*As any computer can have a Java virtual machine; also, any computer-based system that is used for education can have a pedagogical virtual* *machine*". Thus, the vision is wider, but this thesis addressed the main principle of the PVM, which could be then extended further by inspiring researchers.

References

2016 Horizon Report [WWW Document], n.d. URL https://library.educause.edu/resources/2016/2/2016-horizon-report (accessed 7.11.16).

AAgren, O., 1999. Teaching Computer Concepts Using Virtual Machines. SIGCSE Bull 31, 84–85. doi:10.1145/571535.571578

Abenza, P.P.G., Olivo, A.G., Latorre, B.L., 2008. VisualJVM: A Visual Tool for Teaching Java Technology. IEEE Trans. Educ. 51, 86–92. doi:10.1109/TE.2007.906601

Ackermann, E., 2001. Piaget's Constructivism, Papert's Constructionism: What's the difference?, in: CONSTRUCTIVISM: USES AND PERSPECTIVES IN EDUCATION, VOLUMES 1 & 2). CONFERENCE PROCEEDINGS, GENEVA: RESEARCH CENTER IN EDUCATION/CAHIER 8/SEPTEMBER 01. PP. Citeseer.

Agha, G., 1990. Concurrent Object-oriented Programming. Commun ACM 33, 125–141. doi:10.1145/83880.84528

Ahmadzadeh, M., Elliman, D., Higgins, C., 2005. An Analysis of Patterns of Debugging Among Novice Computer Science Students, in: Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE '05. ACM, New York, NY, USA, pp. 84–88. doi:10.1145/1067445.1067472

Alharbi, A., Henskens, F., Hannaford, M., 2012. Student-Centered Learning Objects to Support the Self-Regulated Learning of Computer Science. Creat. Educ. 3, 773–783. doi:10.4236/ce.2012.326116

Alonso Amo, F., López Gómez, G., Manrique Gamo, D., 2008. Learning Objects, Learning Objectives and Learning Design. Innov. Educ. Teach. Int. 45, 389–400.

Alrashidi, M., Callaghan, V., Gardner, M., Elliott, J., 2013. ViewPoint: An Augmented Reality Tool For Viewing and Understanding Deep Technology, in: Ambient Intelligence and Smart Environments. pp. 252–261.

Amstutz, P., Fagg, A.H., 2002. Real time visualization of robot state with mobile virtual reality, in: IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02. Presented at the IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02, pp. 241–247 vol.1. doi:10.1109/ROBOT.2002.1013368

Anastassova, M., Panëels, S., Souvestre, F., 2016. Methods for User Involvement in the Design of Augmented Reality Systems for Engineering Education, in: Szewczyk, R., Kaštelan, I., Temerinac, M., Barak, M., Sruk, V. (Eds.), Embedded Engineering Education, Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 81–91. doi:10.1007/978-3-319-27540-6_5

Anastassova, M., Souvestre, F., González, E.A., Gutiérrez, A.S., Benito, J.R.L., Barak, M., 2014. Learner-centered evaluation of an augmented reality system for embedded engineering education, in: Federated Conference on Computer Science and Information Systems (FedCSIS), Warsaw, Poland.

Andujar, J.M., Mejias, A., Marquez, M.A., 2011. Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory. IEEE Trans. Educ. 54, 492–500. doi:10.1109/TE.2010.2085047

Arkin, R.C., 1989. Motor Schema — Based Mobile Robot Navigation. Int. J. Robot. Res. 8, 92–112. doi:10.1177/027836498900800406

Arvanitis, T.N., Petrou, A., Knight, J.F., Savas, S., Sotiriou, S., Gargalakos, M., Gialouri, E., 2009. Human factors and qualitative pedagogical evaluation of a mobile augmented reality system for science education used by learners with physical disabilities. Pers. Ubiquitous Comput 13, 243–250. doi:10.1007/s00779-007-0187-7

Azuma, R., 1997. A Survey of Augmented Reality. Teleoperators Virtual Environ. 6, 355–385.

Azzari, P., Stefano, L.D., 2009. Vision-Based Markerless Gaming Interface, in: Foggia, P., Sansone, C., Vento, M. (Eds.), Image Analysis and Processing – ICIAP 2009, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 288–296. doi:10.1007/978-3-642-04146-4_32

Bacca, J., Baldiris, S., Fabregat, R., Graf, S., Kinshuk, 2014. Augmented Reality Trends in Education: A Systematic Review of Research and Applications. Educ. Technol. Soc. 17, 133–149.

Bach, C., Scapin, D.L., 2004. Obstacles and perspectives for evaluating mixed reality systems usability, in: Acte Du Workshop MIXER, IUI-CADUI. Citeseer.

Balc, S., 2004. A science lesson designed according to 5E model with the help of instructional technology, in: Proceedings of the International Educational Technologies Conference. pp. 89–93.

Ball, M., Callaghan, V., Gardner, M., Trossen, D., 2010. Achieving human-agent teamwork in eHealth based pervasive intelligent environments, in: 2010 4th International Conference on Pervasive Computing Technologies for Healthcare. Presented at the 2010 4th International Conference on Pervasive Computing Technologies for Healthcare, pp. 1–8. doi:10.4108/ICST.PERVASIVEHEALTH2010.8889

Banister, S., 2010. Integrating the iPod Touch in K–12 Education: Visions and Vices. Comput. Sch. 27, 121–131. doi:10.1080/07380561003801590

Barakonyi, I., Psik, T., Schmalstieg, D., 2004. Agents that talk and hit back: animated agents in augmented reality, in: Third IEEE and ACM International Symposium on Mixed and Augmented Reality. Presented at the Third IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 141–150. doi:10.1109/ISMAR.2004.11

Barker, B.S., Ansorge, J., 2007. Robotics as Means to Increase Achievement Scores in an Informal Learning Environment. J. Res. Technol. Educ. 39, 229–243. doi:10.1080/15391523.2007.10782481

Bartneck, C., Hostlaan, P., 2003. Interacting with an Embodied Emotional Character, in: In Proceedings of the 2003 International Conference on Designing Pleasurable Products and Interfaces. ACM, pp. 55–60.

Beer, R.D., Chiel, H.J., Drushel, R.F., 1999. Using Autonomous Robotics to Teach Science and Engineering. Commun ACM 42, 85–92. doi:10.1145/303849.303866

Benito, J.R.L., González, E.A., Anastassova, M., Souvestre, F., 2014. Engaging computer engineering students with an augmented reality software for laboratory exercises, in: 2014 IEEE Frontiers in Education Conference (FIE)

Proceedings. Presented at the 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, pp. 1–4. doi:10.1109/FIE.2014.7044094

Benitti, F.B.V., 2012. Exploring the educational potential of robotics in schools: A systematic review. Comput. Educ. 58, 978–988. doi:10.1016/j.compedu.2011.10.006

Bentrad, S., Meslati, D., others, 2011. Visual Programming and Program Visualization–Towards an Ideal Visual Software Engineering System. IJIT-ACEEE Int. J. Inf. Technol. 1, 56–62.

Billinghurst, M., Dünser, A., 2012. Augmented Reality in the Classroom. Computer 45, 56–63.

Billinghurst, M., Hakkarainen, M., Woodward, C., 2008. Augmented Assembly Using a Mobile Phone, in: Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia, MUM '08. ACM, New York, NY, USA, pp. 84–87. doi:10.1145/1543137.1543153

Bloom, B.S., 1956. Taxonomy of educational objectives: the classification of educational goals. D. McKay.

Blum, T., Kleeberger, V., Bichlmeier, C., Navab, N., 2012. mirracle: An augmented reality magic mirror system for anatomy education, in: 2012 IEEE Virtual Reality Workshops (VRW). Presented at the 2012 IEEE Virtual Reality Workshops (VRW), pp. 115–116. doi:10.1109/VR.2012.6180909

Booch, G., 1994. Object-oriented Analysis and Design with Applications (2Nd Ed.). Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA.

Borrero, A.M., Márquez, J.M.A., 2011. A Pilot Study of the Effectiveness of Augmented Reality to Enhance the Use of Remote Labs in Electrical Engineering Education. J. Sci. Educ. Technol. 21, 540–557. doi:10.1007/s10956-011-9345-9

Bostanci, E., Kanwal, N., Ehsan, S., Clark, A.F., 2013. User Tracking Methods for Augmented Reality. Int. J. Comput. Theory Eng. 93–98. doi:10.7763/IJCTE.2013.V5.654

Bovet, G., Hennebert, J., 2013. Energy-Efficient Optimization Layer for Event-Based Communications on Wi-Fi Things. Procedia Comput. Sci., The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013) 19, 256–264. doi:10.1016/j.procs.2013.06.037

Boyle, T., 2003. Design principles for authoring dynamic, reusable learning objects. Aust. J. Educ. Technol. 19, 46–58.

Boytchev, P., 2015. Deconstructionism in Education-A Personal Wandering towards Constructionism. Constructionism2014. ifs. tuwien. ac. at,.(2015). Retrieved.

Bransford, J., Stevens, R., Schwartz, D., Meltzoff, A., Pea, R., Roschelle, J., Vye, N., Kuhl, P., Bell, P., Reeves, B., Sabelli, N., 2006. Learning theories and education: Toward a decade of synergy, in: Alexander, P., Winne, P. (Eds.), Handbook of Educational Psychology. Erlbaum, pp. 209–244.

Brittain, N., 2011. IT education remains mired in uncertainty [WWW Document]. http://www.computing.co.uk. URL http://www.computing.co.uk/ctg/analysis/2110259/education-remains-mired-uncertainty (accessed 12.14.16).

Broll, W., Lindt, I., Herbst, I., Ohlenburg, J., Braun, A.K., Wetzel, R., 2008. Toward Next-Gen Mobile AR Games. IEEE Comput. Graph. Appl. 28, 40–48. doi:10.1109/MCG.2008.85

Bronack, S.C., 2011. The Role of Immersive Media in Online Education. J. Contin. High. Educ. 59, 113–117.

Burbaitė, R., Damaševičius, R., Štuikys, V., 2013. Using robots as learning objects for teaching computer science, in: X World Conference on Computers in Education (WCCE'13), Torun, Poland. pp. 103–111.

Burbaite, R., Stuikys, V., Damasevicius, R., 2013. Educational robots as collaborative learning objects for teaching Computer Science, in: 2013 International Conference on System Science and Engineering (ICSSE). Presented at the 2013 International Conference on System Science and Engineering (ICSSE), pp. 211–216. doi:10.1109/ICSSE.2013.6614661

Burnett, M.M., 2001. Visual Programming, in: Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, Inc.

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, Imperial College, London & The Royal Geographical Society. pp. 12–13.

Callaghan, V., Lear, M., Peña-Rios, A., Zamudio, V., Zheng, P., 2013. Putting the Buzz Back into Computer Science Education., in: Intelligent Environments (Workshops). pp. 454–465.

Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., Ivkovic, M., 2011. Augmented reality technologies, systems and applications. Multimed. Tools Appl. 51, 341–377. doi:10.1007/s11042-010-0660-6

Carolis, B.D., Cozzolongo, G., 2009. Interpretation of user's feedback in human-robot interaction. J. Phys. Agents 3, 47–58. doi:10.14198/JoPha.2009.3.2.08

Carozza, L., Tingdahl, D., Bosché, F., van Gool, L., 2014. Markerless Vision-Based Augmented Reality for Urban Planning. Comput.-Aided Civ. Infrastruct. Eng. 29, 2–17. doi:10.1111/j.1467-8667.2012.00798.x

Caudell, T.P., Mizell, D.W., 1992. Augmented reality: an application of heads-up display technology to manual manufacturing processes, in: Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences. Presented at the Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, pp. 659–669 vol.2. doi:10.1109/HICSS.1992.183317

Cervera, E., Martinet, P., Marin, R., Moughlbay, A.A., Pobil, A.P. del, Alemany, J., Esteller, R., Casañ, G., 2016. The Robot Programming Network. J. Intell. Robot. Syst. 81, 77–95. doi:10.1007/s10846-015-0201-7

Chalk, B., Clements, A., 2006. AJava VIRTUAL MACHINE LEARNING OBJECT TO SUPPORT INTRODUCTORY PROGRAMMING COMPUTING.

Chalk, B., Fraser, K., 2006. A Survey on the teaching of introductory programming in Higher Education. Proceeding JICC10 1–6.

Chan, J., Pondicherry, T., Blikstein, P., 2013. LightUp: an augmented, learning platform for electronics. ACM Press, pp. 491–494. doi:10.1145/2485760.2485812

Chastine, J.W., Nagel, K., Zhu, Y., Yearsovich, L., 2007. Understanding the Design Space of Referencing in Collaborative Augmented Reality Environments, in: Proceedings of Graphics Interface 2007, GI '07. ACM, New York, NY, USA, pp. 207–214. doi:10.1145/1268517.1268552

Chen, F.-G., Chen, K.-Y., Lin, C.-Y., 2010. A cooperative learning of computer networking with portable laboratories using virtual machine, in: 2010 3rd International Conference on Computer Science and Information Technology.

Presented at the 2010 3rd International Conference on Computer Science and Information Technology, pp. 118–122. doi:10.1109/ICCSIT.2010.5564422

Chen, I.Y.H., MacDonald, B., Wunsche, B., 2009. Mixed reality simulation for mobile robots, in: 2009 IEEE International Conference on Robotics and Automation. Presented at the 2009 IEEE International Conference on Robotics and Automation, pp. 232–237. doi:10.1109/ROBOT.2009.5152325

Chen, M.-P., Wu, C.-C., Chiu, C.-H., Lee, H.-L., Chen, M.-C., 2009. Teaching abstract computer science concepts to high school students: A multiple-perspective approach. Int. J. Digit. Learn. Technol. 1, 266–286.

Chen, Y.-C., 2006. A Study of Comparing the Use of Augmented Reality and Physical Models in Chemistry Education, in: Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications, VRCIA ’06. ACM, New York, NY, USA, pp. 369–372. doi:10.1145/1128923.1128990

Chen, Y.-C., Chi, H.-L., Hung, W.-H., Kang, S.-C., 2011. Use of Tangible and Augmented Reality Models in Engineering Graphics Courses. J. Prof. Issues Eng. Educ. Pract. 137, 267–276. doi:10.1061/(ASCE)EI.1943-5541.0000078

Cheng, K.-H., Tsai, C.-C., 2012. Affordances of Augmented Reality in Science Learning: Suggestions for Future Research. J. Sci. Educ. Technol. 22, 449–462. doi:10.1007/s10956-012-9405-9

Chin, J., Callaghan, V., Clarke, G., 2010. End-user Customisation of Intelligent Environments, in: Nakashima, H., Aghajan, H., Augusto, J.C. (Eds.), Handbook of Ambient Intelligence and Smart Environments. Springer US, pp. 371–407. doi:10.1007/978-0-387-93808-0_14

Chin, J., Callaghan, V., Clarke, G., 2009. Soft-appliances: A Vision for User Created Networked Appliances in Digital Homes. J Ambient Intell Smart Env. 1, 69–75.

Chin, J.S.Y., Callaghan, V., Clarke, G., 2008. A programming-by-example approach to customising digital homes. Presented at the 2008 IET 4th International Conference on Intelligent Environments, pp. 1–8.

Clarebout, G., 2007. In Search of Pedagogical Agents’ Modality and Dialogue Effects in Open Learning Environments. E-J. Instr. Sci. Technol. 10.

Clarisse, O., Chang, S.-K., 1986. Vicon, in: Chang, S.-K., Ichikawa, T., Ligomenides, P.A. (Eds.), Visual Languages, Management and Information Systems. Springer US, pp. 151–190. doi:10.1007/978-1-4613-1805-7_7

Clark, A., Chalmers, D., 1998. The Extended Mind. Analysis 58, 7–19.

Coelho, E.M., Julier, S.J., Maclntyre, B., 2004. OSGAR: a scene graph with uncertain transformations, in: Third IEEE and ACM International Symposium on Mixed and Augmented Reality. Presented at the Third IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 6–15. doi:10.1109/ISMAR.2004.44

Collett, T.H.J., MacDonald, B.A., 2010. An Augmented Reality Debugging System for Mobile Robot Software Engineers. J. Softw. Eng. Robot. 1, 18–32.

Collins, A., Halverson, R., 2009. Rethinking Education in the Age of Technology: The Digital Revolution and Schooling in America. Teachers College Press, New York, NY, USA.

Conklin, J., 2005. Review of A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives Complete Edition. Educ. Horiz. 83, 154–159.

Consortium, I.G.., 2003. IMS learning design best practice and implementation guide, Final Specification.

Cooperstock, J.R., 2001. The Classroom of the Future: Enhancing Education through Augmented Reality, in: Proc. HCI Inter. 2001 Conf. on Human-Computer Interaction. pp. 688–692.

Cox, B.J., 1984. Message/Object Programming: An Evolutionary Change in Programming Technology. IEEE Softw 1, 50–61. doi:10.1109/MS.1984.233398

Cox, B.J., Novobilski, A., 1991. Object-Oriented Programming; An Evolutionary Approach, 2nd ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Daily, M., Cho, Y., Martin, K., Payton, D., 2003. World embedded interfaces for human-robot interaction, in: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003. Presented at the Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003, p. 6 pp.-. doi:10.1109/HICSS.2003.1174285

Dalgarno, B., Hedberg, J., Harper, B., 2002. The contribution of 3D environments to conceptual understanding. Fac. Educ. - Pap. Arch. 149–158.

Dalgarno, B., Lee, M.J.W., 2010. What are the learning affordances of 3-D virtual environments? Br. J. Educ. Technol. 41, 10–32. doi:10.1111/j.1467-8535.2009.01038.x

Dede, C., 2009. Immersive Interfaces for Engagement and Learning. Science 323, 66–69. doi:10.1126/science.1167311

Dede, C., 2008. Learning via Smart Objects, Intelligent Contexts, and Ubiquitous Computing. Educ. Technol. 3.

Dede, C., Salzman, M.C., Loftin, R.B., Sprague, D., 1999. Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts, in: Feurzeig, W., Roberts, N. (Eds.), Modeling and Simulation in Science and Mathematics Education, Modeling Dynamic Systems. Springer New York, pp. 282–319. doi:10.1007/978-1-4612-1414-4_12

Delman, A., Goetz, L., Langsam, Y., Raphan, T., 2009. Development of a System for Teaching C/C++ Using Robots and Open Source Software in a CS1 Course., in: International Conference on Frontiers in Education: Computer Science & Computer Engineering. pp. 141–146.

Diegmann, P., Schmidt-Kraepelin, M., Van den Eynden, S., Basten, D., 2015. Benefits of Augmented Reality in Educational Environments-A Systematic Literature Review., in: Wirtschaftsinformatik. pp. 1542–1556.

Dobrilovic, D., Jevtic, V., Stojanov, Z., Odadzic, B., 2012. Usability of virtual network laboratory in engineering education and computer network course, in: 2012 15th International Conference on Interactive Collaborative Learning (ICL). Presented at the 2012 15th International Conference on Interactive Collaborative Learning (ICL), pp. 1–6. doi:10.1109/ICL.2012.6402049

Doswell, J.T., 2005. It’s virtually pedagogical: pedagogical agents in mixed reality learning environments, in: ACM SIGGRAPH 2005 Educators Program. ACM, p. 25.

Dunleavy, M., Dede, C., Mitchell, R., 2009. Affordances and Limitations of Immersive Participatory Augmented Reality Simulations for Teaching and Learning. J. Sci. Educ. Technol. 18, 7–22. doi:10.1007/s10956-008-9119-1

Dunleavy, M., Simmons, B., 2011. Assessing Learning and Identity in Augmented Reality Science Games, in: Annetta, L., Bronack, S.C. (Eds.), Serious

Educational Game Assessment. SensePublishers, pp. 221–240. doi:10.1007/978-94-6091-329-7_14

Dünser, A., Grasset, R., Billinghurst, M., 2008. A survey of evaluation techniques used in augmented reality studies, in: SIGGRAPH Asia ’08. ACM, New York, NY, USA, p. 5:1–5:27. doi:10.1145/1508044.1508049

Dünser, A., Grasset, R., Seichter, H., Billinghurst, M., 2007. Applying HCI principles to AR systems design. Presented at the HIT Lab NZ, University of Canterbury, New Zealand.

Dünser, A., Walker, L., Horner, H., Bentall, D., 2012. Creating Interactive Physics Education Books with Augmented Reality, in: Proceedings of the 24th Australian Computer-Human Interaction Conference, OzCHI ’12. ACM, New York, NY, USA, pp. 107–114. doi:10.1145/2414536.2414554

Elliott, J.B., Gardner, M., Alrashidi, M., 2012. Towards a framework for the design of mixed reality immersive education spaces. 2nd Eur. Immersive Educ. Summit 63–76.

Ellis, S.R., Bucher, U.J., 1994. Distance Perception of Stereoscopically Presented Virtual Objects Optically Superimposed on Physical Objects by a Head-Mounted See-Through Display. Proc. Hum. Factors Ergon. Soc. Annu. Meet. 38, 1300–1304. doi:10.1177/154193129403801911

Facer, K., Joiner, R., Stanton, D., Reid, J., Hull, R., Kirk, D., 2004. Savannah: mobile gaming and learning? J. Comput. Assist. Learn. 20, 399–409. doi:10.1111/j.1365-2729.2004.00105.x

Fardinpour, A., Dreher, H., 2012. Towards Developing Automated Assessment in Virtual Learning Environments (SSRN Scholarly Paper No. ID 2412847). Social Science Research Network, Rochester, NY.

Feiner, S., Macintyre, B., Seligmann, D., 1993. Knowledge-based augmented reality. Commun ACM 36, 53–62. doi:10.1145/159544.159587

Feldman, A., Tapia, E.M., Sadi, S., Maes, P., Schmandt, C., 2005. ReachMedia: On-the-move Interaction with Everyday Objects, in: Proceedings of the Ninth IEEE International Symposium on Wearable Computers, ISWC ’05. IEEE Computer Society, Washington, DC, USA, pp. 52–59. doi:10.1109/ISWC.2005.44

Ferscha, A., Keller, M., 2003. Real Time Inspection of Hidden Worlds, in: Proceedings of the Seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications, DS-RT ’03. IEEE Computer Society, Washington, DC, USA, p. 51–.

Furió, D., González-Gancedo, S., Juan, M.-C., Seguí, I., Rando, N., 2013. Evaluation of learning outcomes using an educational iPhone game vs. traditional game. Comput. Educ. 64, 1–23. doi:10.1016/j.compedu.2012.12.001

Gabbard, J.L., Hix, D., Swan, J.E., 1999. User-centered design and evaluation of virtual environments. IEEE Comput. Graph. Appl. 19, 51–59. doi:10.1109/38.799740

Gaspar, A., Langevin, S., Armitage, W.D., Rideout, M., 2008. March of the (virtual) machines: past, present, and future milestones in the adoption of virtualization in computing education. J. Comput. Sci. Coll. 23, 123–132.

Groesbrink, S., Oberthür, S., Baldin, D., 2013. Architecture for Adaptive Resource Assignment to Virtualized Mixed-criticality Real-time Systems. SIGBED Rev 10, 18–23. doi:10.1145/2492385.2492388

Han, P., Krämer, B.J., 2009. Generating Interactive Learning Objects from Configurable Samples, in: Proceedings of the 2009 International Conference on Mobile, Hybrid, and On-Line Learning, ELML ’09. IEEE Computer Society, Washington, DC, USA, pp. 1–6. doi:10.1109/eLmL.2009.9

Harel, I., Papert, S., 1991. Constructionism. Ablex Publishing.

Hart, S.G., Staveland, L.E., 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research, in: Meshkati, P.A.H. and N. (Ed.), Advances in Psychology, Human Mental Workload. North-Holland, pp. 139–183.

Hatcher, S., 2012. Teaching with Objects and Photographs: A Guide for Teachers. Second Edition.

Heiser, J., Phan, D., Agrawala, M., Tversky, B., Hanrahan, P., 2004. Identification and Validation of Cognitive Design Principles for Automated Generation of Assembly Instructions, in: Proceedings of the Working Conference on Advanced Visual Interfaces, AVI ’04. ACM, New York, NY, USA, pp. 311–319. doi:10.1145/989863.989917

Henderson, S.J., Feiner, S., 2009. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret, in: 2009 8th IEEE International Symposium on Mixed and Augmented Reality. Presented at the 2009 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 135–144. doi:10.1109/ISMAR.2009.5336486

Henderson, S.J., Feiner, S.K., 2011. Augmented reality in the psychomotor phase of a procedural task, in: 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR). Presented at the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 191–200. doi:10.1109/ISMAR.2011.6092386

Henriksen, P., Kölling, M., 2004. Greenfoot: Combining Object Visualisation with Interaction, in: Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA ’04. ACM, New York, NY, USA, pp. 73–82. doi:10.1145/1028664.1028701

Henrysson, A., Billinghurst, M., Ollila, M., 2005. Face to face collaborative AR on mobile phones, in: Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’05). Presented at the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’05), pp. 80–89. doi:10.1109/ISMAR.2005.32

Heun, V., Kasahara, S., Maes, P., 2013. Smarter Objects: Using AR Technology to Program Physical Objects and Their Interactions, in: CHI ’13 Extended Abstracts on Human Factors in Computing Systems, CHI EA ’13. ACM, New York, NY, USA, pp. 2939–2942. doi:10.1145/2468356.2479579

Hewett, T.T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., Verplank, W., 1992. ACM SIGCHI curricula for human-computer interaction. ACM.

Holz, T., Dragone, M., O’Hare, G.M.P., 2009. Where Robots and Virtual Agents Meet. Int. J. Soc. Robot. 1, 83–93. doi:10.1007/s12369-008-0002-2

Hu, X., Wang, M., Xu, Y., Qian, K., 2012. Modular design and adoption of embedded system courseware with portable labs in a box, in: Proc. World Congress on Engineering and Computer Science (WCECS).

Ibáñez, M.B., Di Serio, Á., Villarán, D., Delgado Kloos, C., 2014. Experimenting with Electromagnetism Using Augmented Reality: Impact on Flow Student Experience and Educational Effectiveness. Comput Educ 71, 1–13. doi:10.1016/j.compedu.2013.09.004

IEEE Standard for Learning Object Metadata, 2002. . IEEE Std 1484121-2002 i-32. doi:10.1109/IEEESTD.2002.94128

Irawati, S., Green, S., Billinghurst, M., Duenser, A., Ko, H., 2006. "Move the couch where?" : developing an augmented reality multimodal interface, in: 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality. Presented at the 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 183–186. doi:10.1109/ISMAR.2006.297812

Janin, A.L., Mizell, D.W., Caudell, T.P., 1993. Calibration of head-mounted displays for augmented reality applications, in: Proceedings of IEEE Virtual Reality Annual International Symposium. Presented at the Proceedings of IEEE Virtual Reality Annual International Symposium, pp. 246–255. doi:10.1109/VRAIS.1993.380772

Jara, C.A., Candelas, F.A., Torres, F., 2008. An advanced interactive interface for robotics elearning. Int. J. Online Eng. IJOE 4. doi:10.3991/ijoe.v4i4.561

Johnson, L., Smith, R., Willis, H., Levine, A., Haywood, K., 2011. The 2011 Horizon Report. New Media Consortium.

Johnson, L.F., Levine, A., Smith, R.S., Haywood, K., 2010. Key Emerging Technologies for Elementary and Secondary Education. Tech Dir. 70, 33–34,.

Jones, V., Jo, J.H., 2004. Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology, in: Beyond the Comfort Zone: Proceedings of the 21st ASCILITE Conference. Perth, Western Australia, p. 474.

Joy, E.H., Garcia, F.E., 2000. Measuring learning effectiveness: A new look at no-significant-difference findings. J. Asynchronous Learn. Netw. 4, 33–39.

Juan, C.M., Toffetti, G., Abad, F., Cano, J., 2010. Tangible Cubes Used as the User Interface in an Augmented Reality Game for Edutainment, in: 2010 10th IEEE International Conference on Advanced Learning Technologies. Presented at the 2010 10th IEEE International Conference on Advanced Learning Technologies, pp. 599–603. doi:10.1109/ICALT.2010.170

Kamarainen, A.M., Metcalf, S., Grotzer, T., Browne, A., Mazzuca, D., Tutwiler, M.S., Dede, C., 2013. EcoMOBILE: Integrating augmented reality and probeware with environmental education field trips. Comput. Educ. 68, 545–556. doi:10.1016/j.compedu.2013.02.018

Kato, H., Billinghurst, M., 1999. Marker tracking and HMD calibration for a video-based augmented reality conferencing system, in: 2nd IEEE and ACM International Workshop on Augmented Reality, 1999. (IWAR '99) Proceedings. Presented at the 2nd IEEE and ACM International Workshop on Augmented Reality, 1999. (IWAR '99) Proceedings, pp. 85–94. doi:10.1109/IWAR.1999.803809

Kaufmann, H., Schmalstieg, D., 2006. Designing Immersive Virtual Reality for Geometry Education, in: IEEE Virtual Reality Conference (VR 2006). Presented at the IEEE Virtual Reality Conference (VR 2006), pp. 51–58. doi:10.1109/VR.2006.48

Kaufmann, H., Schmalstieg, D., 2003. Mathematics and geometry education with collaborative augmented reality. Comput. Graph. 27, 339–345. doi:10.1016/S0097-8493(03)00028-1

Kenkre, A., Banerjee, G., Mavinkurve, M., Murthy, S., 2012. Identifying Learning Object Pedagogical Features to Decide Instructional Setting, in: 2012 IEEE Fourth International Conference on Technology for Education. Presented at the 2012 IEEE Fourth International Conference on Technology for Education, pp. 46–53. doi:10.1109/T4E.2012.8

Kerawalla, L., Luckin, R., Seljeflot, S., Woolard, A., 2006. “Making it real”: exploring the potential of augmented reality for teaching primary school science. Virtual Real. 10, 163–174. doi:10.1007/s10055-006-0036-4

Khairuddin, N.N., Hashim, K., 2008. Application of Bloom’s Taxonomy in Software Engineering Assessments, in: Proceedings of the 8th Conference on Applied Computer Scince, ACS’08. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 66–69.

Kim, S., Song, S.-M., Yoon, Y.-I., 2011. Smart Learning Services Based on Smart Cloud Computing. Sensors 11, 7835–7850. doi:10.3390/s110807835

Kim, Y., Baylor, A.L., Group, P., 2006. Pedagogical Agents as Learning Companions: The Role of Agent Competency and Type of Interaction. Educ. Technol. Res. Dev. 54, 223–243. doi:10.1007/s11423-006-8805-z

Kim, Y., Wei, Q., 2011. The impact of learner attributes and learner choice in an agent-based environment. Comput. Educ. 56, 505–514. doi:10.1016/j.compedu.2010.09.016

Kizilkaya, G., Askar, P., 2008. The effect of an embedded pedagogical agent on the students’ science achievement. Interact. Technol. Smart Educ. 5, 208–216. doi:10.1108/17415650810930893

Klopfer, E., Sheldon, J., 2010. Augmenting your own reality: Student authoring of science-based augmented reality games. New Dir. Youth Dev. 2010, 85–94.

Klopfer, E., Squire, K., 2008. Environmental Detectives—the development of an augmented reality platform for environmental simulations. Educ. Technol. Res. Dev. 56, 203–228. doi:10.1007/s11423-007-9037-6

Konak, A., Bartolacci, M., 2016. Using a Virtual Computing Laboratory to Foster Collaborative Learning for Information Security and Information Technology Education. J. Cybersecurity Educ. Res. Pract. 2016.

Koohang, A., Harman, K., 2007. Learning Objects and Instructional Design. Informing Science.

Kortuem, G., Kawsar, F., Sundramoorthy, V., Fitton, D., 2010. Smart objects as building blocks for the Internet of things. IEEE Internet Comput. 14, 44–51. doi:10.1109/MIC.2009.143

Kotranza, A., Lind, D.S., Pugh, C.M., Lok, B., 2009. Real-time in-situ visual feedback of task performance in mixed environments for learning joint psychomotor-cognitive tasks, in: 2009 8th IEEE International Symposium on Mixed and Augmented Reality. Presented at the 2009 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 125–134. doi:10.1109/ISMAR.2009.5336485

Krämer, B.J., Schmidt, H.-W., 2001. Components and Tools for On-Line Education. Eur. J. Educ. 36, 195–222.

Krämer, N.C., Bente, G., 2010. Personalizing e-Learning. The Social Effects of Pedagogical Agents. Educ. Psychol. Rev. 22, 71–87. doi:10.1007/s10648-010-9123-x

Krueger, M.W., Gionfriddo, T., Hinrichsen, K., 1985. VIDEOPLACE—an Artificial Reality, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’85. ACM, New York, NY, USA, pp. 35–40. doi:10.1145/317456.317463

Kurz, D., Benhimane, S., 2012. Handheld Augmented Reality involving gravity measurements. Comput. Graph., Augmented RealityComputer Graphics in China 36, 866–883. doi:10.1016/j.cag.2012.03.038

Lalonde, J., Bartley, C.P., Nourbakhsh, I., 2006. Mobile robot programming in education, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, pp. 345–350.

Lambert, J.M., Power, J.F., 2008. Platform Independent Timing of Java Virtual Machine Bytecode Instructions. Electron. Notes Theor. Comput. Sci., Proceedings of the Sixth Workshop on Quantitative Aspects of Programming Languages (QAPL 2008) 220, 97–113. doi:10.1016/j.entcs.2008.11.021

Landowska, A., 2010. Student model representation for pedagogical virtual mentors, in: 2010 2nd International Conference on Information Technology, (2010 ICIT). Presented at the 2010 2nd International Conference on Information Technology, (2010 ICIT), pp. 61–64.

Lattu, M., Meisalo, V., Tarhio, J., 2003. A visualisation tool as a demonstration aid. Comput. Educ. 41, 133–148. doi:10.1016/S0360-1315(03)00032-0

Lau, W.W., Yuen, A.H., 2009. Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. Br. J. Educ. Technol. 40, 696–712.

Laugwitz, B., Held, T., Schrepp, M., 2008. Construction and Evaluation of a User Experience Questionnaire, in: Holzinger, A. (Ed.), HCI and Usability for Education and Work, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 63–76. doi:10.1007/978-3-540-89350-9_6

Lee, Y., Oh, S., Shin, C., Woo, W., 2009. Ubiquitous virtual reality and its key dimension, in: International Workshop on Ubiquitous Virtual Reality. pp. 5–8.

Levin, I., Kolberg, E., Reich, Y., 2004. Robot control teaching with a state machine-based design method. Int. J. Eng. Educ. 234–243.

Li, P., Mohammed, T., 2008. Integration of virtualization technology into network security laboratory, in: 2008 38th Annual Frontiers in Education Conference. Presented at the 2008 38th Annual Frontiers in Education Conference, p. S2A–7–S2A–12. doi:10.1109/FIE.2008.4720550

Li, S., Xu, L., Wang, X., Wang, J., 2012. Integration of hybrid wireless networks in cloud services oriented enterprise information systems. Enterp. Inf. Syst. 6, 165–187. doi:10.1080/17517575.2011.654266

Li, S., Xu, L.D., Zhao, S., 2014. The internet of things: a survey. Inf. Syst. Front. 17, 243–259. doi:10.1007/s10796-014-9492-7

Li, X., Flatt, M., 2015. Medic: Metaprogramming and Trace-oriented Debugging, in: Proceedings of the Workshop on Future Programming, FPW 2015. ACM, New York, NY, USA, pp. 7–14. doi:10.1145/2846656.2846658

Ligeour, v.l., otmane, s., mallem, m., 2005. Augmented reality interface for free teleoperation. IFAC Proc. Vol., 16th IFAC World Congress 38, 523–528. doi:10.3182/20050703-6-CZ-1902.01357

Lin, H.-C.K., Hsieh, M.-C., Wang, C.-H., Sie, Z.-Y., Chang, S.-H., 2011. Establishment and Usability Evaluation of an Interactive AR Learning System on Conservation of Fish. Turk. Online J. Educ. Technol. - TOJET 10, 181–187.

Lister, R., Leaney, J., 2003. Introductory Programming, Criterion-referencing, and Bloom, in: Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '03. ACM, New York, NY, USA, pp. 143–147. doi:10.1145/611892.611954

Liu, A.S., Schunn, C.D., Flot, J., Shoop, R., 2013. The role of physicality in rich programming environments. Comput. Sci. Educ. 23, 315–331. doi:10.1080/08993408.2013.847165

Liu, C., Huot, S., Diehl, J., Mackay, W., Beaudouin-Lafon, M., 2012. Evaluating the Benefits of Real-time Feedback in Mobile Augmented Reality with Hand-held Devices, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12. ACM, New York, NY, USA, pp. 2973–2976. doi:10.1145/2207676.2208706

Liu, T.-Y., Tan, T.-H., Chu, Y.-L., 2009. Outdoor Natural Science Learning with an RFID-Supported Immersive Ubiquitous Learning Environment. Educ. Technol. Soc. 12, 161–175.

Macdonald, M.A., 2004. An analytic tool for deconstructing teaching and learning tasks.

Magnenat, S., Ben-Ari, M., Klinger, S., Sumner, R.W., 2015. Enhancing Robot Programming with Visual Feedback and Augmented Reality, in: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '15. ACM, New York, NY, USA, pp. 153–158. doi:10.1145/2729094.2742585

Maloney, D.P., O'Kuma, T.L., Hieggelke, C.J., Heuvelen, A.V., 2001. Surveying students' conceptual knowledge of electricity and magnetism. Am. J. Phys. 69, S12–S23. doi:10.1119/1.1371296

Marín, R., Sanz, P.J., del Pobil, A.P., 2003. The UJI Online Robot: An Education and Training Experience. Auton. Robots 15, 283–297. doi:10.1023/A:1026220621431

Martin, S., Diaz, G., Sancristobal, E., Gil, R., Castro, M., Peire, J., 2011. New technology trends in education: Seven years of forecasts and convergence. Comput. Educ. 57, 1893–1906. doi:10.1016/j.compedu.2011.04.003

Martín-Gutiérrez, J., Luís Saorín, J., Contero, M., Alcañiz, M., Pérez-López, D.C., Ortega, M., 2010. Design and validation of an augmented book for spatial abilities development in engineering students. Comput. Graph. 34, 77–91. doi:10.1016/j.cag.2009.11.003

Martins, I., Morais, F. de, Schaab, B., Jaques, P., 2016. Pedagogical Agent Gestures to Improve Learner Comprehension of Abstract Concepts in Hints. Int. J. Inf. Commun. Technol. Educ. IJICTE 12, 65–75. doi:10.4018/IJICTE.2016070106

Mateljan, V., Juricic, V., Moguljak, M., 2014. Virtual machines in education, in: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Presented at the 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 603–607. doi:10.1109/MIPRO.2014.6859639

Mattern, F., Floerkemeier, C., 2010. From the Internet of Computers to the Internet of Things, in: Sachs, K., Petrov, I., Guerrero, P. (Eds.), From Active Data Management to Event-Based Systems and More, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 242–259. doi:10.1007/978-3-642-17226-7_15

Mauch, E., 2001. Using Technological Innovation to Improve the Problem-Solving Skills of Middle School Students: Educators' Experiences with the LEGO Mindstorms Robotic Invention System. Clear. House J. Educ. Strateg. Issues Ideas 74, 211–213. doi:10.1080/00098650109599193

Mayer, S., Beckel, C., Scheidegger, B., Barthels, C., Sörös, G., 2012. Demo: Uncovering Device Whispers in Smart Homes, in: Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, MUM '12. ACM, New York, NY, USA, p. 56:1–56:3. doi:10.1145/2406367.2406435

Mayer, S., Schalch, M., George, M., Sörös, G., 2013. Device Recognition for Intuitive Interaction with the Web of Things, in: Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, UbiComp '13 Adjunct. ACM, New York, NY, USA, pp. 239–242. doi:10.1145/2494091.2494168

McEwan, W., 2002. Virtual machine technologies and their application in the delivery of ict, in: Proceedings of the 15th Annual NACCQ. Hamilton, New Zealand.

Medenica, Z., Kun, A.L., Paek, T., Palinko, O., 2011. Augmented reality vs. street views: a driving simulator study comparing two emerging navigation aids, in: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. ACM, pp. 265–274.

Milgram, P., Kishino, F., 1994. A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS.

Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I., 2012. Internet of things: Vision, applications and research challenges. Ad Hoc Netw. 10, 1497–1516. doi:10.1016/j.adhoc.2012.02.016

Mishra, V.K., Mishra, M., Sharma, H.R., 2010. Active learning environment: Gain the most from PBL for teaching virtual machine module on Operating System, in: 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010). Presented at the 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), pp. 378–382. doi:10.1109/PDGC.2010.5679967

Mistry, P., Maes, P., 2009. SixthSense: A Wearable Gestural Interface, in: ACM SIGGRAPH ASIA 2009 Sketches, SIGGRAPH ASIA '09. ACM, New York, NY, USA, p. 11:1–11:1. doi:10.1145/1667146.1667160

Moon, D.A., 1986. Object-oriented Programming with Flavors, in: Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications, OOPSLA '86. ACM, New York, NY, USA, pp. 1–8. doi:10.1145/28697.28698

Mooser, J., Wang, L., You, S., Neumann, U., 2007. An Augmented Reality Interface for Mobile Information Retrieval, in: 2007 IEEE International Conference on Multimedia and Expo. Presented at the 2007 IEEE International Conference on Multimedia and Expo, pp. 2226–2229. doi:10.1109/ICME.2007.4285128

Morozov, M., Tanakov, A., Bystrov, D., 2003. Pedagogical agents in the multimedia natural sciences for children, in: Proceedings 3rd IEEE International Conference on Advanced Technologies. Presented at the Proceedings 3rd IEEE International Conference on Advanced Technologies, pp. 62–65. doi:10.1109/ICALT.2003.1215027

Müller, D., Bruns, F.W., Erbe, H.-H., Robben, B., Yoo, Y.-H., 2007. Mixed reality learning spaces for collaborative experimentation: A challenge for engineering education and training. Int. J. Online Eng. IJOE 3.

Munnerley, D., Bacon, M., Fitzgerald, R., Wilson, A., Hedberg, J., Steele, J., 2015. Augmented Reality: Application in Higher Education : Final Report 2014.

Murray, M., Tenenbaum, G., 2010. Computerized Pedagogical Agents as an Educational Means for Developing Physical Self-Efficacy and Encouraging Activity in Youth. J. Educ. Comput. Res. 42, 267–283. doi:10.2190/EC.42.3.b

Nabhen, R., Maziero, C., 2006. Some Experiences in Using Virtual Machines for Teaching Computer Networks, in: Education for the 21st Century — Impact of ICT and Digital Resources. Springer, Boston, MA, pp. 93–104. doi:10.1007/978-0-387-34731-8_11

Nasereddin, M., Clark, T.K., Konak, A., 2014. Using virtual machines in a K-12 Outreach program to increase interest in information security fields, in: 2014 IEEE Integrated STEM Education Conference. Presented at the 2014 IEEE Integrated STEM Education Conference, pp. 1–5. doi:10.1109/ISECon.2014.6891039

Neumann, U., Majoros, A., 1998. Cognitive, Performance, and Systems Issues for Augmented Reality Applications in Manufacturing and Maintenance, in: Proceedings of the Virtual Reality Annual International Symposium, VRAIS ’98. IEEE Computer Society, Washington, DC, USA, p. 4–.

Neven, F., Duval, E., 2002. Reusable Learning Objects: A Survey of LOM-based Repositories, in: Proceedings of the Tenth ACM International Conference on Multimedia, MULTIMEDIA ’02. ACM, New York, NY, USA, pp. 291–294. doi:10.1145/641007.641067

Ngeow, K., Kong, Y.-S., 2001. Learning To Learn: Preparing Teachers and Students for Problem-Based Learning. ERIC Digest.

Nicklas, D., Großmann, M., Schwarz, T., Volz, S., Mitschang, B., 2001. A Model-Based, Open Architecture for Mobile, Spatially Aware Applications, in: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, SSTD ’01. Springer-Verlag, London, UK, UK, pp. 117–135.

Norman, D.A., 2004. Emotional design: Why we love (or hate) everyday things. Basic books.

Novak, D., Wang, M., Callaghan, V., 2012. Looking In, Looking Out: A Discussion of the Educational Affordances of Current Mobile Augmented Reality Technologies, in: Educational Stages and Interactive Learning: From Kindergarten to Workplace Training. IGI Global, pp. 92–106.

NXP Semiconductors, 2014. UM10204 I^2C-bus specification and user manual.

Oh, S., Byun, Y.C., 2012. The Design and Implementation of Augmented Reality Learning Systems, in: 2012 IEEE/ACIS 11th International Conference on Computer and Information Science. Presented at the 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, pp. 651–654. doi:10.1109/ICIS.2012.106

Oliver, D., Dobele, T., Greber, M., Roberts, T., 2004. This Course Has a Bloom Rating of 3.9, in: Proceedings of the Sixth Australasian Conference on Computing Education - Volume 30, ACE ’04. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 227–231.

Omoda-Onyait, G., Lubega, J.T., Maiga, G., Angole, R.O., 2012. Towards an Interactive Agent-Based Approach to Real-Time Feedback (IAARF) in E-Learning System, in: Hybrid Learning. Presented at the International Conference on Hybrid Learning, Springer, Berlin, Heidelberg, pp. 317–328. doi:10.1007/978-3-642-32018-7_30

Onime, C., Abiona, O., 2016. 3D Mobile Augmented Reality Interface for Laboratory Experiments. Int. J. Commun. Netw. Syst. Sci. 9, 67. doi:10.4236/ijcns.2016.94006

O'Shea, P.M., Dede, C., Cherian, M., 2011. Research Note: The Results of Formatively Evaluating an Augmented Reality Curriculum Based on Modified Design Principles. Int J Gaming Comput Mediat Simul 3, 57–66. doi:10.4018/jgcms.2011040104

O'Shea, T., Koschmann, T., 1997. The Children's Machine: Rethinking School in the Age of the Computer (book). J. Learn. Sci. 6, 401–415.

Papert, S., 1991. Situating Constructionism.

Pastoor, S., Conomis, C., 2005. Mixed Reality Displays, in: Schreer, O., Kauff, P., Sikora, T. (Eds.), 3D Videocommunication. John Wiley & Sons, Ltd, pp. 261–280.

Peña-rios, a., callaghan, v., gardner, m., alhaddad, m.j., 2013. xReality interactions within an immersive blended reality learning space, in: 3rd European Immersive Education Summit (iED). p. 26.

Pena-Rios, A., Callaghan, V., Gardner, M., Alhaddad, M.J., 2012. Towards the Next Generation of Learning Environments: An InterReality Learning Portal and Model, in: 2012 8th International Conference on Intelligent Environments (IE). Presented at the 2012 8th International Conference on Intelligent Environments (IE), pp. 267–274. doi:10.1109/IE.2012.31

Peña-Ríos, A., Callaghan, V., Gardner, M., Alhaddad, M.J., 2012. Remote Mixed Reality Collaborative Laboratory Activities: Learning Activities within the InterReality Portal, in: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT). Presented at the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 362–366. doi:10.1109/WI-IAT.2012.43

Perry, J., Klopfer, E., Norton, M., Sutch, D., Sandford, R., Facer, K., 2008. AR Gone Wild: Two Approaches to Using Augmented Reality Learning Games in Zoos, in: Proceedings of the 8th International Conference on International Conference for the Learning Sciences - Volume 3, ICLS'08. International Society of the Learning Sciences, Utrecht, The Netherlands, pp. 322–329.

Pfeifer, R., Scheier, C., 1999. Understanding Intelligence. MIT Press, Cambridge, MA, USA.

Plauska, I., Damaševičius, R., 2014. Educational Robots for Internet-of-Things Supported Collaborative Learning, in: Dregvaite, G., Damasevicius, R. (Eds.), Information and Software Technologies, Communications in Computer and Information Science. Springer International Publishing, pp. 346–358. doi:10.1007/978-3-319-11958-8_28

Pokkunuri, B.P., 1989. Object Oriented Programming. SIGPLAN Not 24, 96–101. doi:10.1145/71605.71612

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K.J., Proulx, V., Carlisle, M., 2006. Tools for Teaching Introductory Programming: What Works?, in: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '06. ACM, New York, NY, USA, pp. 560–561. doi:10.1145/1121341.1121514

Prensky, M., 2003. Digital Game-based Learning. Comput Entertain 1, 21–21. doi:10.1145/950566.950596

Python Software Foundation, 2001. Python Software Foundation. Learn more about Python.

Qian, K., Hu, X., Hong, L., 2011. Experience on teaching multiple CS courses with portable embedded system labware in a box, in: Proceedings of the World Congress on Engineering and Computer Science. pp. 19–21.

Qian, K., Lo, C.-T.D., Hu, X., 2010. Portable labs in a box for embedded system education, in: Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education. ACM, pp. 318–318.

Qu, L., Wang, N., Johnson, W.L., 2004. Choosing when to interact with learners, in: Proceedings of the 9th International Conference on Intelligent User Interfaces. ACM, pp. 307–309.

Radu, I., 2014. Augmented reality in education: a meta-review and cross-media analysis. Pers. Ubiquitous Comput. 18, 1533–1543. doi:10.1007/s00779-013-0747-y

Reitmayr, G., Schmalstieg, D., 2003. Location Based Applications for Mobile Augmented Reality, in: Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003 - Volume 18, AUIC '03. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 65–73.

Rekimoto, J., Ayatsuka, Y., 2000. CyberCode: designing augmented reality environments with visual tags, in: Proceedings of DARE 2000 on Designing Augmented Reality Environments, DARE '00. ACM, New York, NY, USA, pp. 1–10. doi:10.1145/354666.354667

Resnick, M., 1990. MultiLogo: A Study of Children and Concurrent Programming. Interact. Learn. Environ. 1, 153–170. doi:10.1080/104948290010301

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y., 2009. Scratch: Programming for All. Commun ACM 52, 60–67. doi:10.1145/1592761.1592779

Resnick, M., Ocko, S., Papert, S., 1988. LEGO, LOGO, AND DESIGN. Child. Environ. Q. 5, 14–18.

Rickel, J., Johnson, W.L., 1998. STEVE: A Pedagogical Agent for Virtual Reality, in: In Proceedings of the Second International Conference on Autonomous Agents.

Riel, A.J., 1996. Object-Oriented Design Heuristics, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Rizzo, A., Kim, G.J., Yeh, S.-C., Thiebaux, M., Hwang, J., Buckwalter, J.G., 2005. Development of a Benchmarking Scenario for Testing 3D User Interface Devices and Interaction Methods, in: Proceedings of the 11th International Conference on Human Computer Interaction.

Rogers, C., Portsmore, M., 2004. Bringing Engineering to Elementary School. J. STEM Educ. Innov. Res. 5, 17–28.

Rosenberg, L.B., 1993. Virtual fixtures: Perceptual tools for telerobotic manipulation, in: Proceedings of IEEE Virtual Reality Annual International Symposium. Presented at the Proceedings of IEEE Virtual Reality Annual International Symposium, pp. 76–82. doi:10.1109/VRAIS.1993.380795

Rosenberg, M.J., 2002. E-Learning: Strategies for Delivering Knowledge in the Digital Age. McGraw-Hill Professional.

Ryan, R.., 2006. Intrinsic Motivation Inventory (IMI).

Saengsook, R., 2006. Learning Theories and eLearning, in: Third International Conference. Presented at the eLearning for Knowledge-Based Society, Bangkok, Thailand.

Sanna, A., Manuri, F., Lamberti, F., Paravati, G., Pezzolla, P., 2015. Using handheld devices to sup port augmented reality-based maintenance and assembly tasks, in: 2015 IEEE International Conference on Consumer Electronics (ICCE). Presented at the 2015 IEEE International Conference on Consumer Electronics (ICCE), pp. 178–179. doi:10.1109/ICCE.2015.7066370

Santos, M.E.C., Yamamoto, G., Taketomi, T., Miyazaki, J., Kato, H., 2013. Authoring Augmented Reality Learning Experiences as Learning Objects. IEEE, pp. 506–507. doi:10.1109/ICALT.2013.165

Sato, K., Sakamoto, N., Mihara, S., Shimada, H., 2013. CyPhy-UI: Cyber-Physical User Interaction Paradigm to Control Networked Appliances with Augmented Reality. Presented at the ACHI 2013, The Sixth International Conference on Advances in Computer-Human Interactions, pp. 215–221.

Sayed, E., M, N.A., Zayed, H.H., Sharawy, M.I., El Sayed, N.A.M., Zayed, H.H., Sharawy, M.I., 2011. ARSC: Augmented Reality Student Card--An Augmented Reality Solution for the Education Field. Comput. Educ. 56, 1045–1061.

Schmalstieg, D., Hesina, G., 2002. Distributed applications for collaborative augmented reality, in: IEEE Virtual Reality, 2002. Proceedings. Presented at the IEEE Virtual Reality, 2002. Proceedings, pp. 59–66. doi:10.1109/VR.2002.996505

Schmalstieg, D., Schall, G., Wagner, D., Barakonyi, I., Reitmayr, G., Newman, J., Ledermann, F., 2007. Managing Complex Augmented Reality Models. IEEE Comput. Graph. Appl. 27, 48–57. doi:10.1109/MCG.2007.85

Schoeberl, M., Korsholm, S., Kalibera, T., Ravn, A.P., 2011. A Hardware Abstraction Layer in Java. ACM Trans Embed Comput Syst 10, 42:1–42:40. doi:10.1145/2043662.2043666

Scott, T., 2003. Bloom’s Taxonomy Applied to Testing in Computer Science Classes. J Comput Sci Coll 19, 267–274.

Self, J., 1996. Deconstructionist student models in the computer-based learning of science, in: Sánchez, A.D. de I., Castro, I.F. de (Eds.), Computer Aided Learning and Instruction in Science and Engineering, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 27–36. doi:10.1007/BFb0022587

Shapiro, S.S., Wilk, M.B., 1965. An Analysis of Variance Test for Normality (Complete Samples). Biometrika 52, 591–611. doi:10.2307/2333709

Shaw, E., Johnson, W.L., Ganeshan, R., 1999. Pedagogical agents on the web, in: Proceedings of the Third Annual Conference on Autonomous Agents. ACM, pp. 283–290.

Shelton, B.E., 2002. Using augmented reality for teaching Earth-Sun relationships to undergraduate geography students, in: Students”, ART02, The First IEEE International Augmented Reality Toolkit Workshop.

Shirazi, A., Behzadan, A.H., 2015. Content Delivery Using Augmented Reality to Enhance Students’ Performance in a Building Design and Assembly Project. Adv. Eng. Educ. 4, 1–24.

Shirazi, A., Behzadan, A.H., 2013. Assessing the pedagogical value of augmented reality-based learning in construction engineering, in: Proceedings of the 13th

International Conference on Construction Applications of Virtual Reality (CONVR), London, UK.

Shneider, D.E., Gladkikh, D.O., 2006. Designing questioning strategies for information technology courses. Presented at the The 19th Annual Conference of the National Advisory Committee on Computing Qualifications: Preparing for the Future — Capitalising on IT, Wellington, 243248, National Advisory Committee on Computing Qualifications.

Shukla, M., Shukla, A.N., 2012. Growth of robotics industry early in 21st century. Int. J. Comput. Eng. Res. 2, 1554–1558.

Sklar, E., Richards, D., 2010. Agent-based systems for human learners. Knowl. Eng. Rev. 25, 111–135. doi:10.1017/S0269888910000044

SMBus, 2009. The System Management Interface Forum (SMIF) Inc. SMBus (System Management Bus).

Smith, P.L., Ragan, T.J., 2004. Instructional Design, 3rd Revised edition edition. ed. John Wiley & Sons, Hoboken, N.J.

Soliman, M., Guetl, C., 2010. Intelligent Pedagogical Agents in immersive virtual learning environments: A review, in: The 33rd International Convention MIPRO. Presented at the The 33rd International Convention MIPRO, pp. 827–832.

Sosteric, M., Hesemeier, S., 2002. When is a Learning Object not an Object: A first step towards a theory of learning objects. Int. Rev. Res. Open Distrib. Learn. 3. doi:10.19173/irrodl.v3i2.106

Sotiriou, S., Bogner, F.X., 2008. Visualizing the Invisible: Augmented Reality as an Innovative Science Education Scheme. Adv. Sci. Lett. 1, 114–122. doi:10.1166/asl.2008.012

Specht, M., 2009. Learning in a technology enhanced world.

Squire, K., Klopfer, E., 2007. Augmented Reality Simulations on Handheld Computers. J. Learn. Sci. 16, 371–413.

Squire, K.D., Jan, M., 2007. Mad City Mystery: Developing Scientific Argumentation Skills with a Place-based Augmented Reality Game on Handheld Computers. J. Sci. Educ. Technol. 16, 5–29. doi:10.1007/s10956-006-9037-z

Squire, K.D., Jan, M., Matthews, J., Wagler, M., Martin, J., Devane, B., others, 2007. Wherever you go, there you are: place-based augmented reality games for learning. Des. Use Simul. Comput. Games Educ. 265.

Srivastava, A., Yammiyavar, P., 2016. Augmenting tutoring of students using Tangible Smart Learning Objects: An IOT based approach to assist student learning in laboratories, in: 2016 International Conference on Internet of Things and Applications (IOTA). Presented at the 2016 International Conference on Internet of Things and Applications (IOTA), pp. 424–426. doi:10.1109/IOTA.2016.7562765

Ştefan, L., 2012. Immersive Collaborative Environments for Teaching and Learning Traditional Design. Procedia - Soc. Behav. Sci., The World Conference on Design, Arts and Education (DAE-2012), May 1-3 2012, Antalya, Turkey 51, 1056–1060. doi:10.1016/j.sbspro.2012.08.287

Stoker, G., Arnold, T., Maxwell, P., 2013. Using Virtual Machines to Improve Learning and Save Resources in an Introductory IT Course, in: Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education, SIGITE '13. ACM, New York, NY, USA, pp. 91–96. doi:10.1145/2512276.2512287

Štuikys, V., 2015. Smart Learning Objects for Smart Education in Computer Science. Springer International Publishing, Cham.

Suh, Y.-H., Lee, K.-W., Lee, M., Kim, H., Cho, E.-S., 2012. ICARS : Integrated Control Architecture for the Robotic Mediator in Smart Environments: A Software Framework for the Robotic Mediator Collaborating with Smart Environments, in: 2012 IEEE 14th International Conference on High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS). Presented at the 2012 IEEE 14th International Conference on High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), pp. 1541–1548. doi:10.1109/HPCC.2012.225

Sutherland, I.E., 1968. A Head-mounted Three Dimensional Display, in: Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I, AFIPS ’68 (Fall, Part I). ACM, New York, NY, USA, pp. 757–764. doi:10.1145/1476589.1476686

Swan, J.E., Gabbard, J.L., 2005. Survey of user-based experimentation in augmented reality, in: Proceedings of 1st International Conference on Virtual Reality. pp. 1–9.

Sztipanovits, J., Biswas, G., Frampton, K., Gokhale, A., Howard, L., Karsai, G., Koo, T.J., Koutsoukos, X., Schmidt, D.C., 2005. Introducing Embedded Software and Systems Education and Advanced Learning Technology in an Engineering Curriculum. ACM Trans Embed Comput Syst 4, 549–568. doi:10.1145/1086519.1086524

Szyperski, C., Bosch, J., Weck, W., 1999. Component-Oriented Programming, in: Moreira, A. (Ed.), Object-Oriented Technology ECOOP’99 Workshop Reader, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 184–192. doi:10.1007/3-540-46589-8_10

Tang, A., Owen, C., Biocca, F., Mou, W., 2003. Comparative Effectiveness of Augmented Reality in Object Assembly, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’03. ACM, New York, NY, USA, pp. 73–80. doi:10.1145/642611.642626

Tang, A., Owen, C., Biocca, F., Mou, W., 2002. Experimental Evaluation of Augmented Reality in Object Assembly Task, in: Proceedings of the 1st International Symposium on Mixed and Augmented Reality. IEEE Computer Society, p. 265.

Tanner, P., Karas, C., Schofield, D., 2014. Augmenting a child’s reality: using educational tablet technology. J. Inf. Technol. Educ. Innov. Pract. 13, 45–54.

Temerinac, M., Kastelan, I., Skala, K., Rogina, B.M., Reindl, L., Souvestre, F., Anastassova, M., Szewczyk, R., Piwinski, J., Benito, J.R.L., Gonzalez, E.A., Teslic, N., Sruk, V., Barak, M., 2013. E2LP: A Unified Embedded Engineering Learning Platform, in: Proceedings of the 2013 Euromicro Conference on Digital System Design, DSD ’13. IEEE Computer Society, Washington, DC, USA, pp. 266–271. doi:10.1109/DSD.2013.38

The Linux Kernel Archives, 2009. The Linux Kernel Archives. i2c dev interface.

Thomas, B., Close, B., Donoghue, J., Squires, J., Bondi, P.D., Morris, M., Piekarski, W., 2000. ARQuake: an outdoor/indoor augmented reality first person application, in: Digest of Papers. Fourth International Symposium on Wearable Computers. Presented at the Digest of Papers. Fourth International

Symposium on Wearable Computers, pp. 139–146. doi:10.1109/ISWC.2000.888480

Thompson, E., Luxton-Reilly, A., Whalley, J.L., Hu, M., Robbins, P., 2008. Bloom's Taxonomy for CS Assessment, in: Proceedings of the Tenth Conference on Australasian Computing Education - Volume 78, ACE '08. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 155–161.

Träskbäack, M., Haller, M., 2004. Mixed reality training application for an oil refinery: user requirements, in: Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry, VRCAI '04. ACM, New York, NY, USA, pp. 324–327. doi:10.1145/1044588.1044658

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A., 2003. A model curriculum for K-12 computer science. Final Rep. ACM K-12 Task Force Curric. Comm. CSTA.

Tufte, E.R., 1991. Envisioning information. Optom. Vis. Sci. 68, 322–324.

Uematsu, Y., Saito, H., 2008. Vision-based Augmented Reality Applications, in: Zhihui, X. (Ed.), Computer Vision. InTech.

van Krevelen, D., Poelman, R., 2010. A Survey of Augmented Reality Technologies, Applications and Limitations. Int. J. Virtual Real. 9, 1–20.

Vasilyeva, E., Puuronen, S., Pechenizkiy, M., Rasanen, P., 2007. Feedback adaptation in web-based learning systems. Int. J. Contin. Eng. Educ. Life Long Learn. 17, 337–357. doi:10.1504/IJCEELL.2007.015046

Vate-U-Lan, P., 2012. An Augmented Reality 3D Pop-Up Book: The Development of a Multimedia Project for English Language Teaching, in: 2012 IEEE International Conference on Multimedia and Expo. Presented at the 2012 IEEE International Conference on Multimedia and Expo, pp. 890–895. doi:10.1109/ICME.2012.79

Veletsianos, G., Russell, G.S., 2014. Pedagogical Agents, in: Spector, J.M., Merrill, M.D., Elen, J., Bishop, M.J. (Eds.), Handbook of Research on Educational Communications and Technology. Springer New York, pp. 759–769. doi:10.1007/978-1-4614-3185-5_61

Venners, B., 1996. Inside the Java Virtual Machine. McGraw-Hill, Inc., New York, NY, USA.

Wagner, D., Pintaric, T., Ledermann, F., Schmalstieg, D., 2005. Towards Massively Multi-user Augmented Reality on Handheld Devices.

Wagner, D., Schmalstieg, D., 2006. Handheld Augmented Reality Displays, in: Virtual Reality Conference, 2006. Presented at the Virtual Reality Conference, 2006, pp. 321–321. doi:10.1109/VR.2006.67

Wang, X., Ong, S.K., Nee, A.Y.C., 2016. A comprehensive survey of augmented reality assembly research. Adv. Manuf. 4, 1–22. doi:10.1007/s40436-015-0131-4

Wang, Y., 2007. Software Engineering Foundations: A Software Science Perspective, 1st ed. Auerbach Publications, Boston, MA, USA.

Wann, J., Mon-Williams, M., 1996. What does virtual reality NEED?: human factors issues in the design of three-dimensional computer environments. Int. J. Hum.-Comput. Stud. 44, 829–847.

Weiser, M., 1999. The Computer for the 21st Century. SIGMOBILE Mob Comput Commun Rev 3, 3–11. doi:10.1145/329124.329126

West, M., Ross, S., 2002. Retaining females in computer science: A new look at a persistent problem. J. Comput. Sci. Coll. 17, 1–7.

Whitmore, A., Agarwal, A., Xu, L.D., 2014. The Internet of Things—A survey of topics and trends. Inf. Syst. Front. 17, 261–274. doi:10.1007/s10796-014-9489-2

Wiedenbeck, s., ramalingam, v., 1999. Novice comprehension of small programs written in the procedural and object-oriented styles. Int. J. Hum.-Comput. Stud. 51, 71–87. doi:10.1006/ijhc.1999.0269

Wiley, D.., Gibbons, A.., Recker, M.., 2000. A reformulation of the issue of learning object granularity and its implications for the design of learning objects.

Wiley, D.A., 2000. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. Instr. Use Learn. Objects Online Version.

Wing, J.M., 2006. Computational Thinking. Commun ACM 49, 33–35. doi:10.1145/1118178.1118215

Wu, H.-K., Lee, S.W.-Y., Chang, H.-Y., Liang, J.-C., 2013. Current status, opportunities and challenges of augmented reality in education. Comput. Educ. 62, 41–49. doi:10.1016/j.compedu.2012.10.024

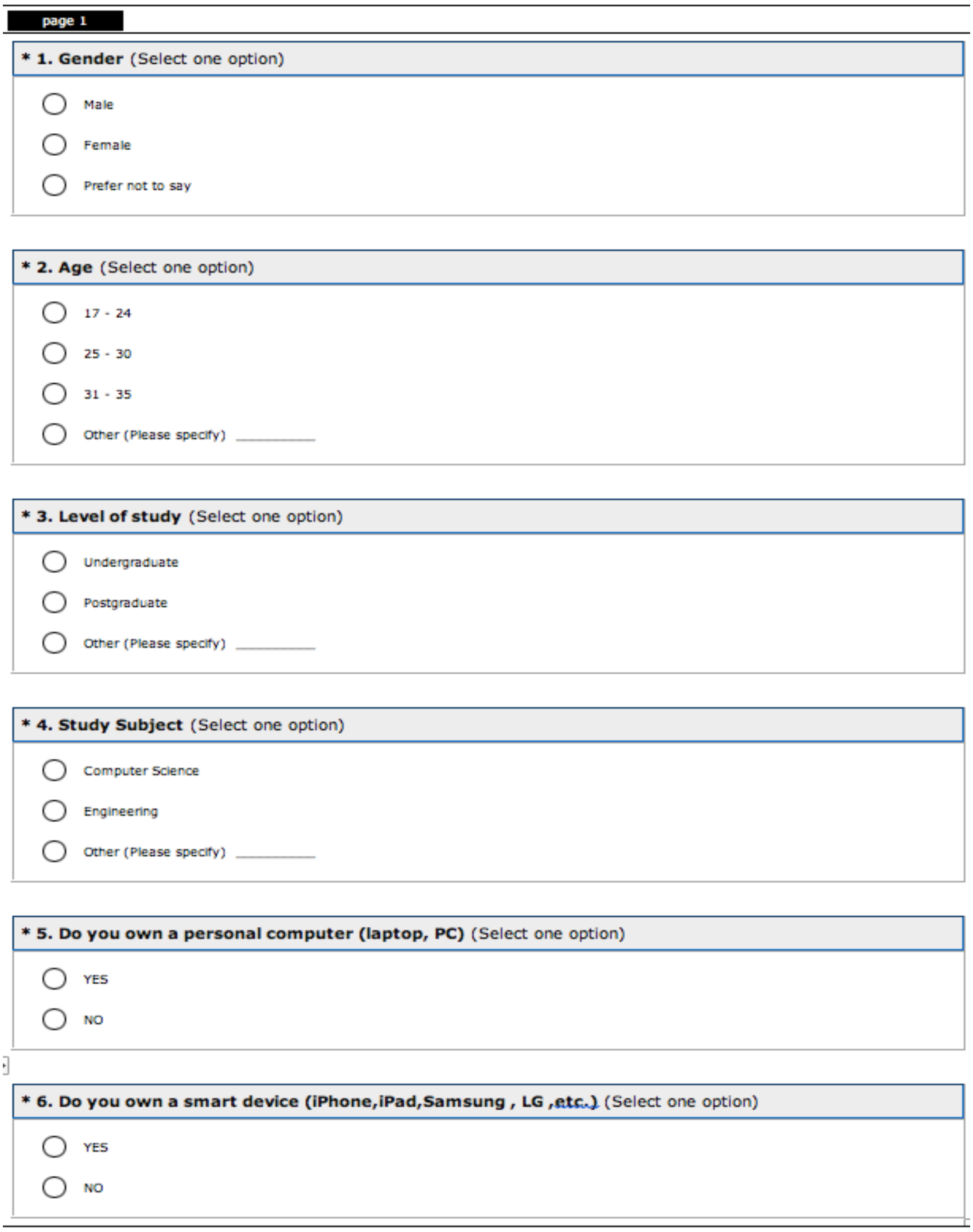
Zhong, X., Liu, P., Georganas, N.D., Boulanger, P., 2003. Designing a Vision-based Collaborative Augmented Reality Application for Industrial Training. It - Inf. Technol. 45, 7–19. doi:10.1524/itit.45.1.7.19032

Zhou, F., Duh, H.B.-L., Billinghurst, M., 2008. Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR, in: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR ’08. IEEE Computer Society, Washington, DC, USA, pp. 193–202. doi:10.1109/ISMAR.2008.4637362

Appendices

A. Experiment I instruments

- **Knowledge and background survey**



page 1

*** 1. Gender** (Select one option)

○ Male
○ Female
○ Prefer not to say

*** 2. Age** (Select one option)

○ 17 - 24
○ 25 - 30
○ 31 - 35
○ Other (Please specify) ________

*** 3. Level of study** (Select one option)

○ Undergraduate
○ Postgraduate
○ Other (Please specify) ________

*** 4. Study Subject** (Select one option)

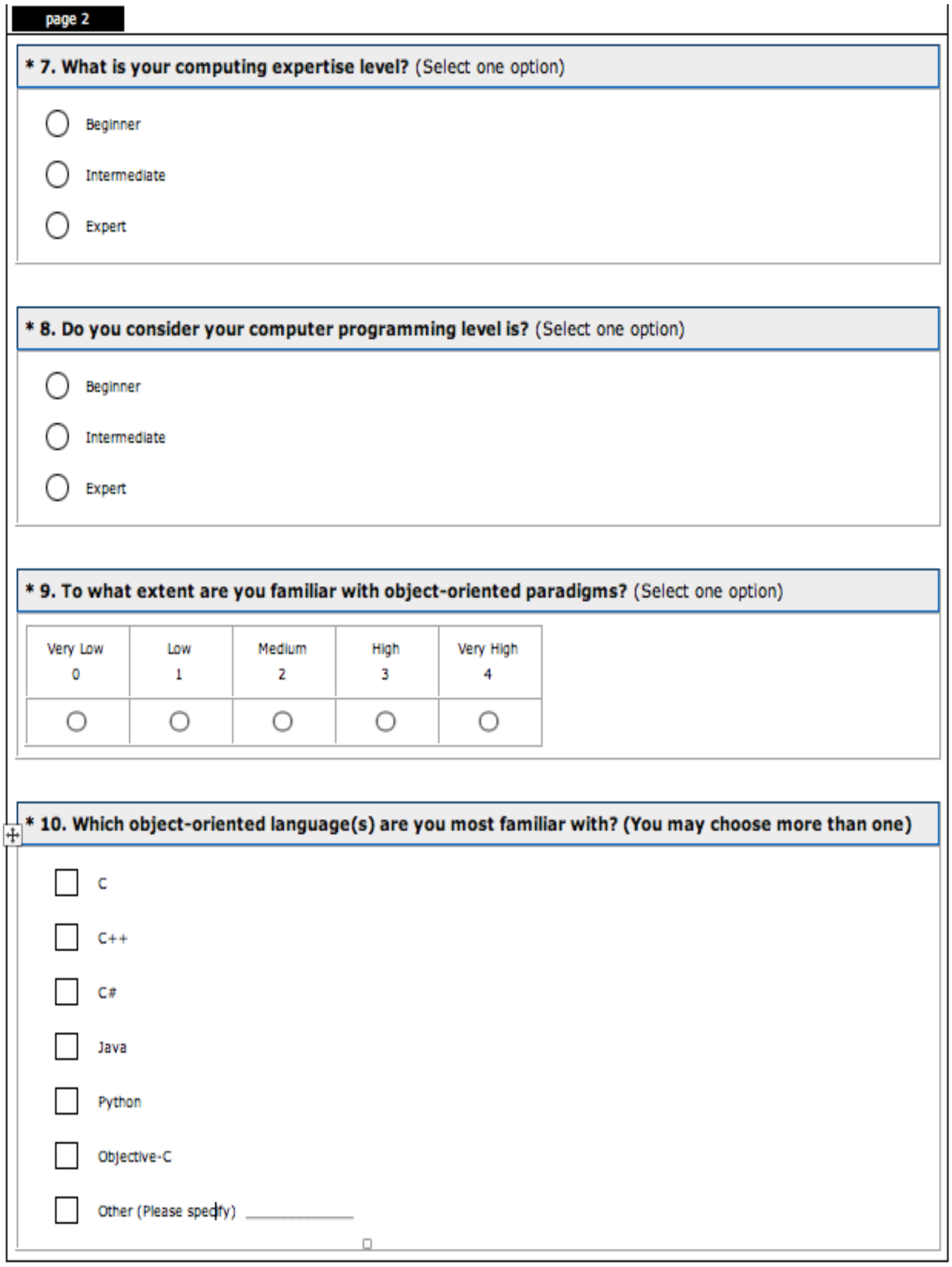
○ Computer Science
○ Engineering
○ Other (Please specify) ________

*** 5. Do you own a personal computer (laptop, PC)** (Select one option)

○ YES
○ NO

*** 6. Do you own a smart device (iPhone,iPad,Samsung , LG ,etc.)** (Select one option)

○ YES
○ NO



page 2

*** 7. What is your computing expertise level?** (Select one option)

○ Beginner

○ Intermediate

○ Expert

*** 8. Do you consider your computer programming level is?** (Select one option)

○ Beginner

○ Intermediate

○ Expert

*** 9. To what extent are you familiar with object-oriented paradigms?** (Select one option)

Very Low 0	Low 1	Medium 2	High 3	Very High 4
○	○	○	○	○

*** 10. Which object-oriented language(s) are you most familiar with? (You may choose more than one)**

☐ C

☐ C++

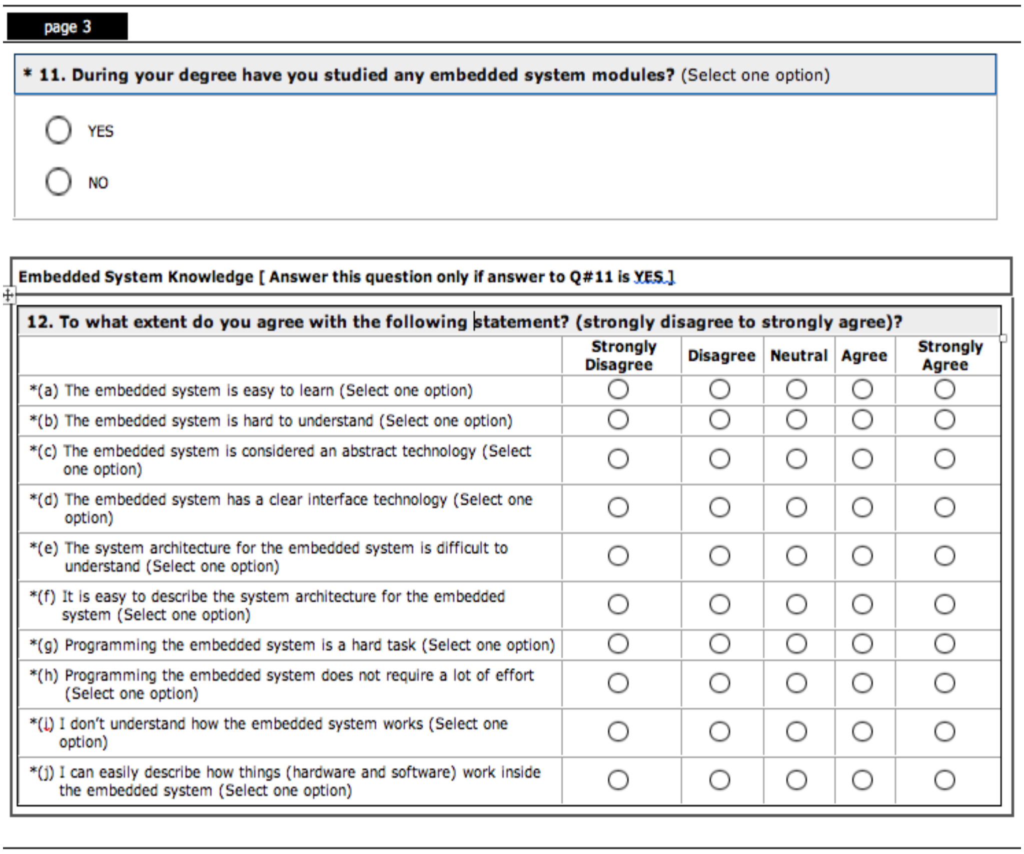
☐ C#

☐ Java

☐ Python

☐ Objective-C

☐ Other (Please specify) ______________



page 3

*** 11. During your degree have you studied any embedded system modules?** (Select one option)

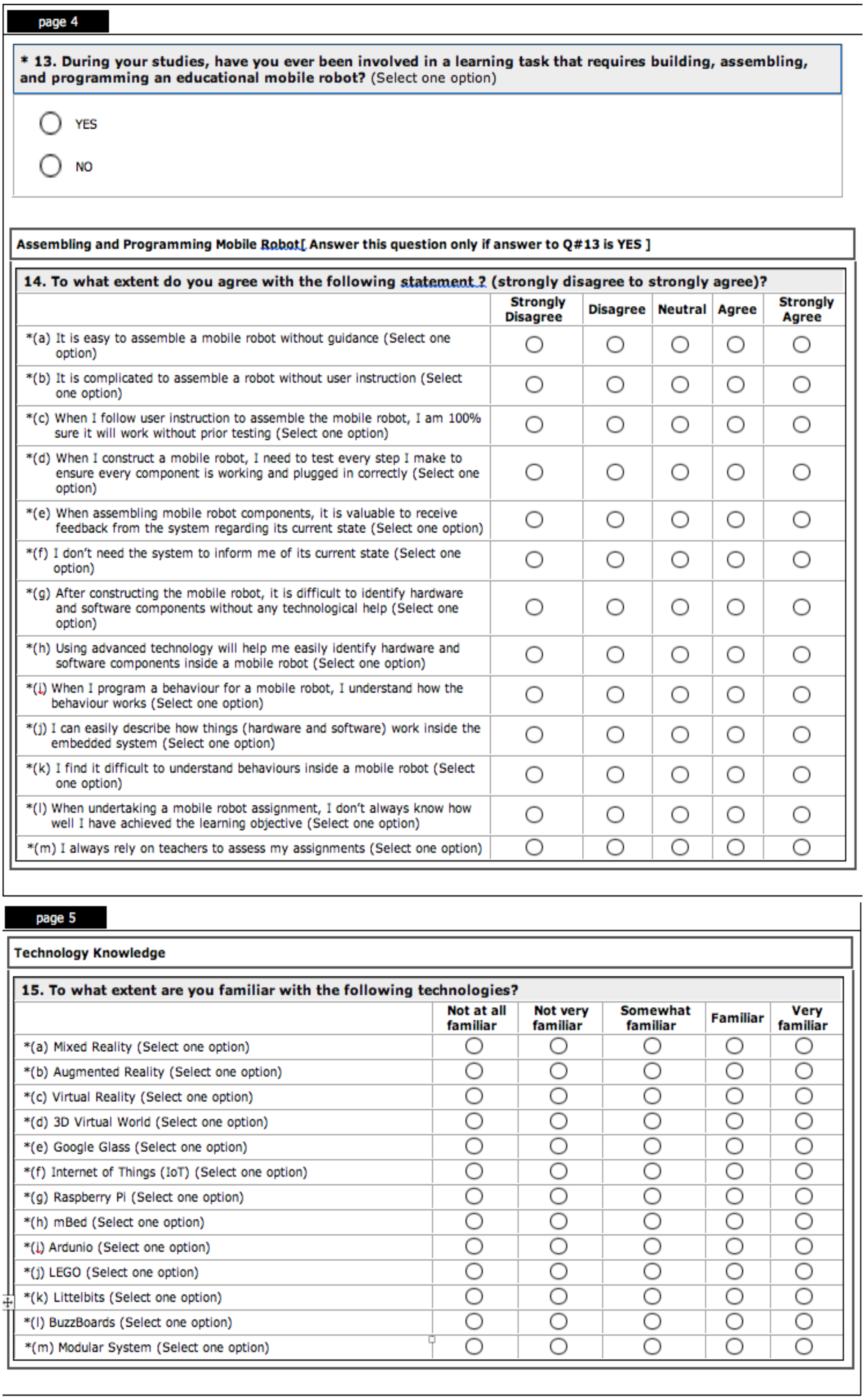
○ YES

○ NO

Embedded System Knowledge [Answer this question only if answer to Q#11 is YES]

12. To what extent do you agree with the following statement? (strongly disagree to strongly agree)?

	Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
*(a) The embedded system is easy to learn (Select one option)	○	○	○	○	○
*(b) The embedded system is hard to understand (Select one option)	○	○	○	○	○
*(c) The embedded system is considered an abstract technology (Select one option)	○	○	○	○	○
*(d) The embedded system has a clear interface technology (Select one option)	○	○	○	○	○
*(e) The system architecture for the embedded system is difficult to understand (Select one option)	○	○	○	○	○
*(f) It is easy to describe the system architecture for the embedded system (Select one option)	○	○	○	○	○
*(g) Programming the embedded system is a hard task (Select one option)	○	○	○	○	○
*(h) Programming the embedded system does not require a lot of effort (Select one option)	○	○	○	○	○
*(i) I don't understand how the embedded system works (Select one option)	○	○	○	○	○
*(j) I can easily describe how things (hardware and software) work inside the embedded system (Select one option)	○	○	○	○	○



page 4

*** 13. During your studies, have you ever been involved in a learning task that requires building, assembling, and programming an educational mobile robot?** (Select one option)

○ YES

○ NO

Assembling and Programming Mobile Robot[Answer this question only if answer to Q#13 is YES]

14. To what extent do you agree with the following statement ? (strongly disagree to strongly agree)?

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
*(a) It is easy to assemble a mobile robot without guidance (Select one option)	○	○	○	○	○
*(b) It is complicated to assemble a robot without user instruction (Select one option)	○	○	○	○	○
*(c) When I follow user instruction to assemble the mobile robot, I am 100% sure it will work without prior testing (Select one option)	○	○	○	○	○
*(d) When I construct a mobile robot, I need to test every step I make to ensure every component is working and plugged in correctly (Select one option)	○	○	○	○	○
*(e) When assembling mobile robot components, it is valuable to receive feedback from the system regarding its current state (Select one option)	○	○	○	○	○
*(f) I don't need the system to inform me of its current state (Select one option)	○	○	○	○	○
*(g) After constructing the mobile robot, it is difficult to identify hardware and software components without any technological help (Select one option)	○	○	○	○	○
*(h) Using advanced technology will help me easily identify hardware and software components inside a mobile robot (Select one option)	○	○	○	○	○
*(i) When I program a behaviour for a mobile robot, I understand how the behaviour works (Select one option)	○	○	○	○	○
*(j) I can easily describe how things (hardware and software) work inside the embedded system (Select one option)	○	○	○	○	○
*(k) I find it difficult to understand behaviours inside a mobile robot (Select one option)	○	○	○	○	○
*(l) When undertaking a mobile robot assignment, I don't always know how well I have achieved the learning objective (Select one option)	○	○	○	○	○
*(m) I always rely on teachers to assess my assignments (Select one option)	○	○	○	○	○

page 5

Technology Knowledge

15. To what extent are you familiar with the following technologies?

	Not at all familiar	Not very familiar	Somewhat familiar	Familiar	Very familiar
*(a) Mixed Reality (Select one option)	○	○	○	○	○
*(b) Augmented Reality (Select one option)	○	○	○	○	○
*(c) Virtual Reality (Select one option)	○	○	○	○	○
*(d) 3D Virtual World (Select one option)	○	○	○	○	○
*(e) Google Glass (Select one option)	○	○	○	○	○
*(f) Internet of Things (IoT) (Select one option)	○	○	○	○	○
*(g) Raspberry Pi (Select one option)	○	○	○	○	○
*(h) mBed (Select one option)	○	○	○	○	○
*(i) Ardunio (Select one option)	○	○	○	○	○
*(j) LEGO (Select one option)	○	○	○	○	○
*(k) Littelbits (Select one option)	○	○	○	○	○
*(l) BuzzBoards (Select one option)	○	○	○	○	○
*(m) Modular System (Select one option)	○	○	○	○	○

- Observation Sheet

Task	How Many?	
Debugging		
Interacting with the physical object		
Has the student interact with all objects?		
Tasks Time	1	
	2	
	3	
	Overall Time	

- Assembly Robot Post-Test

*Required

The main processor in assembling mobile robot is? *

  - ○ Buzz_Berry
  - ○ Raspberry Pi
  - ○ Buzz_Link3
  - ○ Buzz_Bot

To use Buttons sensor, you need to connect the following boards? *

  - ○ Buzz_Bot, Buzz_Link3, Raspberry pi
  - ○ Buzz_Link3, Raspberry Pi
  - ○ Buzz_Berry, Raspberry Pi
  - ○ Buzz_Berry,Raspberry Pi, Buzz_Link3

Light Sensor belongs to which buzzboards? *

  - ○ Buzz_Bot
  - ○ Buzz_Berry
  - ○ Buzz_link3
  - ○ Raspberry Pi

LED's belongs to which buzzboards? *

  - ○ Buzz_Link3
  - ○ Buzz_Berry
  - ○ Raspberry Pi
  - ○ Buzz_Bot

Buzz_Berry has the following? *

  - ○ Line Sensor, LED's Sensor, Buttons Sensor
  - ○ IR ranger finder, Buttons Sensor, Light Sensor
  - ○ LED's Sensor, Butttons Sensor
  - ○ IR range finder, Line Sensor, Light Sensor

Buzz_Bot has the following? *

- ○ Line Sensor, LED's Sensor, Buttons Sensor
- ○ IR ranger finder, Buttons Sensor, Light Sensor
- ○ LED's Sensor, Butttons Sensor
- ○ IR range finder, Line Sensor, Light Sensor

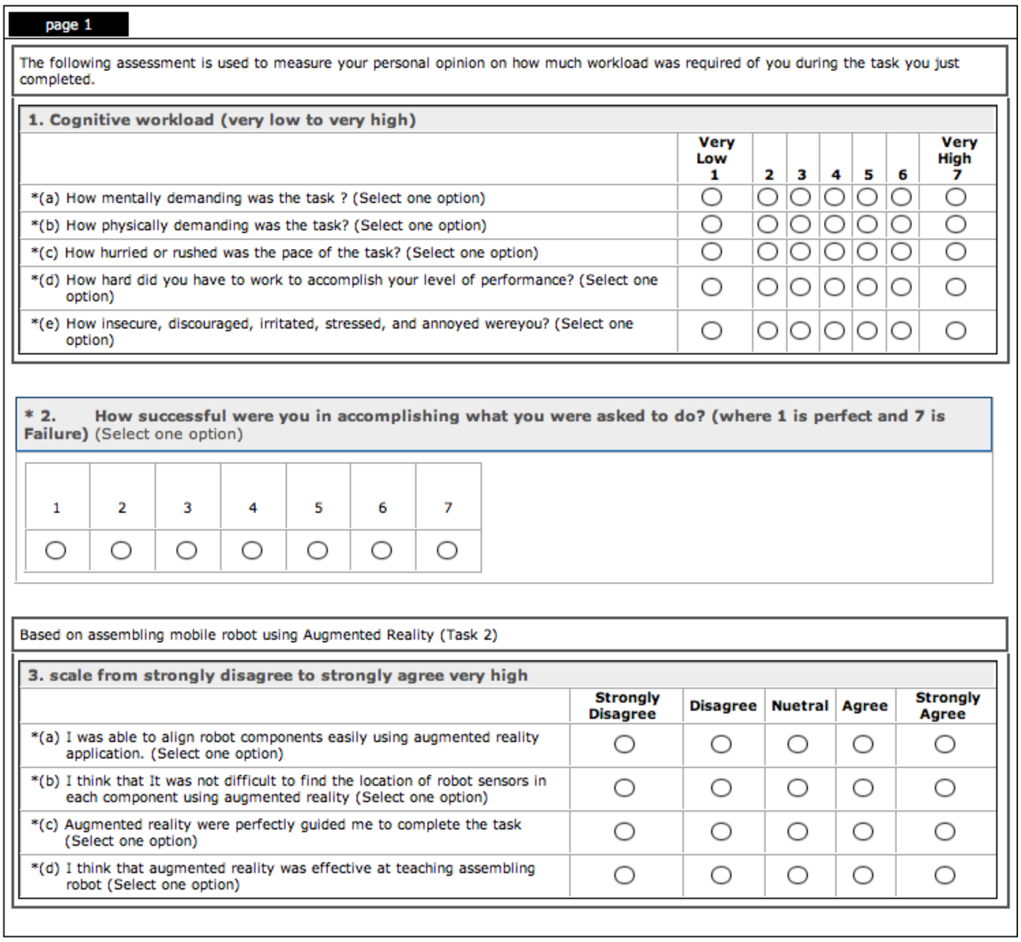
Which of the following is wrong for plugging buzz_Link3 into Buzz_Bot?

- ○

- ○

- ○

- Post Questionnaire



page 1

The following assessment is used to measure your personal opinion on how much workload was required of you during the task you just completed.

1. Cognitive workload (very low to very high)

	Very Low 1	**2**	**3**	**4**	**5**	**6**	**Very High 7**
*(a) How mentally demanding was the task ? (Select one option)	○	○	○	○	○	○	○
*(b) How physically demanding was the task? (Select one option)	○	○	○	○	○	○	○
*(c) How hurried or rushed was the pace of the task? (Select one option)	○	○	○	○	○	○	○
*(d) How hard did you have to work to accomplish your level of performance? (Select one option)	○	○	○	○	○	○	○
*(e) How insecure, discouraged, irritated, stressed, and annoyed wereyou? (Select one option)	○	○	○	○	○	○	○

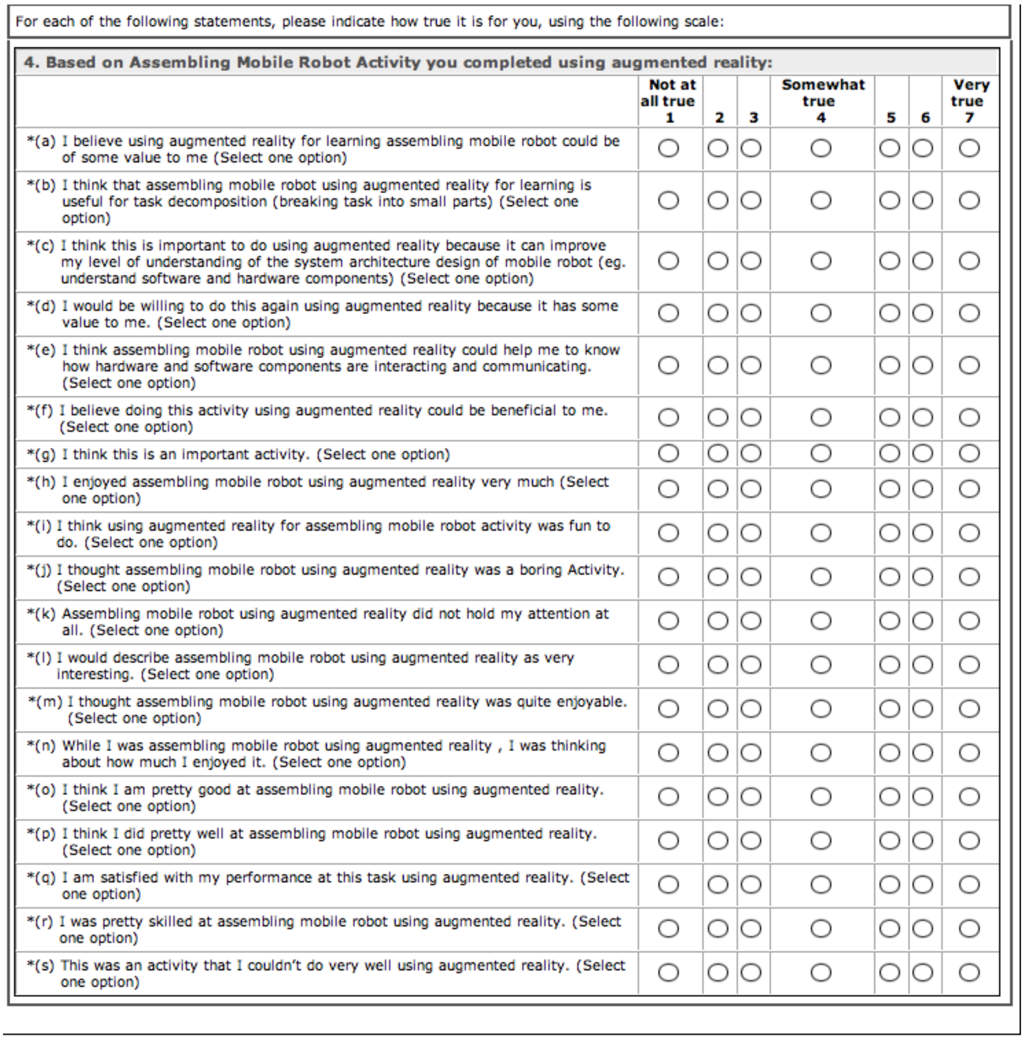
*** 2. How successful were you in accomplishing what you were asked to do? (where 1 is perfect and 7 is Failure)** (Select one option)

1	2	3	4	5	6	7
○	○	○	○	○	○	○

Based on assembling mobile robot using Augmented Reality (Task 2)

3. scale from strongly disagree to strongly agree very high

	Strongly Disagree	**Disagree**	**Nuetral**	**Agree**	**Strongly Agree**
*(a) I was able to align robot components easily using augmented reality application. (Select one option)	○	○	○	○	○
*(b) I think that It was not difficult to find the location of robot sensors in each component using augmented reality (Select one option)	○	○	○	○	○
*(c) Augmented reality were perfectly guided me to complete the task (Select one option)	○	○	○	○	○
*(d) I think that augmented reality was effective at teaching assembling robot (Select one option)	○	○	○	○	○

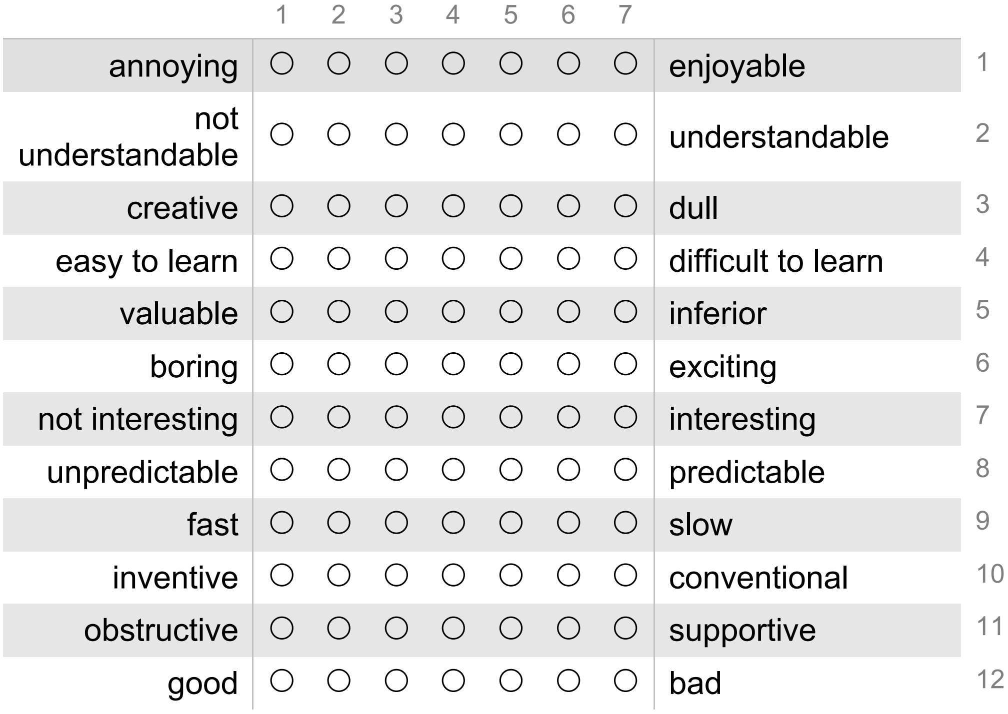


For each of the following statements, please indicate how true it is for you, using the following scale:

4. Based on Assembling Mobile Robot Activity you completed using augmented reality:

	Not at all true 1	2	3	Somewhat true 4	5	6	Very true 7
*(a) I believe using augmented reality for learning assembling mobile robot could be of some value to me (Select one option)	○	○	○	○	○	○	○
*(b) I think that assembling mobile robot using augmented reality for learning is useful for task decomposition (breaking task into small parts) (Select one option)	○	○	○	○	○	○	○
*(c) I think this is important to do using augmented reality because it can improve my level of understanding of the system architecture design of mobile robot (eg. understand software and hardware components) (Select one option)	○	○	○	○	○	○	○
*(d) I would be willing to do this again using augmented reality because it has some value to me. (Select one option)	○	○	○	○	○	○	○
*(e) I think assembling mobile robot using augmented reality could help me to know how hardware and software components are interacting and communicating. (Select one option)	○	○	○	○	○	○	○
*(f) I believe doing this activity using augmented reality could be beneficial to me. (Select one option)	○	○	○	○	○	○	○
*(g) I think this is an important activity. (Select one option)	○	○	○	○	○	○	○
*(h) I enjoyed assembling mobile robot using augmented reality very much (Select one option)	○	○	○	○	○	○	○
*(i) I think using augmented reality for assembling mobile robot activity was fun to do. (Select one option)	○	○	○	○	○	○	○
*(j) I thought assembling mobile robot using augmented reality was a boring Activity. (Select one option)	○	○	○	○	○	○	○
*(k) Assembling mobile robot using augmented reality did not hold my attention at all. (Select one option)	○	○	○	○	○	○	○
*(l) I would describe assembling mobile robot using augmented reality as very interesting. (Select one option)	○	○	○	○	○	○	○
*(m) I thought assembling mobile robot using augmented reality was quite enjoyable. (Select one option)	○	○	○	○	○	○	○
*(n) While I was assembling mobile robot using augmented reality , I was thinking about how much I enjoyed it. (Select one option)	○	○	○	○	○	○	○
*(o) I think I am pretty good at assembling mobile robot using augmented reality. (Select one option)	○	○	○	○	○	○	○
*(p) I think I did pretty well at assembling mobile robot using augmented reality. (Select one option)	○	○	○	○	○	○	○
*(q) I am satisfied with my performance at this task using augmented reality. (Select one option)	○	○	○	○	○	○	○
*(r) I was pretty skilled at assembling mobile robot using augmented reality. (Select one option)	○	○	○	○	○	○	○
*(s) This was an activity that I couldn't do very well using augmented reality. (Select one option)	○	○	○	○	○	○	○

- User experience questionnaire



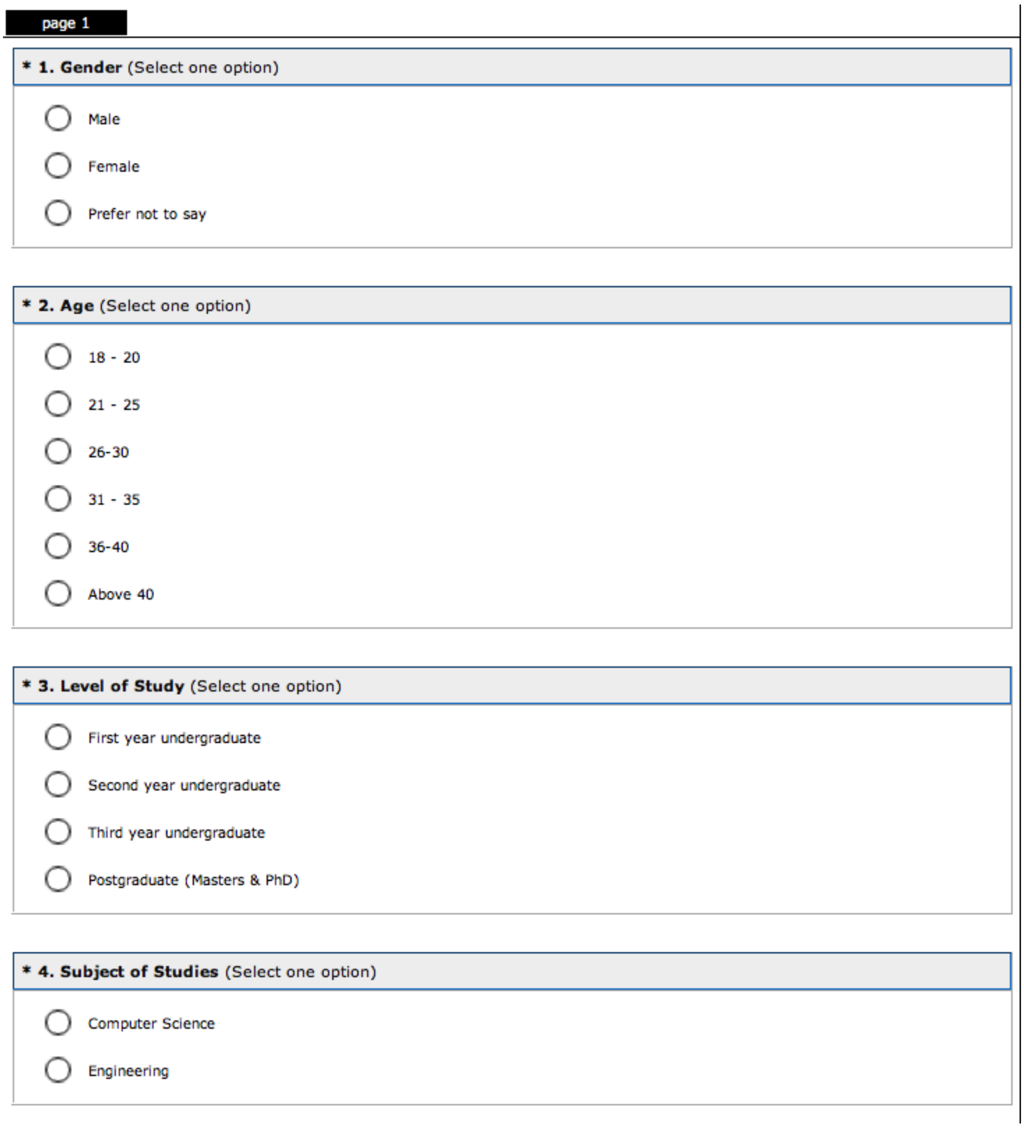
	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12

complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

-

B. Experiment II instruments

- Background and knowledge survey



*** 1. Gender** (Select one option)

○ Male

○ Female

○ Prefer not to say

*** 2. Age** (Select one option)

○ 18 - 20

○ 21 - 25

○ 26-30

○ 31 - 35

○ 36-40

○ Above 40

*** 3. Level of Study** (Select one option)

○ First year undergraduate

○ Second year undergraduate

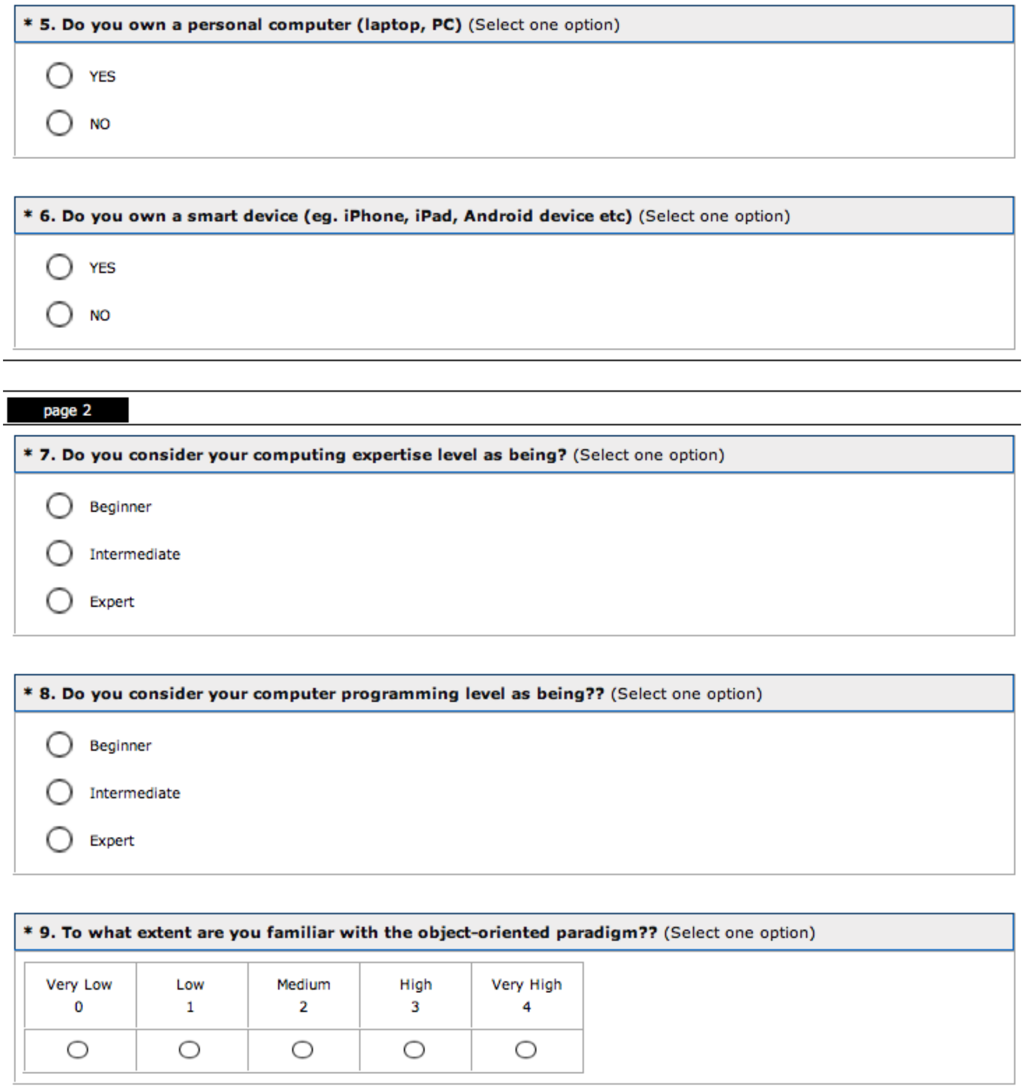
○ Third year undergraduate

○ Postgraduate (Masters & PhD)

*** 4. Subject of Studies** (Select one option)

○ Computer Science

○ Engineering



*** 5. Do you own a personal computer (laptop, PC)** (Select one option)

○ YES

○ NO

*** 6. Do you own a smart device (eg. iPhone, iPad, Android device etc)** (Select one option)

○ YES

○ NO

page 2

*** 7. Do you consider your computing expertise level as being?** (Select one option)

○ Beginner

○ Intermediate

○ Expert

*** 8. Do you consider your computer programming level as being??** (Select one option)

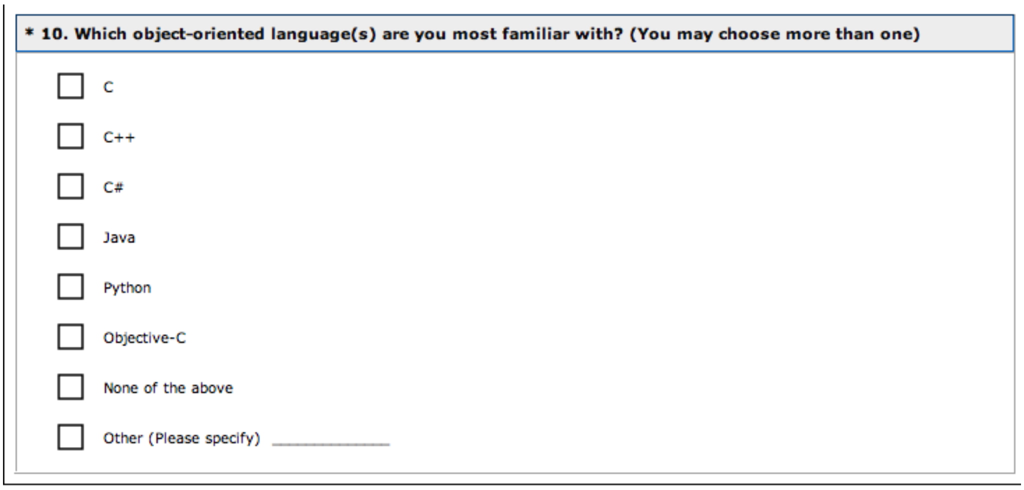
○ Beginner

○ Intermediate

○ Expert

*** 9. To what extent are you familiar with the object-oriented paradigm??** (Select one option)

Very Low 0	Low 1	Medium 2	High 3	Very High 4
○	○	○	○	○



*** 10. Which object-oriented language(s) are you most familiar with? (You may choose more than one)**

☐ C

☐ C++

☐ C#

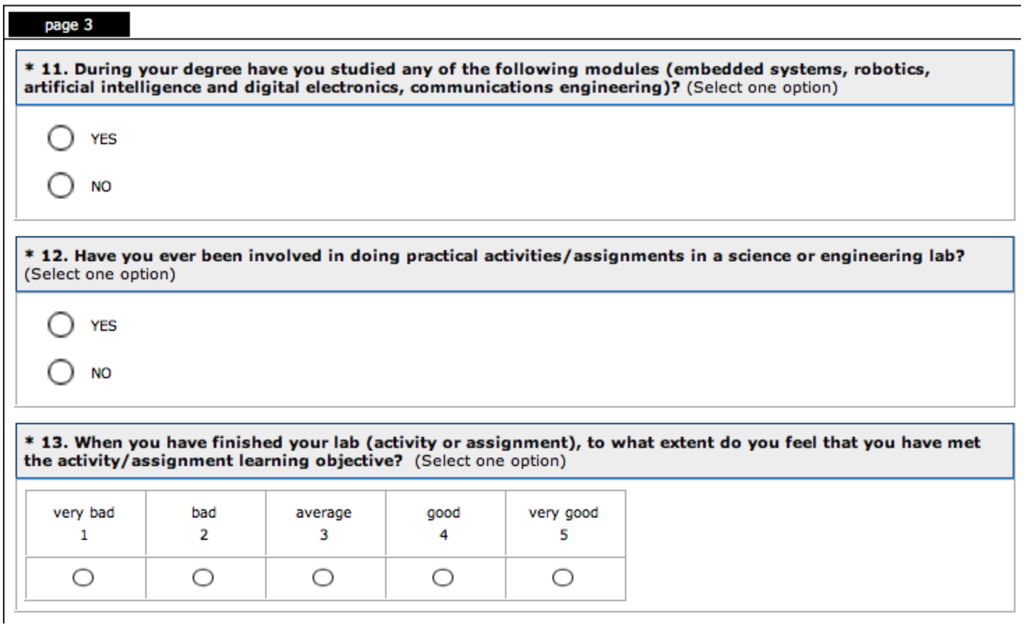
☐ Java

☐ Python

☐ Objective-C

☐ None of the above

☐ Other (Please specify) ____________



page 3

*** 11. During your degree have you studied any of the following modules (embedded systems, robotics, artificial intelligence and digital electronics, communications engineering)?** (Select one option)

○ YES

○ NO

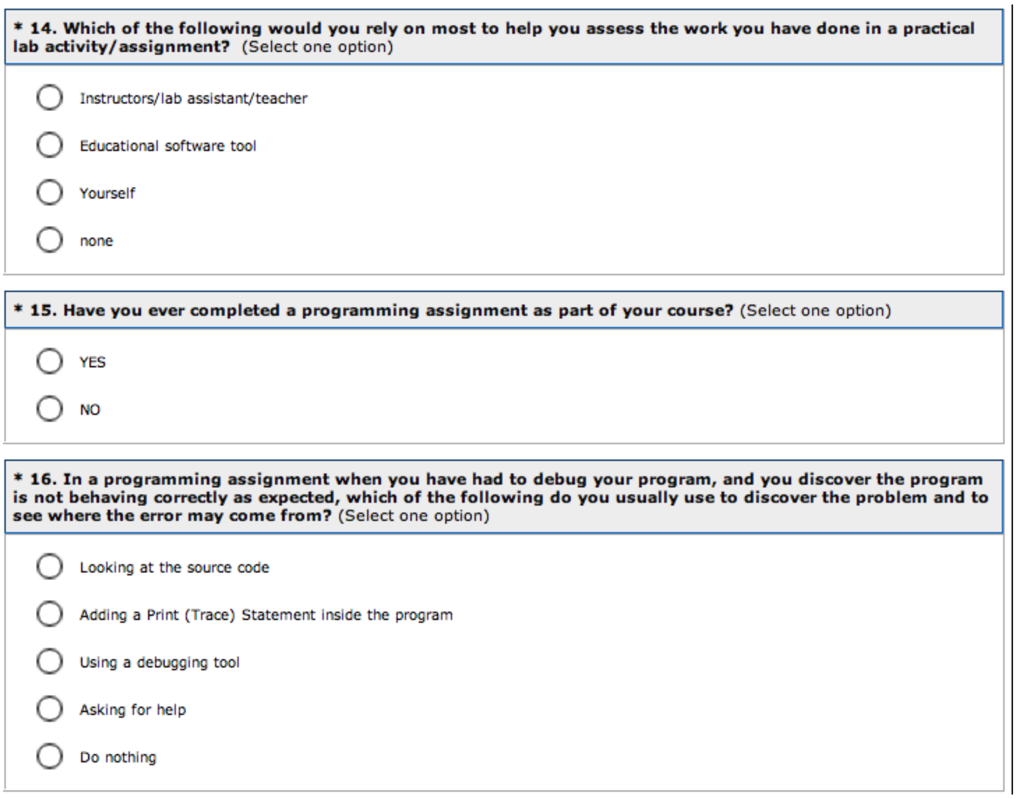
*** 12. Have you ever been involved in doing practical activities/assignments in a science or engineering lab?** (Select one option)

○ YES

○ NO

*** 13. When you have finished your lab (activity or assignment), to what extent do you feel that you have met the activity/assignment learning objective?** (Select one option)

very bad 1	bad 2	average 3	good 4	very good 5
○	○	○	○	○



*** 14. Which of the following would you rely on most to help you assess the work you have done in a practical lab activity/assignment?** (Select one option)

○ Instructors/lab assistant/teacher

○ Educational software tool

○ Yourself

○ none

*** 15. Have you ever completed a programming assignment as part of your course?** (Select one option)

○ YES

○ NO

*** 16. In a programming assignment when you have had to debug your program, and you discover the program is not behaving correctly as expected, which of the following do you usually use to discover the problem and to see where the error may come from?** (Select one option)

○ Looking at the source code

○ Adding a Print (Trace) Statement inside the program

○ Using a debugging tool

○ Asking for help

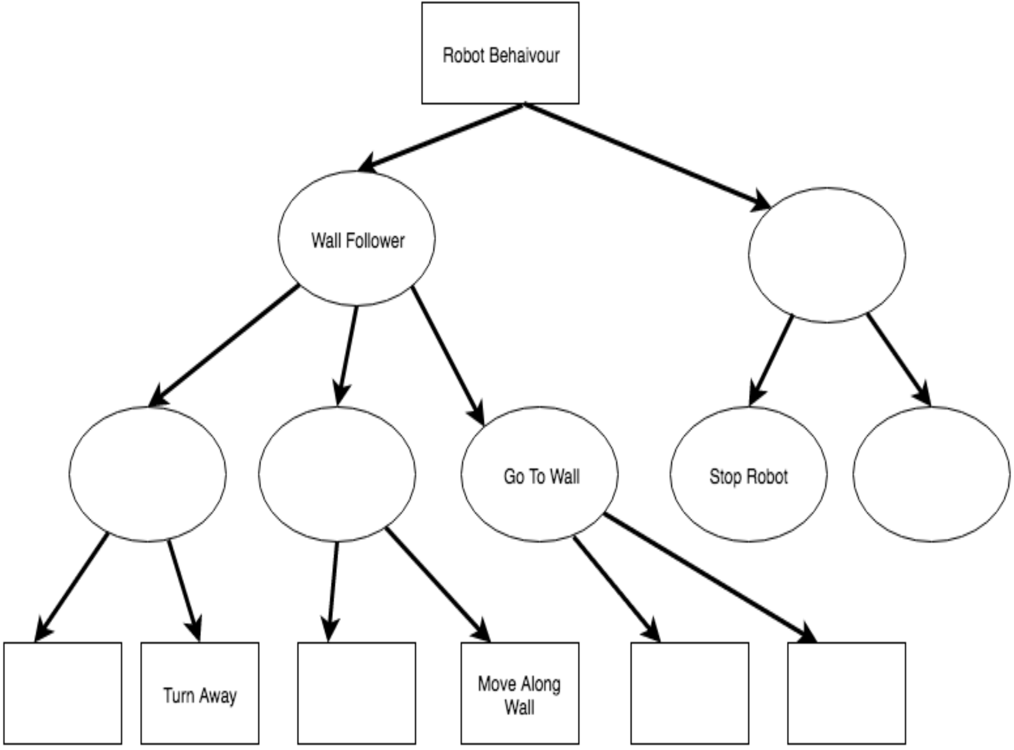
○ Do nothing

- Post Knowledge test

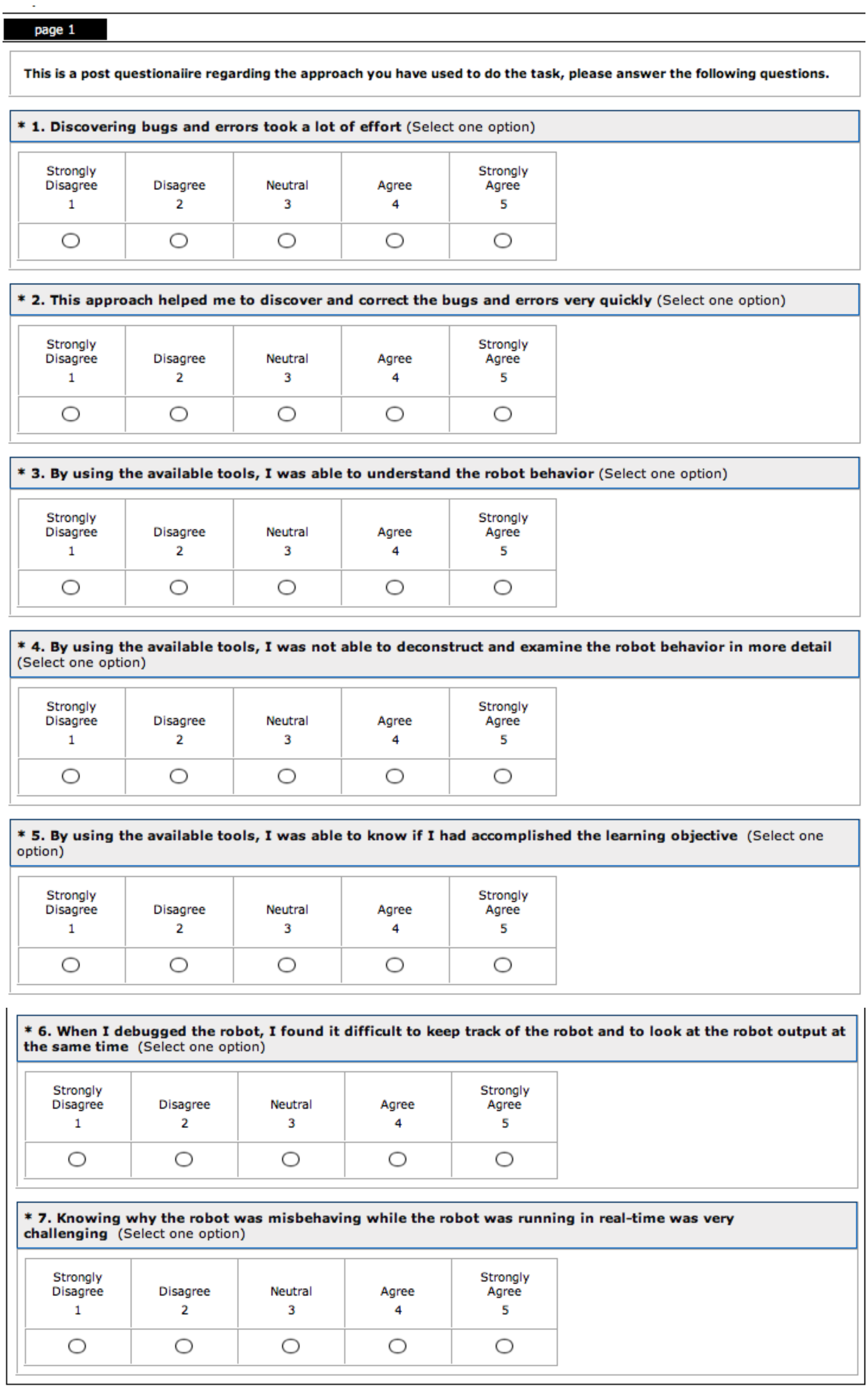
Post Test:

1. **When the robot was near the wall from left side, the robot made a wrong action; please name the wrong action and correct it?**
    a. **The wrong action was……………………………………………**
    b. **The correct action is ………………………………………………**
2. **When the robot detected a wall in a moderate distance from the left side, the robot made a wrong action; please name the wrong action and correct it?**
    a. **The wrong action was……………………………………………**
    b. **The correct action is ………………………………………………**
3. **When the robot detected a wall from a far distance from the left side, the robot made a wrong action; please name the wrong action and correct it?**
    a. **The wrong action was……………………………………………**
    b. **The correct action is ………………………………………………**
4. **When the robot detected an obstacle from near distance from the front side, the robot made a wrong action; please name the wrong action and correct it?**
    a. **The wrong action was……………………………………………**
    b. **The correct action is ………………………………………………**
5. **Move_Along_Wall module can be decomposed into two actions; can you name these actions?**
    a. **……………………**
    b. **……………………**
6. **Turn_Away module can be decomposed into two actions; can you name these actions?**
    a. **……………………**
    b. **……………………**

- Pre and post computational thinking test



- Post questionnaire



page 1

This is a post questionaiire regarding the approach you have used to do the task, please answer the following questions.

*** 1. Discovering bugs and errors took a lot of effort** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 2. This approach helped me to discover and correct the bugs and errors very quickly** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 3. By using the available tools, I was able to understand the robot behavior** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 4. By using the available tools, I was not able to deconstruct and examine the robot behavior in more detail** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 5. By using the available tools, I was able to know if I had accomplished the learning objective** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 6. When I debugged the robot, I found it difficult to keep track of the robot and to look at the robot output at the same time** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 7. Knowing why the robot was misbehaving while the robot was running in real-time was very challenging** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

- Cognitive workload

Mental Demand How mentally demanding was the task?

Very Low Very High

Physical Demand How physically demanding was the task?

Very Low Very High

Temporal Demand How hurried or rushed was the pace of the task?

Very Low Very High

Performance How successful were you in accomplishing what you were asked to do?

Perfect Failure

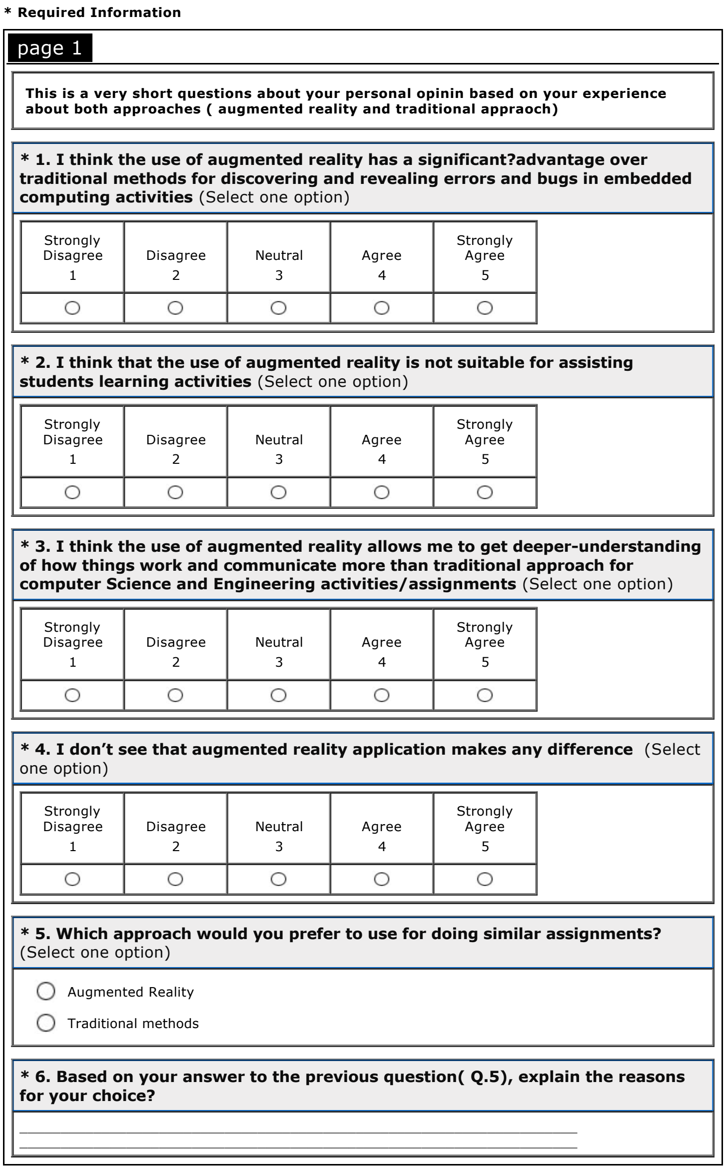
Effort How hard did you have to work to accomplish your level of performance?

Very Low Very High

Frustration How insecure, discouraged, irritated, stressed, and annoyed were you?

Very Low Very High

- Post Opinion



*** Required Information**

page 1

This is a very short questions about your personal opinin based on your experience about both approaches (augmented reality and traditional appraoch)

*** 1. I think the use of augmented reality has a significant?advantage over traditional methods for discovering and revealing errors and bugs in embedded computing activities** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 2. I think that the use of augmented reality is not suitable for assisting students learning activities** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 3. I think the use of augmented reality allows me to get deeper-understanding of how things work and communicate more than traditional approach for computer Science and Engineering activities/assignments** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 4. I don’t see that augmented reality application makes any difference** (Select one option)

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
○	○	○	○	○

*** 5. Which approach would you prefer to use for doing similar assignments?** (Select one option)

○ Augmented Reality

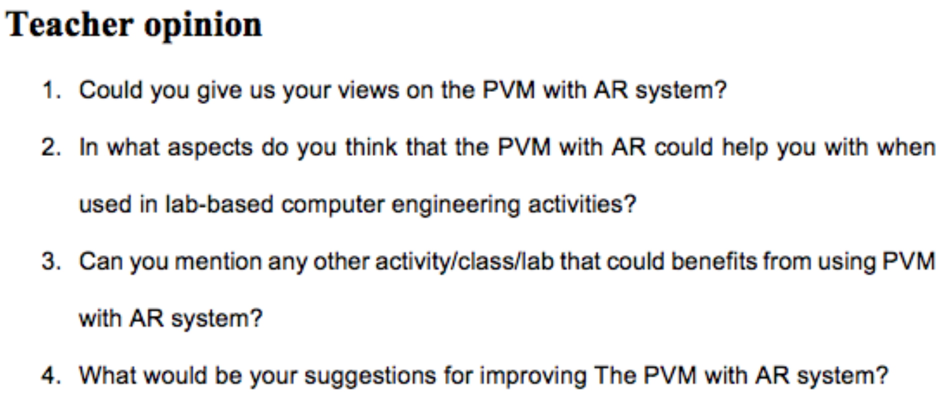
○ Traditional methods

*** 6. Based on your answer to the previous question(Q.5), explain the reasons for your choice?**

- User experience questionnaire

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

- Teacher opinion



Teacher opinion

1. Could you give us your views on the PVM with AR system?
2. In what aspects do you think that the PVM with AR could help you with when used in lab-based computer engineering activities?
3. Can you mention any other activity/class/lab that could benefits from using PVM with AR system?
4. What would be your suggestions for improving The PVM with AR system?

- Participant Consent Form

Please tick the appropriate boxes	**Yes**	**No**
Taking Part		
I confirm that I have read and understood the project information sheet for the above study.	☐	☐
I have been given the opportunity to ask questions about the project.	☐	☐
I agree to take part in the project. Taking part in the project will include observations.	☐	☐
I understand that my taking part is voluntary; I can withdraw from the study at any time and I do not have to give any reasons for why I no longer want to take part.	☐	☐
I agree that the researcher may contact me directly to arrange a research interview.	☐	☐

Name of participant [printed] Signature Date

Researcher [printed] Signature Date

Further information (optional):

Email: ………………………………….